

# An enhanced dynamic-ID-based remote user authentication protocol with smart card

Haoran Chen, Jianhna Chen, Han Shen

**Abstract**— With the blooming development of network technologies, remote user identity authentication is becoming more and more important to ensure that only the legal user can consume the services of the system. Recently, Shi et al. presented an improved remote user authentication scheme with key agreement that attempts to resist various attacks and to achieve perfect user anonymity. However, in this paper we shall show that their scheme is prone to smart card loss attack, offline password guessing attack, impersonation attack and server spoofing attack. What is more, Shi et al. scheme fails to provide user anonymity as they claimed. Then, we put forward an enhanced protocol, which is more secure and suitable for the application environment.

**Index Terms**— dynamic identity; mutual authentication; protocol; smart card.

## I. INTRODUCTION

Owing to the rapid progress in wireless network technology, it is becoming more and more convenient for users to enjoy desired services from service provider servers. At the same time it raises security concern about a system's protected services might be utilized by illegal users in a fraudulent manner. As a result, how to identify a remote user in the open network becomes a crucial issue. To solve this problem, in 1981, Lamport [1] proposed the first password-based scheme by employing a one-way hash function. Due to its simplicity and that the password is easy to memorize, password-based authentication schemes have been widely used to validate the remote user. Since then, numerous [2-5] password-based protocols were proposed. However, in these schemes, the server has to store a sensitive verifier table that contains the passwords of all the registered users. One security threat of this kind of schemes is that once this password-verifier table is leaked, all the registered users' password will be at risk. Thus, some [6-13] user authentication schemes are designed with no verifier table. Owing its portability, low cost, cryptographic and computational capacity nature, the smart card is widely used in these protocols. However, based on fixed identity, schemes [1-13] may leak the registered users' information to malicious attackers and further damage users' privacy. To preserve users' privacy, Das et al. [14] proposed the first dynamic ID-based two-factor authentication scheme in 2004, which they claimed is secure against ID-theft and can resist the reply attacks, forgery attacks, guessing attacks, insider attacks and stolen verifier attacks. Unfortunately, in 2009 Wang et al. [15] pointed out that Das et al.'s scheme is completely insecure for its independence of using passwords. Moreover, their protocol suffers from impersonation attack and fails to provide mutual authentication and user

anonymity. To increase security, they presented an improved version, which was revealed by Chang et al. [16] not a true dynamic-identity based scheme in fact and possesses security holes in the password change phase because an attacker can update the password in a user's smart card at his will. Then they proposed an untraceable remote user authentication scheme with verifiable password update. Unfortunately, Kumari et al. [17] found that the scheme of Chang et al. is completely insecure. So, Kumari et al. also put up with their improved scheme. In 2015, Shi et al. [18] presented an improvement scheme of Kumari et al.'s protocol and they claim that their scheme can resist various attacks and achieve user anonymity. However, based on the security analysis, we find that their scheme is vulnerable to smart card loss attack, offline guessing attack, user impersonation attack and server spoofing attack. Besides, their protocol cannot provide user anonymity. In this paper, an enhanced dynamic-ID-based remote user authentication scheme with smart card is proposed. We will illustrate that with little computational cost our protocol can not only withstand various attacks but also achieve truly user anonymity.

The rest of this paper is organized as follows: in section 2, we review Shi et al.'s scheme briefly. Then their scheme is analyzed in detail in section 3. Next, we proposed an enhanced dynamic-ID-based remote user authentication scheme with smart card in section 4. In section 5, we discuss the security of our new protocol and provide details of the proof. Section 6 describes performance evaluation. Finally, we draw a conclusion in section 7.

## II. REVIEW OF SHI ET AL.'S SCHEME

We will briefly review Shi et al.'s protocol in this section. It consists of the following four phase: registration phase, login phase, authentication phase and password change phase. The notations used throughout this paper are described as follows:

$U_i/U_a/S_i$ : User/Attacker/Server;

$ID_i$ : Identity of  $U_i$ ;

$PW_i$ : Password of  $U_i$ ;

$CID_i$ : Dynamic identity of  $U_i$ ;

$SC_i$ : Smart card of  $U_i$ ;

$r_i$ : Unique random number assigned to  $U_i$  by  $S_i$ ;

$r$ : A random number selected by the smart card;

$T_1, T_2, T_3$ : Current timestamps;

$h(\bullet)$ : One way hash function;

$E_k(\bullet)$ : A symmetric key encryption algorithm and  $k$  is the secret key;

$\oplus$ : Bit-wise exclusive-or (XOR) operation;

$||$  : Connection operation;

### 2.1. Registration Phase

If a user would like to register as a legal user of the system, he would perform the following procedures:

- 1) The user  $U_i$  first chooses his identity  $ID_i$  and password  $PW_i$  freely and chooses a random number  $a$  to compute the value  $R_i = h(a || PW_i)$  and transmits  $\{ID_i, R_i\}$  to the server via a secure channel.
- 2) On receiving the message  $\{ID_i, R_i\}$ , the server chooses a random number  $r_i$  for every registered user  $U_i$ , then computes the value of  $TN_i = h(h(ID_i) || x) \oplus R_i$  and  $TY_i = r_i \oplus h(h(ID_i) || x) \oplus R_i$ ,  $TD_i = h(ID_i || r_i || R_i)$ ,  $TE_i = r_i \oplus h(y || x)$ , he stores  $\{TY_i, TD_i, TE_i\}$  into  $SC_i$  and send it together with the value  $\{TN_i\}$  to  $U_i$  over a secure channel.
- 3) When the user  $U_i$  receives the message from the server  $S_i$ , he computes the value of  $A_i = h(ID_i || pw_i) \oplus a$ ,  $TM_i = TN_i \oplus a$  and keeps the values  $\{A_i, TM_i\}$  in  $SC_i$ .

