International Journal of Computer Applications (0975 - 8887)
Volume 69 - No. 22, May 2013

An Improved Efficient Remote User Authentication Scheme in Multi-server Environment using Smart Card

Ruhul Amin
Haldia Institute of Technology
Haldia, West Bengal-721657, India

Tanmoy Maitra
Haldia Institute of Technology
Haldia, West Bengal-721657, India

Debasis Giri
Haldia Institute of Technology
Haldia, West Bengal-721657, India

ABSTRACT

In a single server environment, one server is responsible for providing services to all the authorized remote users. However, the problem arises if a user wishes to access several network services. To overcome this weakness, various multi-server authentication schemes have been proposed. In 2012, Taygi et al. [1] proposed a scheme for multi-server environment. But it is found that their proposed scheme is insecure against user impersonation attack, server masquerading attack, collaboration attack between a valid user and a server, smart card stolen attack, password guessing attack and password change attack. Then we propose an improved scheme can overcome possible attacks and also provides better computational cost as well as communication cost than related schemes published earlier.

Keywords:

Attack, Authentication, Password, Smart Card

1. INTRODUCTION

In the client/server environment password-based authentication scheme with smart card is widely used in remote user authentication. Li et al. [2] first proposed authentication scheme for multi-server architecture based on neural networks. They had shown that their scheme allows users to choose the password freely. Lin et al. [3] found that Li et al.'s scheme takes long time on training neural networks and they proposed an improved scheme using ElGamal digital signature [4] and geometric properties on the Euclidean plane. Cao and Zhong [5] proved that there is impersonation attack and every users have to store large amount of public parameters in memory of smart card for authentication in Lin et al.'s scheme. Juang [6] proposed a scheme based on nonce, one-way hash function and symmetric cryptosystem, which is efficient with an additional trait of generating session key. Neverless, Ku et al. [7] proved that Juang's scheme is exposed to insider attack and fails to provide forward secrecy. Chang and Lee [8] proposed an improvement scheme over Juang's scheme and claimed that their scheme is able to resist stolen-verified attack, server spoofing attack, smart card loss attack, reply attack and provides mutual authentication and forward secrecy. Liao and Wang [9] proposed a dynamic ID-based remote user authentication scheme based on one way hash function. But Chen et al. [10] pointed out that Liao-Wang's scheme does not provide forward secrecy. Hsiang and Shih [11] found that Liao-Wang's scheme is insecure against insider attack, impersonation attack, server spoofing attack and shows inadequacy in providing mutual authentication. To remove these drawbacks, they also proposed an improved scheme. Though, Sood et al. [12] pointed out that Hsiang-Shih's improved scheme is insecured against reply attack, impersonation attack, stolen smart card attack. Tsai [13] suggested a nonce based scheme based on one way hash function. However, Zhu [14] proved that Tsai's scheme is vulnerable to denning-sacco [15] attack, server spoofing attack and does not provide perfect forward secrecy. To overcome these drawbacks, they proposed an improved scheme.

In 2012, Taygi et al. [1] proposed a scheme based on nonce and one way hash function. In this paper, it is demonstrated that this scheme is insecure against user impersonation attack, server masquerading attack, collaboration attack between a valid user and a server, smart card stolen attack, password guessing attack and password change attack. To overcome these weakness we propose the improvement of their scheme. In addition, proposed scheme also provides better computational cost as well as communication cost than the related scheme published earlier.

The remainder of this paper is organized as follows: Section 2 briefly reviews the Taygi et al.'s [1] scheme. Section 3 shows the brief description of attacks on Taygi et al.'s [1] scheme. Section 4 describes the proposed scheme which withstand the weaknesses of Taygi et al.'s [1] scheme. Section 5 describes cryptanalysis of proposed improved scheme and section 6 compares the performances of proposed scheme with previously published scheme. Conclusion of this paper appears in section 7. Finally References are given in section 8.

2. BRIEF REVIEW OF TAYGI ET AL.'S SCHEME

This section presents briefly description of Taygi et al.'s [1] remote user authentication scheme in multi server environment using smart card. The notations used throughout this paper are summarized as follows:

RC	$\longrightarrow$	registration center
U_i	$\longrightarrow$	$i-th$ remote user
ID_i	$\longrightarrow$	identity of U_i

Their scheme consists of following four phases: Registration phase, Login phase, Authentication phase and Password Change phase.

2.1 Registration Phase

This phase is divided into two sub-phases: Server Registration phase and User Registration phase.

2.1.1 Server Registration Phase

In this phase, S_j selects SID_j and submits it to RC over a secure channel. After receiving it, RC computes the server secret parameter $SS_j = (g^{h(SID_j,\, h(x))} \ mod\ p) \oplus h(d)$ and sends $\{SS_j,\ h(d)\}$ to S_j over a secure channel.