### 2.2. Login Phase

After executing the registration phase,  $U_i$  becomes the legal user of the system. In order to communicate with the server,  $U_i$  inputs his identity  $ID_i$  and password  $PW_i$  into the smart card then  $SC_i$  performs the following steps:

- 1) First, smart card computes  $a = A_i \oplus h(ID_i || PW_i)$ ,  $R_i = h(a || PW_i)$  then he further calculates the value of  $h(h(ID_i) || x) = TM_i \oplus a \oplus R_i$ ,  $r_i = TY_i \oplus h(h(ID_i) || x) \oplus R_i$ .
- 2) Next, smart card  $SC_i$  computes  $TD_i' = h(ID_i || r_i || R_i)$  and checks whether the equation  $TD_i = TD_i'$  holds or not, if it holds then the smart card  $SC_i$  continues to calculate  $h(y || x) = r_i \oplus TE_i$ ,  $TN_i = TM_i \oplus a$ . Otherwise,  $SC_i$  terminates the session.
- 3) At last, smart card  $SC_i$  gets the current timestamp  $T_1$  and computes user's dynamic identity  $CID_i = h(ID_i) \oplus h(TN_i || r_i || T_1)$  and then  $SC_i$  further calculates the value of  $TG_i = TN_i \oplus h(r_i || T_1)$ ,  $TB_i = TN_i \oplus R_i$ . After that,  $SC_i$  randomly choose a number  $b$  to calculate the value of  $Q_i = h(h(ID_i) || b)$  then  $SC_i$  can obtain the value of  $DS_i = TB_i \oplus Q_i$ ,  $TC_i = h(TN_i || r_i || Q_i || T_1)$  and

$TF_i = r_i \oplus (h(y || x) || T_1)$ . After that,  $SC_i$  transmits the login request  $\{CID_i, TG_i, TC_i, TF_i, DS_i, T_1\}$  to  $S_i$ .

### 2.3. Authentication Phase

In this phase both the user and the server start to take the following steps to authenticate the legitimacy of each other and further consult the common session key.

- 1) Upon receiving the login request  $\{CID_i, TG_i, TC_i, TF_i, DS_i, T_1\}$  from  $U_i$ ,  $S_i$  obtain the current timestamp  $T_2$  and examines the validity of  $T_1$ . That is, if  $T_2 - T_1 \leq \Delta T$  holds,  $T_1$  is valid and  $S_i$  continues to execute the next step. If not so, the procedure will be aborted.
- 2) Next,  $S_i$  retrieves the values  $r_i = TF_i \oplus h(y || x) || T_1$ ,  $TN_i = TG_i \oplus h(r_i || T_1)$  by using his private key  $x$ , then the server continues to compute  $h(ID_i) = CID_i \oplus h(TN_i || r_i || T_1)$ ,  $TB_i^* = h(h(ID_i) || x)$  and  $Q_i^* = TB_i^* \oplus DS_i$ . Next, the server checks whether the equation  $TC_i = h(TN_i || r_i || Q_i^* || T_1)$  holds or not. If not,  $S_i$  terminates the session. Otherwise,  $U_i$  is authenticated as a legal user by  $S_i$ . At last,  $S_i$  gets the current time  $T_3$  to compute  $V_i = h(TB_i^* || r_i || T_3)$  and sends the response message  $\{V_i, T_3\}$  to  $U_i$  immediately.
- 3) Upon receiving  $\{V_i, T_3\}$  from  $S_i$ ,  $U_i$  examines the freshness of  $T_3$ . If  $T_3$  is fresh,  $U_i$  continues to compute  $V_i' = h(TB_i || r_i || T_3)$  and compares  $V_i'$  with the stored value  $V_i$ , if they are equal then  $S_i$  is authenticated as a valid server by  $U_i$ . Otherwise,  $U_i$  drops the message and terminates the session.
- 4) Finally,  $U_i$  computes his session key  $SK_u = h(TB_i || r_i || T_1 || T_3 || h(y || x) || Q_i)$  and  $S_i$  calculates  $SK_s = h(TB_i^* || r_i || T_1 || T_3 || h(y || x) || Q_i^*)$ . From the discussion above, we know that  $TB_i = TB_i^*$  and  $Q_i = Q_i^*$ . Thus the server  $S_i$  and user  $U_i$  generate the same session key  $SK = SK_u = SK_s$  to encrypt or decrypt the messages transmitted between them.

### 2.4. Password Change Phase

This phase is carried out when the user wants to update his password without connecting the server. Then he should execute the following steps:

- 1)  $U_i$  inserts his smart card  $SC_i$  into the card reader and inputs his identity  $ID_i$  and password  $PW_i$ , so as to request for password changing.
- 2) Next,  $SC_i$  verifies the correctness of  $TD_i$  in the way the

login phase performs. If  $TD_i \neq TD_i'$ , smart card  $SC_i$  rejects the password change request. Only if  $TD_i = TD_i'$  will  $SC_i$  proceeds on.

3) Finally,  $SC_i$  reminds the user  $U_i$  to input the new password  $PW_i^{new}$  and computes the value of  $A_i^{new} = h(ID_i \parallel PW_i^{new}) \oplus a$ ,  $R_i^{new} = h(a \parallel PW_i^{new})$  and  $SC_i$  continues to compute

$$TM_i^{new} = TM_i \oplus R_i \oplus R_i^{new}, TD_i^{new} = h(ID_i \parallel r_i \parallel R_i^{new}).$$

Then  $SC_i$  replaces the stored values  $\{A_i, TD_i, TM_i, TY_i\}$  with  $\{A_i^{new}, TD_i^{new}, TM_i^{new}, TY_i^{new}\}$ .

### III. SECURITY ANALYSIS OF SHI ET AL.'S PROTOCOL

Before analyzing of Shi et al.'s protocol, we first point out that smart card can no longer be deemed as fully tamper-proof device. When a user lost his smart card, the adversary can extract the information stored in the smart card by means of analyzing the power consumption, which has proposed by Kocher et al. [19] and Messerges et al. [20]. In this phase, we illustrate that there exists many security holes in Shi et al.'s scheme and describe them in details.

#### 3.1 Smart Card Loss Attack and Off-line Password Guessing Attack

If the smart card of the user  $U_i$  was stolen by an adversary  $U_\alpha$ , who is also a legal user of the system and has his own smart card  $SC_\alpha$  and suppose  $U_\alpha$  can intercept the login request  $\{CID_i, TG_i, TC_i, TF_i, DS_i, T_1\}$  of  $U_i$ . We point out that Shi et al.'s scheme is vulnerable to offline password guessing attack owing to smart card loss and the procedure is as follows:

**Step 1:**  $U_\alpha$  extracts  $\{A_\alpha, TM_\alpha, TY_\alpha, TD_\alpha, TE_\alpha\}$  from his smart card  $SC_\alpha$  and computes  $a' = A_\alpha \oplus h(ID_\alpha \parallel PW_\alpha)$ ,  $R_\alpha = h(a' \parallel PW_\alpha)$  and  $r_\alpha = TY_\alpha \oplus h(h(ID_\alpha) \parallel x) \oplus R_\alpha$ . Thus  $U_\alpha$  can obtain the system constant value  $h(y \parallel x)$  by computing  $h(y \parallel x) = TE_\alpha \oplus r_\alpha$ .

**Step 2:** As the attacker  $U_\alpha$  obtains the login request  $\{CID_i, TG_i, TC_i, TF_i, DS_i, T_1\}$  of  $U_i$ , he can use these values together with the value  $h(y \parallel x)$  obtained in **Step 1** to compute

$$r_i = TF_i \oplus h(y \parallel x) \parallel T_1, TN_i = TG_i \oplus h(r_i \parallel T_1), TN_i = TG_i \oplus h(r_i \parallel T_1)$$

**Step 3:** When the user's smart card  $SC_i$  was stolen by an adversary  $U_\alpha$ , he can extract the

messages  $\{A_i, TM_i, TY_i, TD_i, TE_i, h(\square)\}$  stored in  $SC_i$ . Then he can obtain the value of  $a$  by using the extracted value  $TM_i$  and the value  $TN_i$ , which was computed in **Step 2**, this is because  $a = TM_i \oplus TN_i$ . Consequently, the attacker  $U_\alpha$  can obtain the hashed value  $h(ID_i \parallel PW_i)$  by calculating  $h(ID_i \parallel PW_i) = A_i \oplus a$ .

**Step 4:** Now  $U_\alpha$  launches offline password guessing attack using the important value  $h(ID_i \parallel PW_i)$ . First,  $U_\alpha$  chooses the candidate identity  $ID_i^*$  and password  $PW_i^*$  from two independent dictionaries respectively.

**Step 5:** The attacker  $U_\alpha$  further computes the value of  $h(ID_i^* \parallel PW_i^*)$  and compare it with  $h(ID_i \parallel PW_i)$ , if they are equal, it indicates that the attacker  $U_\alpha$  has successfully guessed the right identity and password of  $U_i$ . Otherwise,  $U_\alpha$  returns to **Step 4** until he finally seek out the true identity and password of  $U_i$ .

In this way, the attacker can eventually obtain the identity and password of the system's arbitrary user. Hence, Shi et al.'s protocol suffers from smart card loss attack and offline password guessing attack.

#### 3.2 User Impersonation Attack

When the smart card of the legal user was stolen or obtained by the attacker and he had intercepted the login request from the open network then he can launch offline password guessing attack to obtain the identity and password of the user  $U_i$  as we explained in section 3.1. In this case, the attacker  $U_\alpha$  possess the following private values:  $ID_i, PW_i, a, R_i, r_i, TM_i, TN_i$ . We show that he can impersonate  $U_i$  in the following manner without a new smart card:

1) The attacker  $U_\alpha$  acquires the current timestamp  $T_\alpha$  and computes the following values:  $CID_i = h(ID_i) \oplus h(TN_i \parallel r_i \parallel T_\alpha)$ ,  $TG_i = TN_i \oplus h(r_i \parallel T_\alpha)$ ,  $TB_i = TN_i \oplus R_i$  in order to compute  $DS_i$ ,  $TC_i$  and  $TF_i$ , he first compute  $Q_i = h(h(ID_i) \parallel x)$  and next the attacker  $U_\alpha$  can obtain these values by computing  $DS_i = TB_i \oplus Q_i$ ,  $TC_i = h(TN_i \parallel r_i \parallel Q_i \parallel T_\alpha)$ ,  $TF_i = r_i \oplus h(y \parallel x) \parallel T_\alpha$  and transmits the login request  $\{CID_i, TG_i, TC_i, TF_i, DS_i, T_\alpha\}$  to the server.

2) Obviously, the attacker's login request  $\{CID_i, TG_i, TC_i, TF_i, DS_i, T_\alpha\}$  will be accepted by the server because it is computed by using the valid identity  $ID_i$  and password  $PW_i$ . So, in Shi et al.'s protocol, the attacker

$U_\alpha$  can impersonate the legal user  $U_i$  of the system. Therefore, their scheme is vulnerable to user impersonation attack.

### 3.3 Server Spoofing Attack

As described in section 3.1 and section 3.2, the attacker  $U_\alpha$  can obtain the system value  $h(y \| x)$ , which is common for all users. Subsequently, he can get the identity  $ID_i$  and password  $PW_i$  of the user by launching offline password guessing attack then he computes the value of  $a = A_i \oplus h(ID_i \| PW_i)$  and  $R_i = h(a \| PW_i)$  in order to get the value of  $h(h(ID_i) \| x) = TM_i \oplus a \oplus R_i$ . With these values in hand,  $U_\alpha$  can masquerade the legal server simply in the following procedures.