PW_i	$\longrightarrow$	password chosen by U_i
PW_i^*	$\longrightarrow$	password guessed by an adversary
S_j	$\longrightarrow$	$j-th$ authentication server $(1 \leq j \leq n)$
SID_j	$\longrightarrow$	identity of S_j
x	$\longrightarrow$	secret key of RC
d	$\longrightarrow$	secret number of RC
p	$\longrightarrow$	large prime number
g	$\longrightarrow$	primitive element over $GF(p)$
$h(.)$	$\longrightarrow$	cryptographic one way hash function
$\oplus$	$\longrightarrow$	bitwise XOR operation
$SKey_{ij}$	$\longrightarrow$	session key shared between U_i and S_j
N_1	$\longrightarrow$	random nonce generated by U_i
k	$\longrightarrow$	Secret number chosen by U_i
N_2	$\longrightarrow$	random nonce generated by S_j
n_i	$\longrightarrow$	random nonce generated by RC for each user
w_j	$\longrightarrow$	random nonce generated by RC for each server

2.1.2 User Registration Phase

U_i selects ID_i and PW_i, computes $h(PW_i)$ and submits $\{ID_i,\ h(PW_i)\}$ to RC over a secure channel. After receiving registration request, RC computes $x_i = (g^{h(PW_i)}\ mod\ p) \oplus h(x)$, $y_i = h(ID_i,\ h(d))$, $z_i = y_i \oplus h(PW_i)$ and issues a smart card for U_i by storing $\{x_i,\ y_i,\ z_i,\ p,\ g,\ h(\cdot)\}$ for $(i = 1\ to\ n)$ into memory of smart card.

2.2 Login Phase

U_i inserts his/her smart card to the card reader and then supply ID_i and PW_i. The smart card computes $z_i^{'} = y_i \oplus h(PW_i^{'})$ and checks whether computed $z_i^{'}$ equals stored z_i or not. If equal, smart card generates a random nonce N_1, computes $a_i = g^{y_i}\ mod\ p$, $b_i = {a_i}^{y_i \times N_1}\ mod\ p$, $c_i = {a_i}^{h(PW_i) \times N_1}\ mod\ p$, $d_i = g^{h(PW_i)}\ mod\ p$, $Q_j = g^{h(SID_j,\ (x_i \oplus d_i))}\ mod\ p$, $e_i = (h(PW_i) + y_i \times h(ID_i,\ a_i,\ b_i,\ c_i,\ d_i,\ N_1,\ Q_j))\ mod\ (p-1)$ and sends the login request message $\{ID_i,\ SID_j,\ d_i,\ e_i,\ N_1\}$ to S_j.

2.3 Authentication phase

After receiving the login request message $\{ID_i,\ SID_j,\ d_i,\ e_i,\ N_1\}$, S_j first checks the validity of ID_i. If valid, S_j computes $y_i = h(ID_i,\ h(d))$, $a_i = g^{y_i}\ mod\ p$, $b_i = {a_i}^{y_i \times N_1}\ mod\ p$, $c_i = {d_i}^{y_i \times N_1}\ mod\ p$, $Q_j = SS_j \oplus h(d)$ and checks whether $g^{e_i} = d_i \times {a_i}^{h(ID_i,\ a_i,\ b_i,\ c_i,\ d_i,\ N_1,\ Q_j)}\ mod\ p$ is true. If it is true, S_j further checks whether ${a_i}^{e_i \times N_1} = c_i \times {b_i}^{h(ID_i,\ a_i,\ b_i,\ c_i,\ d_i,\ N_1,\ Q_j)}\ mod\ p$ is true. If it holds good, S_j generates a nonce N_2, computes $X_1 = y_i \oplus N_1 \oplus N_2$, $X_2 = {Q_j}^{N_2}\ mod\ p$ and sends the message $\{ID_i,\ X_1,\ X_2\}$ to U_i. After getting the message $\{ID_i,\ X_1,\ X_2\}$ from S_j, U_i computes $N_2 = y_i \oplus X_1 \oplus N_1$, $X_2^{'} = {Q_j}^{N_2}\ mod\ p$ and checks whether X_2 and $X_2^{'}$ are equal. If equality holds, S_j is authentic, otherwise terminate the session. Subsequently, U_i computes $X_3 = {Q_j}^{N_1 \times N_2}\ mod\ p$ and sends $\{ID_i,\ X_3\}$ to S_j. Once the message $\{ID_i,\ X_3\}$ is received, S_j computes $X_3^{'} = {Q_j}^{N_1 \times N_2}\ mod\ p$ and checks whether X_3 and $X_3^{'}$ are equal. If equality holds, mutual authentication is achieved. Both the parties agree upon a common shared session key $SKey_{ij} = h(ID_i,\ SID_j,\ Q_j,\ N_1,\ N_2)$.