1) At first,  $U_\alpha$  intercepts the login request  $\{CID_i, TG_i, TC_i, TF_i, DS_i, T_1\}$  of  $U_i$  and then computes

$$r_i = TF_i \oplus h(y \| x) \| T_1, TN_i = TG_i \oplus h(r_i \| T_1), TB_i^* = h(h$$

2) After that,  $U_\alpha$  acquires the current time  $T_3$  and calculates  $A_i = h(TB_i^* \| r_i \| T_3)$  and send the response message  $\{A_i, T_3\}$  to  $U_i$ .

3) Since the timestamp  $T_3$  is fresh and that  $U_\alpha$  had successfully guessed the identity  $ID_i$  and password  $PW_i$  of  $U_i$ , the response message  $\{A_i, T_3\}$  will certainly pass the authentication test at the user's side .

In this way, the attacker  $U_\alpha$  can trick the user  $U_i$  by imitating the legal server .

### 3.4 User's Identity is Traceable

Based on the discussion in section 3.1, the attacker has the ability to obtain the identity and password of the system's legal user if he has the smart card of the user and intercepted all messages transmitted in a login-authentication session. That is, the adversary can obtain the identity of the arbitrary user, so we can see that the user's identity is traceable. So, Shi et al.'s scheme could not protect user's privacy as they claimed.

## IV. OUR ENHANCED PROTOCOL

A fresh protocol is proposed in this section, which can resist the attacks described in the previous sections. The proposed scheme has the same four phases like Shi et al.'s scheme. The details of the proposed scheme are shown below.

### 4.1 Registration Phase

Before a user login in the remote server and become the legal user of the system he should execute the following steps as shown in **Figure 2**:

1) Firstly,  $U_i$  chooses his  $ID_i$ ,  $PW_i$  and a random number  $a$  and computes the value of  $R_i = h(PW_i \| a)$ , next transmits the message  $\{ID_i, R_i\}$  to  $S_i$  via a secure channel .

2) Upon receiving the message  $\{ID_i, R_i\}$  from  $U_i$ ,  $S_i$  generates a random number  $r_i$  for the corresponding user  $U_i$  and continues to compute the value of  $TN_i = h(ID_i \| x) \oplus R_i$ ,  $TY_i = r_i \oplus h(ID_i \| x)$ ,  $TD_i = h(ID_i \| r_i \| R_i)$  and  $TE_i = E_x(y \oplus ID_i) \oplus r_i$ . Then the server  $S_i$  keeps  $\{TY_i, TD_i, TE_i, h(\square)\}$  into the smart card  $SC_i$  and delivers  $\{SC_i, TN_i\}$  to  $U_i$ .

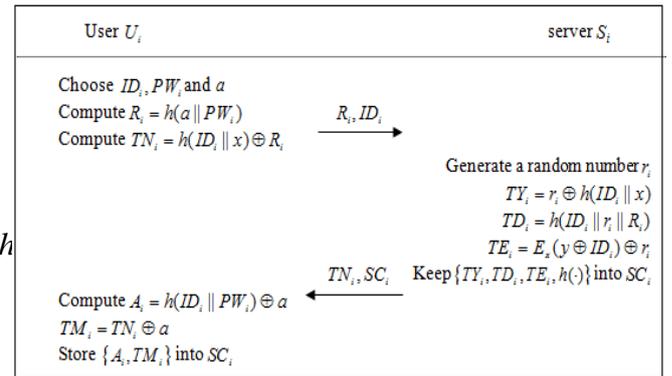


Figure 2. The Registration Phase of The Proposed Scheme

### 4.2. Login Phase

When a registered user  $U_i$  would like to login into the server  $S_i$  and access the services,  $U_i$  inserts smart card  $SC_i$  into a terminal device and inputs his identity  $ID_i$  and password  $PW_i$  then  $SC_i$  performs the following steps which are shown in **Figure 3**:

1) Firstly, the smart card  $SC_i$  successively computes the following values:  $a = A_i \oplus h(ID_i \| PW_i)$ ,  $R_i = h(a \| PW_i)$ ,  $h(ID_i \| x) = TM_i \oplus a \oplus R_i$  and computes the value of  $r_i = TY_i \oplus h(ID_i \| x)$ ,  $TD_i' = h(ID_i \| r_i \| R_i)$ .

2) If  $TD_i' = TD_i$ ,  $SC_i$  continues to calculate the value  $E_x(y \oplus ID_i) = TE_i \oplus r_i$ , we write  $TF_i = E_x(y \oplus ID_i)$  for simplicity. Otherwise,  $SC_i$  drops the session.

3) Then,  $SC_i$  acquires the current timestamp  $T_1$  and computes the user's dynamic identity  $CID_i = h(ID_i \| T_1) \oplus r_i$ ,  $TG_i = TN_i \oplus h(r_i \| T_1)$ ,  $TB_i = TN_i \oplus R_i$ . After that, the smart card  $SC_i$  generates a random number  $r$  and further computes the value of  $Q_i = h(ID_i \| r)$ ,  $DS_i = TB_i \oplus Q_i$ ,  $TC_i = h(TN_i \| r_i \| Q_i \| T_1)$ , then  $SC_i$  sends the

login request  $\{TF_i, CID_i, TG_i, DS_i, TC_i, T_1\}$  to  $S_i$ .

### 4.3. Authentication Phase

In this phase, the user and server take the following steps to achieve mutual authentication and further consult the common session key.

- 1) On receiving the login request  $\{TF_i, CID_i, TG_i, DS_i, TC_i, T_1\}$ ,  $S_i$  obtains the current time  $T_2$  and verifies the validity of  $T_1$ . Only when  $T_1$  is fresh will the server  $S_i$  continue further. Otherwise,  $S_i$  rejects all the login requests. Subsequently  $S_i$  decrypts  $TF_i$  with the secret key  $x$  then he can obtain the user's identity and  $r_i = CID_i \oplus h(ID_i \parallel T_1)$ ,  $TN_i = TG_i \oplus h(r_i \parallel T_1)$ ,  $TB_i^* = h(ID_i \parallel x)$ ,  $Q_i = DS_i \oplus TB_i^*$  so  $S_i$  can compute the value of  $TC_i' = h(TN_i \parallel r_i \parallel Q_i^* \parallel T_1)$ .
- 2) Next,  $S_i$  checks whether the computed  $TC_i'$  and the stored  $TC_i$  are equal or not. If not,  $S_i$  drops the

session. Only if  $TC_i' = TC_i$  will the user  $U_i$  be authenticated and the session proceeds further.

- 3)  $S_i$  acquires the current time  $T_3$  to calculate  $V_i = h(TB_i^* \parallel r_i \parallel T_3)$  and delivers the message  $\{V_i, T_3\}$  to  $U_i$ .
- 4) Upon receiving the response message  $\{V_i, T_3\}$ ,  $SC_i$  checks  $T_3$  for freshness. If  $T_3$  is fresh,  $SC_i$  computes  $V_i' = h(TB_i \parallel r_i \parallel T_3)$  and verifies whether the equation  $V_i' = V_i$  holds or not. If so,  $U_i$  authenticates  $S_i$  as a legal server or else  $SC_i$  stops the procedure and neglect the response message.

Finally,  $S_i$  computes his session key  $SK_u = h(TB_i^* \parallel r_i \parallel T_1 \parallel T_2 \parallel Q_i^*)$  and smart card  $SC_i$  computes  $SK_s = h(TB_i \parallel r_i \parallel T_1 \parallel T_3 \parallel Q_i)$ . Hence, they have negotiated the common session key  $SK = SK_u = SK_s = h(TB_i \parallel r_i \parallel T_1 \parallel T_3 \parallel Q_i)$ .

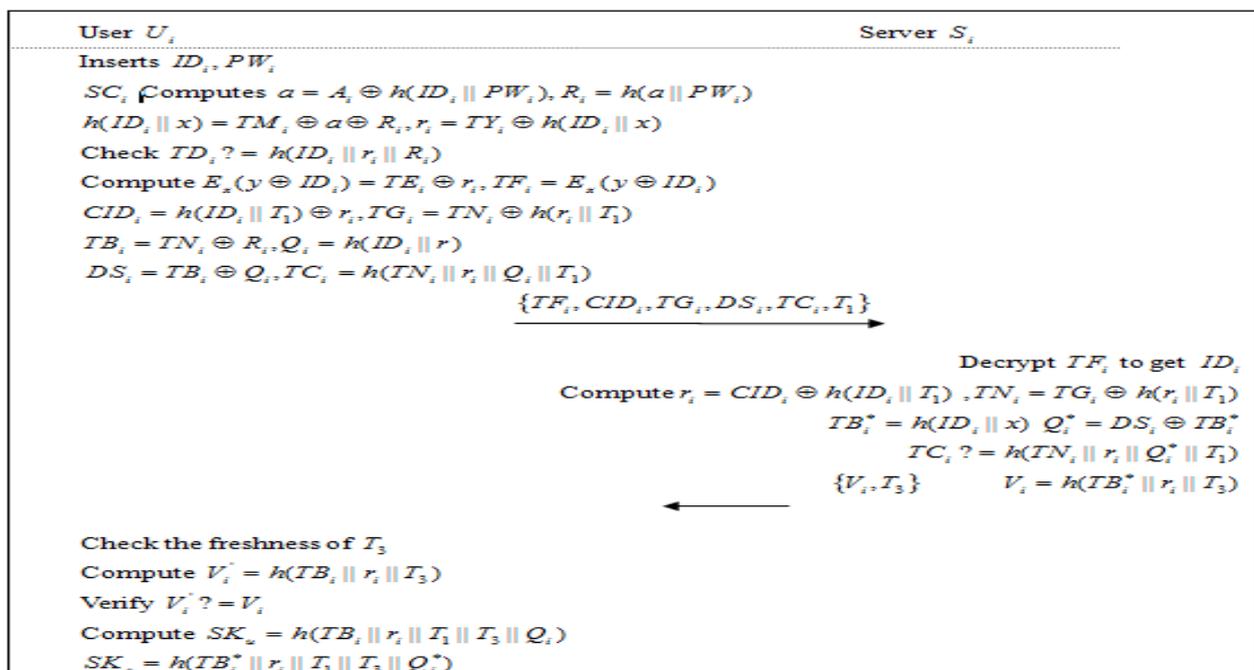


Figure 3. Login and Authentication Phase

### 4.4. Password Change Phase

When the user wants to update his password without connecting the server. Then he should perform the following steps:

- 1) Firstly,  $U_i$  inserts his smart card into the card reader and inputs his identity and password to request for changing his password.
- 2) Next,  $SC_i$  verifies the correctness of  $TD_i$  in the way the login phase performs. If  $TD_i \neq TD_i'$ ,  $SC_i$  drops the password change request. But after thrice failures  $SC_i$  will get blocked and the user must enter the private

unblocking key to re-activate his smart card. Only if  $TD_i = TD_i'$  will  $SC_i$  proceed on.

- 3) Then  $SC_i$  reminds  $U_i$  to input the new password  $PW_i^{new}$  and computes the value of  $A_i^{new} = h(ID_i \parallel PW_i^{new}) \oplus a$ ,  $R_i^{new} = h(a \parallel PW_i^{new})$ ,  $TM_i^{new} = TM_i \oplus R_i \oplus R_i^{new}$  and computes  $TD_i^{new} = h(ID_i \parallel r_i \parallel R_i^{new})$ . Finally,  $SC_i$  stores  $A_i^{new}$ ,  $TD_i^{new}$  and  $TM_i^{new}$  in place of  $A_i$ ,  $TD_i$  and  $TM_i$  respectively.