2.4 Password Change phase

If user U_i wants to change his/her password, U_i inserts the smart card to the card reader and submits ID_i and $PW_i^{'}$. The reader computes $z_i^{'} = y_i \oplus h(PW_i^{'})$ and checks whether computed $z_i^{'}$ equals stored z_i. If equal, U_i enters a new password PW_i^{new}. The card reader computes $z_i^{new} = y_i \oplus h(PW_i^{new})$, $x_i^{new} = x_i \oplus g^{h(PW_i)} \oplus g^{h(PW_i^{new})}\ mod\ p$ and stores z_i^{new}, x_i^{new} instead of z_i, x_i respectively in the memory of his/her smart card.

3. SECURITY ANALYSIS OF TAYGI ET AL.'S SCHEME

In this section, we will analyze the security of Taygi et al.'s [1] remote user authentication scheme in multi-server environment using smart card. To analyze the security weaknesses, it can be assumed that an attacker could obtain the secret values stored in the smart card by monitoring the power consumption [16][17] and intercepting messages communicating between the user and the server. Under this assumption, it will be discussed the various attacks, such as the user impersonation attack, server masquerading attack, collaboration attack between a valid user and a server, smart card stolen attack and password guessing attack on Taygi et al.'s [1] scheme. In addition, Taygi et al.'s [1] scheme does not provide mutual authentication between user and authentication server.

3.1 User Impersonation Attack

If an attacker can obtain the secret values (x_i, y_i, z_i) from the user's smart card illegally by some means and intercept the message $\{ID_i,\ SID_j,\ d_i,\ e_i,\ N_1\}$ in the login phase, the attacker can perform the user impersonation attack as the following steps:

Step 1: Attacker computes which is equal to $h(PW_i) = z_i \oplus y_i$

Step 2: The attacker generates a random nonce N^* and compute $a_i = g^{y_i}\ mod\ p$, $b_i^* = {a_i}^{y_i \times N^*}\ mod\ p$, $c_i^* = {a_i}^{h(PW_i) \times N^*}\ mod\ p$, $d_i = g^{h(PW_i)}\ mod\ p$ and $Q_j = g^{h(SID_j,\ (x_i \oplus d_i))}\ mod\ p$, $e_i^* = (h(PW_i) + y_i \times h(ID_i,\ a_i,\ b_i^*,\ c_i^*,\ d_i,\ N^*,\ Q_j))\ mod\ (p-1)$ then attacker sends $msg = \{ID_i,\ SID_j,\ d_i,\ e_i^*,\ N^*\}$ to the authenticated server S_j.

Step 3: After receiving the forged message msg, S_j checks the validity of ID_i and compute $y_i = h(ID_i, h(d))$, $a_i = g^{y_i}\ mod\ p$, $b_i^* = {a_i}^{y_i \times N^*}\ mod\ p$, $c_i^* = {d_i}^{y_i \times N^*}\ mod\ p$, $Q_j = SS_j \oplus h(d)$. Then verifies whether $g^{e_i^*} = d_i \times {a_i}^{h(ID_i,\ a_i,\ b_i^*,\ c_i^*,\ d_i,\ N^*,\ Q_j)}\ mod\ p$ and ${a_i}^{e_i^* \times N^*} = c_i^* \times {b_i^*}^{h(ID_i,\ a_i,\ b_i^*,\ c_i^*,\ d_i,\ N^*,\ Q_j)}\ mod\ p$. It can be shown that both are equal then authenticated server will be convinced the message sent from the legal user.

Step 4: After choosing a random nonce N_2, authenticated server S_j makes the reply message $\{ID_i, X_1, X_2\}$ by computing $X_1 = y_i \oplus N^* \oplus N_2$, $X_2 = {Q_j}^{N_2}\ mod\ p$ in the authenticated phase.

3.2 Server Masquerading Attack

After obtaining the secret values (x_i, y_i, z_i) from the user's smart card, further the attacker intercept the message $\{ID_i,\ SID_j,\ d_i,\ e_i,\ N_1\}$ and (X_1, X_2) in the login phase and authentication phase respectively, then the attacker can perform the Server masquerading attack as the following steps:

Step 1: The attacker computes $Q_j = g^{h(SID_j,\ (x_i \oplus d_i))}\ mod\ p$). Then it chooses a random number r and again compute $X_1 = y_i \oplus N_1 \oplus r$, $X_2 = {Q_j}^r\ mod\ p$ and sends $\{ID_i,\ X_1,\ X_2\}$ to the user U_i.

Step 2: After receiving the forged message, user U_i compute $r = y_i \oplus X_1 \oplus N_1$, $X_2^{'} = {Q_j}^r\ mod\ p$ and verifies whether $X_2 = X_2^{'}$. It can be easily verified that $X_2 = X_2^{'}$.

Step 3: Then user U_i makes the reply message $\{ID_i, X_3\}$ by computing $X_3 = {Q_j}^{N_1 \times r}\ mod\ p$.