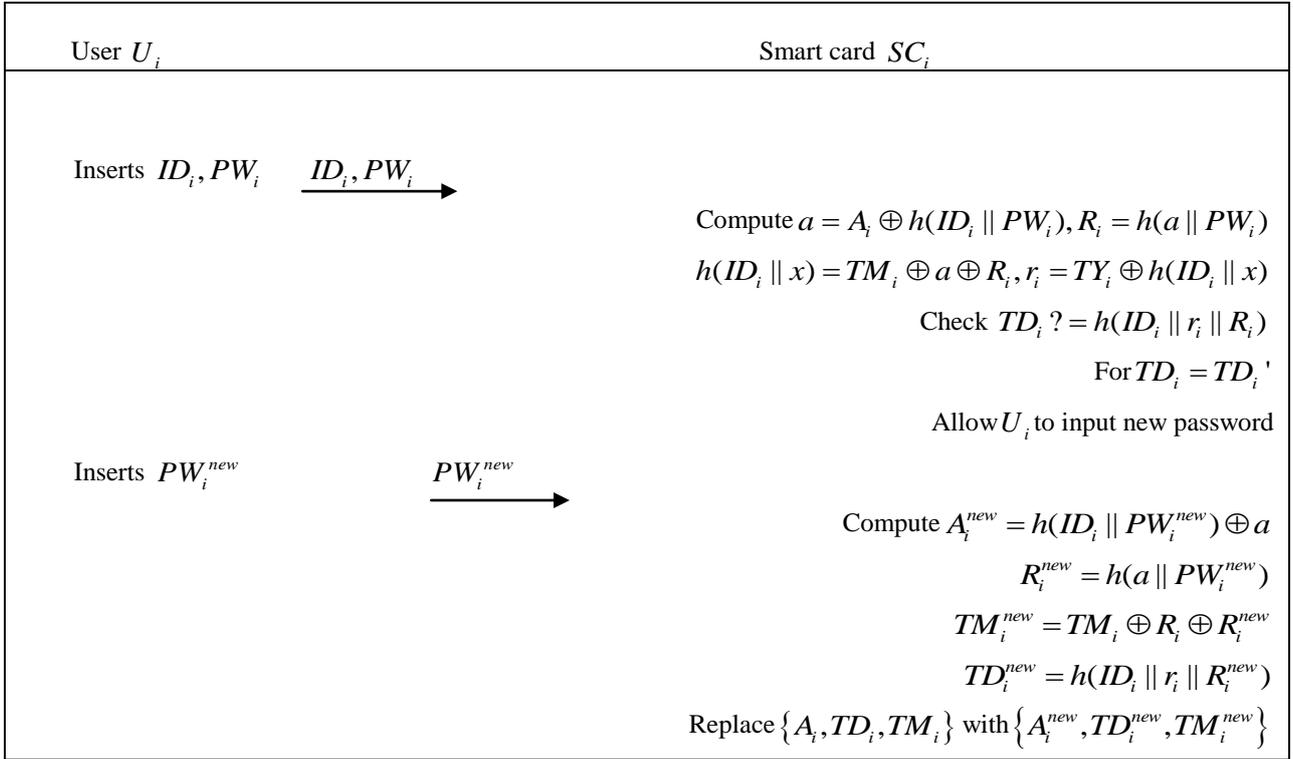


Figure 4. Password Change Phase

## V. SECURITY ANALYSIS

In this section, we will demonstrate that our enhanced protocol is secure against various attacks discussed in the previous sections, the details are shown in the following content:

### 5.1. Resists replay attack

Replay attack is an offensive action through which an attacker may impersonate the legal user or the server by repaying the previous message. In our protocol, we assume that the attacker had intercepted the previous authentication information and we use timestamp  $T_1$  to prevent the attacker from repaying the request message and likewise  $T_3$  is used to resist the attacker reusing the response message to imitate the valid server. This is because both the request and response message undergo the timestamp freshness check through the similar freshness verification process at each other's side. Therefore, our protocol can withstand replay attack.

### 5.2. Resists offline password guessing attack

Off-line password guessing attack means that the attacker can use user's smart card and the interactive information between the legal user and the server to successfully guess the user's password off-line. In this scheme, consider that an attacker obtains a user's smart card, either by stealing or lost by the user. Then he can extract all secret values  $\{A_i, TM_i, TY_i, TD_i, TE_i\}$  stored in  $SC_i$ . Among these values:

$$A_i = h(ID_i \parallel PW_i) \oplus a, \quad TM_i = h(ID_i \parallel x) \oplus R_i \oplus a, \quad TY_i = r_i \oplus h(ID_i \parallel x),$$

$TD_i = h(ID_i \parallel r_i \parallel R_i)$  and  $TE_i = E_x(y \oplus ID_i) \oplus r_i$ , due to the property of one way hash function, we cannot retrieve the values  $\{ID_i, r_i, R_i\}$  out of  $TD_i$ . And in order to obtain the value  $h(ID_i \parallel x)$  from  $TM_i$ , the attacker  $U_\alpha$  should know the random number  $a$  and the password  $PW_i$  of  $U_i$ . But he cannot guess the two values  $\{a, PW_i\}$  at the same time. And as  $A_i$  contains another unknown value  $ID_i$ ,  $U_\alpha$  could not retrieve these values from  $A_i$  either. As a result,  $U_\alpha$  is unable to compute the value  $r_i = TY_i \oplus h(ID_i \parallel x)$ . In a word,  $U_\alpha$  cannot obtain these values  $\{x, r_i, h(ID_i \parallel x)\}$  and without knowing these values it is impossible for him to guess an arbitrary password  $PW_i^*$  and verify his guess using these five values:  $A_i, TM_i, TY_i, TD_i, TE_i$ . Next, suppose that  $U_\alpha$  have eavesdropped the login request  $\{TF_i, CID_i, TG_i, DS_i, TC_i, T_1\}$  of  $U_i$ , we show that he still cannot using these values to verify his guess. From the login phase, we can obtain the value  $TF_i = E_x(y \oplus ID_i)$ ,  $CID_i = h(ID_i \parallel T_1) \oplus r_i$ ,  $TG_i = TN_i \oplus h(r_i \parallel T_1)$ ,  $TB_i = TN_i \oplus R_i$ ,  $Q_i = h(ID_i \parallel r)$  and successively we can get by calculating the value of:  $DS_i = TB_i \oplus Q_i$ ,  $TC_i = h(TN_i \parallel r_i \parallel Q_i \parallel T_1)$ . Similarly,  $TN_i = h(ID_i \parallel x) \oplus R_i$  and  $R_i = h(a \parallel PW_i)$  can be obtained from the registration