3.3 Collaboration Attack between a Valid User and a Server

In Taygi et al.'s [1] scheme a valid user U_i can compute $h(x)$ = $x_i \oplus d_i$ where d_i = $g^{h(PW_i)}\ mod\ p$ and x_i is stored in smart card of user U_i. Now let valid user U_i provides this computed $h(x)$ to an another authenticated server S_m. Then S_m can easily compute the secret key of another authenticated server S_j, by computing SS_j = $(g^{h(SID_j,\ h(x))}\ mod\ p) \oplus h(d)$, where SID_j is known from login message and $h(d)$ is known to all authenticated servers. Suppose another valid user U_k is trying to communicate securely to authenticated server S_j. After sending a valid login message to S_j, another authenticated server S_m can verify the user U_k because S_m can compute the secret key of S_j. If the server S_m sends forge message $\{ID_k,\ X_{1m},\ X_{2m}\}$ earlier than S_j to the user U_k by computing X_{1m} = $y_k \oplus N_1 \oplus N_3$, X_{2m} = ${Q_j}^{N_3}\ mod\ p$, where N_3 is randomly chosen by S_m. Then following scenario can be happened.

Step 1: User U_k will accept the forge message which is send by S_m.

Step 2: At that time if another valid verification message $\{ID_k,\ X_1,\ X_2\}$ comes from S_j by computing X_1 = $y_k \oplus N_1 \oplus N_2$, X_2 = ${Q_j}^{N_2}\ mod\ p$ where N_2 is randomly chosen by S_j. The card reader will confuse which one will be accepted.

Step 3: If card reader choose forge message of S_m and send (ID_k, X_3) to S_j by computing X_3=${Q_j}^{N_1 \times N_3}\ mod\ p$. S_j will reject the valid user U_k because $X_3 = {Q_j}^{N_1 \times N_3}\ mod\ p$ and $X_3^{'} = {Q_j}^{N_1 \times N_2}\ mod\ p$ will not same.

3.4 Password Guessing Attack

After extracting the secret values (x_i, y_i, z_i) from a legal user's smart card by some means, the attacker can easily find out PW_i using following steps:

Step 1: An attacker computes $h(PW_i^*)$ and checks the correctness whether $h(PW_i^*) = z_i \oplus y_i$.

Step 2: An attacker repeats the above process until the correct password is obtained. After some guessing, an attacker can find out the correct password.

3.5 Smart Card Stolen Attack

Suppose a user U_i either lost or stolen by an attacker of his/her smart card. After getting the smart card, the attacker can extract the secret information $\{x_i, y_i, z_i, p, g, h(.)\}$ from the user's smart card. We also assume that attacker store the $i-th$ login message of the user U_i. As a result attacker can store the value $\{ID_i,\ SID_j,\ d_i,\ e_i,\ N_1\}$. By using these secret information and stored parameter, attacker can create valid login message, described in user impersonation and server masquerading attack procedure in section 3.

3.6 Password Change Attack

After obtaining secret values (x_i, y_i, z_i) from the legal user's smart card, an attacker can easily find out PW_i by performing the password guessing technique as describe in attack subsection 3.4. After getting correct PW_i, the attacker can change the valid user's password from the password change phase. So card reader will always reject the valid user in login phase.

4. PROPOSED SCHEME

In this section, we will propose the improvement of Taygi et al.'s [1] remote user authentication scheme in multi-server environment using smart card to overcome their weaknesses. It is assumed that Registration Center (RC) is a trusted authority.

We now describe the registration procedure of proposed scheme is as follows:

4.1 Registration Phase

This phase is divided into two sub-phases: Server Registration phase and User Registration phase.

4.1.1 Server Registration Phase

In this phase, S_j selects SID_j and submits it to RC over a secure channel. After receiving the registration request from S_j, RC generate a random nonce w_j for each server ($j = 1\ to\ m$) and computes the server secret parameter SS_j = $h(SID_j,\ h(x \parallel d)) \oplus h(x \parallel w_j)$ and sends $\{SS_j,\ h(d),\ h(x \parallel w_j)\}$ to S_j over a secure channel.

4.1.2 User Registration Phase

U_i selects ID_i, PW_i and computes $B_i = g^{PW_i \times k}$ mod p, IDU_i = $h(ID_i \parallel B_i)$ and submits $\{IDU_i, B_i, ID_i\}$ to RC over a secure channel. After receiving registration request from U_i, RC chooses a random nonce n_i for each user ($i = 1\ to\ n$) and computes Y_i = $h(h(IDU_i) \parallel h(d))$, Z_i = $h(h(IDU_i) \parallel h(B_i))$, K_i = $h(B_i) \oplus h(x \parallel d) \oplus h(x \parallel n_i)$, D_i = $Z_i \oplus h(x \parallel n_i)$, G_i = $h(IDU_i \parallel Z_i \parallel h(x \parallel n_i) \parallel h(x \parallel d))$, X = $g^{h(d)}\ mod\ p$ and issues a smart card for U_i after storing $\{K_i,\ Y_i,\ G_i,\ D_i,\ X,\ p,\ g,\ h(\cdot)\}$ into memory of user's smart card. After getting smart card, user U_i store k into memory of smart card.

4.2 Login Phase

U_i inserts the smart card into the card reader and submit ID_i and PW_i. The card reader computes B_i^* = $g^{PW_i \times k}\ mod\ p$, IDU_i^* = $h(ID_i \parallel B_i^*)$, Z_i^* = $h(h(IDU_i^*) \parallel h(B_i^*))$, R_1 = $Z_i^* \oplus D_i$, R_2 = $K_i \oplus h(B_i^*) \oplus R_1$, G_i^* = $h(IDU_i^* \parallel Z_i^* \parallel R_1 \parallel R_2)$ and checks whether computed G_i^* equals stored G_i. If true, card reader generates a random nonce N_1 and computes C_1 = ${B_i^*}^{N_1}\ mod\ p$, C_2 = ${X^{PW_i \times k \times N_1}}\ mod\ p$, Q_i = $h(SID_j,\ R_2) \oplus h(C_2)$, C_3 = $h(IDU_i^*) \oplus h(C_1 \parallel C_2) \oplus Q_i$, L_i = $h(h(IDU_i^*) \parallel Y_i)$, E_i = $(h(IDU_i^*) + Y_i \times h(L_i \parallel Q_i \parallel C_2 \parallel h(IDU_i^*)))\ mod\ (p-1)$ and sends the login request message $\{ID_i,\ SID_j,\ Y_i,\ C_1,\ E_i,\ C_3\}$ to S_j.

In the following we prove the **correctness of G_i^* = G_i :**

G_i^* = $h(IDU_i^* \parallel Z_i^* \parallel R_1 \parallel R_2)$

R_1 = $Z_i^* \oplus D_i$ = $Z_i^* \oplus Z_i \oplus h(x \parallel n_i)$ = $h(x \parallel n_i)$, since Z_i^* = Z_i

R_2 = $K_i \oplus h(B_i^*) \oplus R_1$

R_2 = $h(B_i) \oplus h(x \parallel d) \oplus h(x \parallel n_i) \oplus h(B_i^*) \oplus h(x \parallel n_i)$

R_2 = $h(x \parallel d)$, Since $B_i = B_i^*$

G_i^* = $h(IDU_i^* \parallel Z_i^* \parallel h(x \parallel n_i) \parallel h(x \parallel d))$ = G_i, since IDU_i^* = IDU_i.

4.3 Authentication Phase

After receiving the login request message $\{ID_i, SID_j, Y_i, C_1, E_i, C_3\}$, S_j first check the format of ID_i. If it is valid then computes C_2^* = ${C_1}^{h(d)}\ mod\ p$, Q_i^* = $SS_j \oplus h(x \parallel w_j) \oplus h(C_2^*)$, R_3 = $h(C_1 \parallel C_2^*) \oplus Q_i^* \oplus C_3$, Y_i^* = $h(R_3 \parallel h(d))$ and checks whether computed Y_i^* equals Y_i. If equals then computes L_i^* = $h(R_3 \parallel Y_i^*)$, T_2 = $h(L_i^* \parallel Q_i^* \parallel C_2^* \parallel R_3)$, E_i^* = $R_3 + Y_i^* \times T_2\ mod\ (p-1)$ and checks whether E_i^* equals E_i. If equals then proceed to mutual authentication Phase else reject user U_i. Further S_j generates a random nonce N_2 and computes A_1 = $N_2 \oplus Q_i^* \oplus R_3$, $SKey_{ij}$ = $h(Q_i^* \parallel N_2 \parallel L_i^*)$, A_2 = $h(N_2 \parallel SKey_{ij})$. Then sends $\{A_1, A_2\}$ to user U_i. After receiving it, U_i

computes N_2^* = $A_1 \oplus Q_i \oplus h(IDU_i^*)$, $SKey_{ij}^*$ = $h(Q_i \parallel N_2 \parallel L_i)$, A_2^* = $h(N_2^* \parallel SKey_{ij}^*)$ and checks A_2^* equals A_2. If equals then mutual authentication is achieved and both the parties agree with a common shared session key $SKey_{ij}$ for securing future data communications.

In the following we prove the correctness of Y_i^* = Y_i, E_i^* = E_i and A_2^* = A_2.

Correctness of Y_i^* = Y_i:

Y_i^* = $h(R_3 \parallel h(d))$

R_3 = $h(C_1 \parallel C_2^*) \oplus Q_i^* \oplus C_3$

R_3 = $h(C_1 \parallel C_2^*) \oplus Q_i^* \oplus h(IDU_i^*) \oplus h(C_1 \parallel C_2) \oplus Q_i$

R_3 = $h(IDU_i^*)$, since C_2^* = C_2 and Q_i^* = Q_i

Y_i^* = $h(IDU_i^* \parallel h(d))$ = Y_i, Since $IDU_i^* = IDU_i$

Correctness of E_i^* = E_i:

$E_i^* = R_3 + Y_i^* \times T_2 \; mod \; (p-1)$

$E_i^* = h(IDU_i^*) + Y_i \times h(L_i^* \parallel Q_i^* \parallel C_2^* \parallel h(IDU_i))$

E_i^* = E_i, Since L_i^* = L_i, Q_i^* = Q_i and C_2^*= C_2.