phase. So, we have  $TG_i = h(ID_i || x) \oplus h(a || PW_i) \oplus h(r_i || T_1)$  and  $TB_i = h(ID_i || x)$  and likewise we have  $DS_i = h(ID_i || x) \oplus h(ID_i || r)$ ,  $TC_i = h(TN_i || r_i || h(ID_i || r) || T_1)$ . In each item of  $\{TF_i, CID_i, TG_i, DS_i, TC_i, T_1\}$ , it contains at least two unknown values for the attacker  $U_\alpha$  and it is not possible for him to guess two unknown values in polynomial time.

Thus, the proposed protocol is secure against the most damaging attack: offline password guessing attack.

### 5.3. Resists privileged insider attack

Privileged insider attack happens when an insider of the system like the administrator obtains the password of the legal user by monitoring the registration message transmitted from the user to the server through a secure channel. In the proposed scheme the user use a randomly selected number  $a$  and submits hashed value  $R_i = h(a || PW_i)$  to protect his password instead of sending it in a plain text. As the attacker doesn't know the random number  $a$  and the password  $PW_i$ , it is impossible for him to simultaneously guess two values in polynomial time. Thus, it is not hard to see that the proposed protocol is secure against privileged insider attack.

### 5.4. Resists user impersonation and server masquerading attack

In the proposed scheme, if an attacker would like to impersonate  $U_i$ , he should compute the login request  $\{TF_i, CID_i, TG_i, DS_i, TC_i, T_1\}$  to pass the server's verification. Without knowing the identity  $ID_i$  and password  $PW_i$  of  $U_i$ ,  $U_\alpha$  cannot calculate the random number  $a$  not to mention  $R_i$  and  $h(ID_i || x)$  even if he has the smart card of  $U_i$ . And he cannot synchronously guess the possible identity and password of the user because he has no option to verify his guess. Thus, it is not feasible for  $U_\alpha$  to launch user impersonation attack. Similarly, assume that the attacker  $U_\alpha$  intercepted the login request  $\{TF_i, CID_i, TG_i, DS_i, TC_i, T_1\}$  of  $U_i$ . In order to successfully impersonate a legal server, he should compute a valid response message  $\{r_i, T_3\}$  as an answer to the login request. From the discussion above,  $U_\alpha$  cannot compute  $h(ID_i || x)$  and he doesn't know the secret key  $x$  of the server so he cannot decrypt  $TF_i$  to get  $ID_i$ , therefore he cannot calculate  $r_i = CID_i \oplus h(ID_i || T_1)$ . Based on the

discussion above, it is not feasible for an attacker to launch server masquerading attack on the proposed scheme.

### 5.5. Resists smart card loss attack

Suppose an attacker obtains the smart card of the user and eavesdrops the information transmitted between the two sides from the open network. Section 4.2 shows that the attacker cannot obtain any useful information such as  $\{x, r_i, h(ID_i || x)\}$  even if he has stolen the smart card of  $U_i$  and intercepted all messages transmitted in a login-authentication session. Hence, a lost or stolen smart card is helpless for  $U_\alpha$  to obtain the private information of the user. Consequently, the security of the proposed protocol remains unaffected to smart card loss attack.

### 5.6. Mutual authentication

In our protocol, the server ensures that the login request is from the legal user by means of checking whether  $TC_i$  is equal to  $h(TN_i || r_i || Q_i^* || T_1)$  or not after accepting the login request depending upon the freshness of the timestamp  $T_1$ . Likewise, the user authenticates the server by means of verifying whether the equation  $V_i' = V_i$  holds or not after accepting the response request depending on the freshness of the timestamp  $T_3$ . Therefore, the proposed protocol provides secure mutual authentication between the legal user and the valid server.

## VI. PERFORMANCE COMPARISON AND FUNCTIONALITY ANALYSIS

Efficiency and functionality comparisons among the proposed protocol and the related three protocols: Shi et al.'s [18], Kumari et al.'s [19], Chang et al.'s [20] are shown in this section. Table 1 shows the efficiency comparison among the four protocols and Table 2 shows the functionality comparison among these schemes.

In **Table 1**, each scheme consists of five parts: registration phase, login phase, authentication phase, password change phase and the sum of computational complexity.

For convenience, some notations which will be used in the efficiency comparison are given in the following:

- $t_H$ : the time complexity for one way hash function ;
- $t_E$ : the time complexity for symmetric encryption;
- $t_{XOR}$ : the time complexity for XOR operation ;

**Table 1. Computational Cost**

Computational cost comparison	Ours	Shi et al.	Kumari et al.	Chang et al.
Registration phase				

## An enhanced dynamic-ID-based remote user authentication protocol with smart card

	$6t_{XOR} + 5t_H + t_E$	$6t_{XOR} + 8t_H$	$5t_{XOR} + 5t_H$	$t_{XOR} + t_H$
Login phase	$10t_{XOR} + 9t_H + t_E$	$11t_{XOR} + 12t_H$	$10t_{XOR} + 8t_H$	$3t_{XOR} + 4t_H$
Authentication phase	$3t_{XOR} + 6t_H + t_E$	$4t_{XOR} + 8t_H$	$3t_{XOR} + 7t_H$	$6t_{XOR} + 10t_H$
Password change phase	$7t_{XOR} + 6t_H$	$10t_{XOR} + 7t_H$	$6t_{XOR} + 6t_H$	$8t_{XOR} + 12t_H$
Sum of computational complexity	$26t_{XOR} + 26t_H + 3t_E$	$31t_{XOR} + 35t_H$	$24t_{XOR} + 26t_H$	$18t_{XOR} + 27t_H$