Correctness of A_2^* = A_2:

A_2^* = $h(N_2^* \parallel SKey_{ij}^*)$

N_2^* = $A_1 \oplus Q_i \oplus h(IDU_i^*)$

N_2^* = $N_2 \oplus Q_i^* \oplus R_3 \oplus Q_i \oplus h(IDU_i^*)$

N_2^* = N_2, since Q_i^* = Q_i and R_3 = $h(IDU_i^*)$

$SKey_{ij}^*$ = $SKey_{ij}$, since N_2^* = N_2

A_2^* = $h(N_2 \parallel SKey_{ij})$

A_2^* = A_2, since $SKey_{ij}^*$ = $SKey_{ij}$.

4.4 Password Change Phase

This phase is invoked when U_i wants to change the password. U_i inserts the smart card into the card reader and submits ID_i and PW_i. Then card reader computes B_i^* = $g^{PW_i \times k}$ mod p, IDU_i^* = $h(ID_i \parallel B_i)$, Z_i^* = $h(h(IDU_i^*) \parallel h(B_i^*))$, R_4 = $Z_i^* \oplus D_i$, R_5 = $K_i \oplus h(B_i^*) \oplus R_4$, G_i^* = $h(IDU_i^* \parallel Z_i^* \parallel R_4 \parallel R_5)$ and checks whether G_i^* equals stored G_i. If equals, U_i enters a new password PW_i^{new}. Then card reader computes B_i^{new} = $g^{PW_i^{new} \times k}$ mod p, Z_i^{new} = $h(h(IDU_i^*) \parallel h(B_i^{new}))$, K_i^{new} = $h(B_i^{new}) \oplus R_4 \oplus R_5$, D_i^{new} = $Z_i^{new} \oplus R_4$, G_i^{new} = $h(IDU_i^* \parallel Z_i^{new} \parallel R_4 \parallel R_5)$ and stores K_i^{new}, D_i^{new} and G_i^{new} instead of K_i, D_i and G_i respectively in the memory of smart card. Thus U_i can change the password without taking any assistance from S_j.

5. CRYPTANALYSIS OF PROPOSED SCHEME

This section describes cryptanalysis of proposed scheme. In this paper, to analyze the security analysis of proposed scheme, it can be assumed that an attacker could obtain the secret values stored in the smart card by monitoring the power consumption [16][17] and intercept messages communicating between the user and the server. Also it can be assumed that an attacker may possess the capabilities to thwart the security scheme.

5.1 User Impersonation Attack

To impersonate as a legitimate user, an attacker attempts to make a forged login request message which can be authenticated to a server. However, the attacker can not impersonate as the legitimate user by forging the login request message even if the attacker can extract the secret values $\{K_i, \; Y_i, \; G_i, \; D_i, \; X, \; k\}$ stored in the user's smart card, because the attacker can not compute the valid login request message $\{E_i, \; C_1, \; C_3\}$ without knowing the secret password PW_i of valid user U_i. In proposed scheme if the attacker wants to get the secret parameter PW_i, he/she must have to solve the inversion of cryptographic hash function which is computationally hard. So proposed scheme is secured against user impersonation attack.

5.2 Server Masquerading Attack

To masquerade as a legitimate server, an attacker attempts to make the forged reply message which can be masqueraded to the user when receiving the users login request message. However, the attacker can not masquerade as the server by forging the reply message, because it is hard to compute $\{A_1, A_2\}$ by an attacker without knowing the secret value $h(d)$ and $h(x \parallel w_j)$ kept by the server. Hence the attacker can not masquerade as the legitimate server to the user by launching the server masquerading attack.

5.3 Collaboration Attack between a Valid User and a Server

In proposed scheme to perform collaboration attack between a valid user and a server, another authenticated server S_m needs to compute the secret key of S_j. To compute the secret key of S_j, S_m needs to compute $h(x \parallel w_j)$ from the communicating message between the user U_i and the server S_j. But the communicating message between the user and server are independent of $h(x \parallel w_j)$. Hence it is hard to compute the secret key SS_j of the server S_j by another server S_m. So proposed scheme is secured against the collaboration attack between a valid user and a server.

5.4 Password Guessing Attack

After getting secret values $\{K_i, \; Y_i, \; G_i, \; D_i, \; X, \; k\}$ stored in the user U_i smart card, the attacker attempts to derive the user U_i's password PW_i using K_i = $h(B_i) \oplus h(x \parallel d) \oplus h(x \parallel n_i)$, G_i and D_i in the user registration phase. To guess the user's password, attacker needs to compute B_i from K_i, G_i and D_i but it is not possible. So proposed scheme is secured against password guessing attack.

5.5 Smart Card Stolen Attack

It can be assumed that the user U_i has either lost his/her smart card or stolen by an attacker. After getting the smart card, an attacker can extract the secret information $\{K_i, \; Y_i, \; G_i, \; D_i, \; X, \; k\}$ from the smart card. Also can be assumed that attacker stores the $i - th$ login message $\{SID_j, \; Y_i, \; C_1, \; E_i, \; C_3\}$ of the user U_i. After getting all these parameter that is login message and smart card parameter, it is hard to derive PW_i, server secret information $h(d)$ and $h(x \parallel w_j)$ by the attacker. So attacker can not create the valid login message. As a result proposed scheme is secured against smart card stolen attack.

5.6 Insider Attack

Generally, many user uses same password for their convenience of remembering and easy of use whenever required. However if the system manager or privileged insider of the server knows user's password, he/she may try to access user's U_i other accounts in other server. But in proposed scheme, the system manager or privileged insider of the server can not derive user's password PW_i because user have submitted $h(B_i)$ = $g^{PW_i \times k}$ mod p instead of PW_i. To get correct PW_i, system manager must have to solve Diffie-Hellman problem [18] which is computationally hard. So proposed scheme resists insider attack.

5.7 Password Change Attack

As described in attack 5.4 and attack 5.5 in section 5, proposed scheme can withstand the Password guessing attack and smart card stolen attack. To perform password change attack, an attacker have to provide correct ID_i and PW_i to the card reader. But in proposed scheme, there is no way to get correct PW_i to the attacker. So proposed scheme is secured against password change attack.

6. PERFORMANCE ANALYSIS OF PROPOSED SCHEME

In this section we evaluated the performance of proposed scheme with Taygi et al.'s [1] scheme. We have compare login and authentication phases of proposed scheme with Taygi et al.'s [1] scheme because these phases are used frequently. Table 1 shows the computation over head and Table 2 shows the communication cost and storage cost of proposed scheme and Taygi et al.'s [1] scheme. In Table 1, T_h is the time required for hashing operation and T_e is the time required for exponentiation operation and T_m is the time required for multiplication operation. Though, proposed scheme resists different possible attacks of Taygi et al.'s [1] scheme, in spite of proposed scheme provides better computation cost than the related scheme because of exponentiation and multiplication operation. In proposed scheme, only four (4) exponentiation and five (5) multiplication are used whereas Taygi et al.'s [1] scheme requires sixteen (16) exponentiation and eight (8) multiplication operations.

It can be reasonably assumed that the length of ID_i, PW_i and SID_j is 64 bits each and $h(\cdot)$, random nonce returns 128 bits and length of g, p are 1024 bits each. The communication cost (capacity of transmitting message) of proposed scheme and Taygi et al.'s [1] scheme is 2688 $bits$ = $(64+64+128+1024+1024+128+128+128)$ and 4608 $bits$ = $(64+64+1024+1024+128+64+128+1024+64+1024)$ respectively. Also the storage cost (stored into the memory of smart card) takes almost same bits of proposed scheme and their scheme that is 3840 $bits$, 3456 $bits$ respectively. Table 3 shows that their scheme is insecure against different possible attacks. Further proposed scheme provides strong authentication against different attacks described in section 5. After resisting all possible attacks of related scheme, proposed scheme provides low computational and low communication cost than their scheme. Hence proposed scheme is more efficient and secure than Taygi et al.'s [1] scheme.

Table 1. comparison of computational cost of proposed scheme with related scheme

Scheme ⇒ Phase ⇓	Taygi et al.'s [1]	Our
Login Phase	$3T_h + 5T_e + 3T_m$	$10T_h + 3T_e + 4T_m$
Authentication Phase	$4T_h + 11T_e + 5T_m$	$9T_h + 1T_e + 1T_m$

Table 2. comparison of communication and storage cost of proposed scheme with related scheme

Scheme ⇒ Item ⇓	Taygi et al.'s [1]	Our
Communication Cost	4608 bits	2688 bits
Storage Cost	3456 bits	3840 bits

ADVANTAGES :

Table 3. attack comparison of proposed scheme with related scheme

Scheme ⇒ Attack ⇓	Taygi et al.'s [1]	Our
User Impersonation Attack	Yes	No
Server Masquerading Attack	Yes	No
Collaboration Attack between a Valid User and a Server	Yes	No
Password Guessing Attack	Yes	No
Smart Card Stolen Attack	Yes	No
Password Change Attack	Yes	No
Mutual Authentication	No	Yes

(1) **Mutual Authentication**

As described the attack procedure in section 5, proposed scheme can withstand the user impersonation attack and the server masquerading attack, consequently the proposed scheme provides mutual authentication between the user and the remote server. Namely, even if the attacker can extract the secret values $\{K_i,\ Y_i,\ G_i,\ D_i,\ X,\ k\}$ stored in the users smart card, the user can be authenticated to the server and vice versa. Because the attacker can not make the login request message $\{SID_j,\ Y_i,\ C_1,\ E_i,\ C_3\}$ and the reply message $\{A_1, A_2\}$ without knowing the secret value PW_i of user U_i and the secret values SS_i, $h(d)$ of the server S_j respectively.