**Table 1** shows that the proposed agreement takes more computations cost than those of others due to the application of the symmetrical encryption technique. But if merely hash arithmetic and XOR operation are introduced in a scheme, it is unable to guarantee the safety of the scheme. As we can clearly see from the **Table 1** that the other three protocols, which are designed only with one way hash function and the

XOR operation, are prone to various attack such as smart card loss attack, offline password guessing attack, user impersonation attack and server masquerading attack. Thanks to the adoption of symmetrical encryption algorithm, the proposed scheme can resist the above attacks and provide perfect user anonymity. Hence, our protocol is suitable for the application environment.

**Table 2. Security Comparisonom**

Security comparison	Ours	Shi et al.	Kumari et al.	Chang et al.
Resisting insider attack	Y	Y	Y	N
Resisting smart card loss attack	Y	N	N	N
Resisting impersonation attack	Y	N	N	N
Resisting server spoofing attack	Y	N	N	N
Offline password guessing attack	Y	N	N	N
Offline password guessing attack	Y	N	N	N
Resisting replay attack	Y	Y	Y	Y
Providing mutual authentication	Y	N	N	N
Providing mutual authentication	Y	N	N	N

From **Table 2**, it is obvious to see that our protocol has many important secure properties. Compared with the other three related works, our scheme is secure against insider attack, smart card loss attack, impersonation attack, server spoofing attack, password guessing attack and replay attack. Furthermore, the proposed scheme can provide mutual authentication and user anonymity.

### VII. CONCLUSION

In this paper, we first briefly reviewed an improved anonymous remote user authentication scheme based on

dynamic identity. But after basic secure analysis of Shi et al.'s protocol, we found that their scheme is still subject to various attacks and lack of user anonymity. To overcome the weakness of Shi et al. protocol, we presented an enhanced

dynamic-ID-based remote user authentication scheme with smart card. Through security analysis and performance comparison, we have illustrated that with little increase in computational cost, the proposed key agreement can resist various attacks and provide perfect user anonymity.

### REFERENCES

- [1] Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM*, 24, 770–772.
- [2] N. Haller, 'The S/KEY one-time password system, *Proceedings of the ISOC Symposium on Network and Distributed System Security*, 1994, pp. 151–157.
- [3] Peyravian, M. (2000). Methods for protecting password transmission. *Computers & Security*, 19(5), 466–469.
- [4] Y. L. Tang, M. S. Hwang and C. C. Lee, "A simple remote user authentication scheme", *Mathematical and Computer Modeling*, vol. 36, Issues 1-2, (2002), pp. 103–107.

- [5] Lin, C. (2003).A password authentication scheme with secure password updating. *Computers & Security*,22(1), 68–72.
- [6] C. L. Hsu, “Security of Chien, et al., “Remote user authentication scheme using smart cards”, *Computer Standards and Interfaces*, vol. 26, no. 3 , (2004), pp. 167-169.
- [7] W. C. Ku and S. M. Chen, “Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards”, *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, (2004), pp. 204-207.
- [8] Y. F. Chang and C. C. Chang, “Authentication schemes with no verification Table”, *Applied Mathematics and Computation*, vol. 167, no. 2, (2005), pp. 820–832.
- [9] He,D., Chen, J., & Hu, J. (2011).Further improvement of Juang et al.’s password-authenticated key agreement scheme using smart cards. *Kuwait Journal of Science and Engineering*, 38(2A), 55–68.
- [10] Juang, W., Chen, S., & Liaw, H.(2008)Robust and efficient password-authenticated key agreement using smart cards.*IEEE Transactions on Industrial Electronics*, 55(6), 2551–2556.
- [11] Yang, G., et al. (2008).Two-factor mutual authentication based on smart cards and passwords. *Journal of Computer and System Sciences*, 74(7), 1160–1172.
- [12] Xu, J., Zhu, W., & Feng, D. (2009). An improved smart card based password authentication schemewith provable security. *Computer Standards & Interfaces*, 31(4), 723–728.
- [13] Yeh, K., et al. (2010).Two robust remote user authentication protocols using smart cards. *Journal of Systems and Software*, 83(12), 2556–2565.
- [14] M. Das, A. Saxena, V. Gulati, A dynamic id-based remote user authentication scheme, *IEEE Trans. Consum.Electron.* 50 (2) (2004) 629–631.
- [15] Y. Wang, J. Liu, F. Xiao, J. Dan.A more efficient and secure dynamic id-based remote user authentication scheme, *Comp. Commun.Syst* 32 (4) (2009) 583–585.
- [16] Chang YF, Tai WL, Chang HC. Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update.*Int J Commun.Syst* 2013.
- [17] S. Kumari, M. K. Khan and X. Li,“An improved remote user authentication scheme with key agreement”, *Computers and Electrical Engineering*, vol. 40, no. 6, (2014), pp. 1997–2012.
- [18] Y. Shi, H. Shen, Y. Y. Z and J. H. Chen, ”An improved anonymous remote user authentication scheme with key agreement based on dynamic identity”, *InternationalJournal of Security and Its Applications*,Vol.9,No.5(2015), pp.255-268.
- [19] P. Kocher, J. Jaffe and B. Jun,“Differential power analysis”, *Proceedings of advances in cryptology CRYPTO’99*, (1999), pp. 388–97.
- [20] T. S. Messerges, E. A. Dabbish and R. H. Sloan, “Examining smart-card security under the threat of power analysis attacks”, *IEEE Trans Comp*, vol. 51, no. 5, (2002), pp. 541-52