(2) **Single Registration**

In proposed scheme, if a user register to registration server once then the user can access all the registered servers.

(3) **No Verification Table**

None of the registered servers need to maintain a verification table.

(4) **Securely Password Change Without Taking Help From Server and RC**

A valid user can change the password freely and securely without taking any help from the servers or registration center. As the card reader first verifies the old password in the password change phase, so unauthorized users cannot change the authorized users password even if they get the valid user's smart card.

(5) **Early Wrong Password Detection**

If the user U_i inputs a wrong password by mistake in password change phase, it will be quickly detected by the card reader itself since card reader computes B_i^* = $g^{PW_i \times k}\ mod\ p$, IDU_i^* = $h(ID_i \parallel B_i^*)$, Z_i^* = $h(h(IDU_i^*) \parallel h(B_i^*))$, R_1 = $Z_i^* \oplus D_i$, R_2 = $K_i \oplus h(B_i^*) \oplus R_1$, G_i^* = $h(IDU_i^* \parallel Z_i^* \parallel R_1 \parallel R_2)$ and checks whether G_i^* equals G_i. Hence proposed scheme provides early wrong password detection.

(6) **Server's Unique Secret Parameter**

In proposed scheme, each server has unique secret parameter SS_j = $h(SID_j,\ h(x \parallel d)) \oplus h(x \parallel w_j)$ used to authenticate the user. Hence there is no need to store the secret parameter of all the servers in the memory of smart card.

(7) **Mutual Authentication and Session Key Agreement Without Help of RC**

Any valid users and valid servers can authenticate each other and then agree on a session key without any support from

the registration center. The generated session key $SKey_{ij}$ = $h(Q_i^* \parallel N_2 \parallel L_i^*)$ will be different for each login session.

(8) **Solves Time Synchronization Problem**

The proposed scheme uses randomly generated nonce N_1 and N_2 instead of time stamps to avoid time synchronization problem.

7. CONCLUSION

This paper shows that Taygi et al.'s [1] scheme suffers from different attacks. To over come these weaknesses a proposed scheme is given in this paper. Further, the proposed scheme is more efficient in terms of computational and communication cost than that of Taygi et al.'s [1] scheme. In addition, proposed scheme provides mutual authentication between user and authentication server and also user can change his/her password freely without help of registration server.

It is shown that the proposed scheme provides strong security protocol based on the user's password. For the better security, biometric feature as well as password can be incorporate in the protocol.

8. REFERENCES

[1] J.K. Tyagi, A.K. Srivastava and P.S. Patwal, "Remote user authentication scheme in multi-server environment using smart card", International Journal of Computer Applications (0975 8887), vol. 57, no. 12, pp. 1-5, November 2012.

[2] Li, L., Lin, I., Hwang, M. S., "A remote password authentication scheme for multi-server architecture using neural networks", IEEE Transaction on Neural Networks, vol. 12, pp. 1498-1504, 2001.

[3] Lin, I. C., Hwang, M. S., Li, L. H., "A new remote user authentication scheme for multi-server architecture", Future Generation Computer Systems, vol. 19, pp. 13-22, 2003.

[4] Elgamal, T., "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, vol. 31(4), pp. 469-472, 1985.

[5] Cao, X., Zhong, S., "Breaking a remote user authentication scheme for multi-server architecture", IEEE Communications Letters, vol. 10, pp. 580-581, 2006.

[6] Juang, W. S., "Efficient password authenticated key agreement using smart cards", Computers and Security vol. 23, pp. 167-173, 2004.

[7] Ku, W. C., Chuang, H. M., Chiang, M. H. and Chang, K. T., "Weaknesses of a multi-server password authenticated key agreement scheme", In Proceedings of the 2005 National computer Symposium, vol. 138, pp. 1-5, 2005.

[8] Chang, C. C., Lee, J. S., "An efficient and secure multi-server password authentication scheme using smart cards", In Proceedings of the International Conference on Cyberworlds, pp. 417-422, 2004.

[9] Liao, Y. P., Wang, S. S., "A secure dynamic ID based remote user authentication scheme for multi-server environment", Computer Standards and Interfaces, vol. 31, pp. 24-29, 2009.

[10] Chen, T. Y., Hwang, M. S., Lee, C. C. and Jan, J. K., "Cryptanalysis of a secure dynamic ID based remote user authentication scheme for multi-server environment", In Proceedings of the 4th International Conference on Innovative Computing, Information and Control, pp. 725-728, 2009.

[11] Hsiang, C., Shih, W. K., "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment", Computer Standards and Interfaces, vol. 31, pp. 1118-1123, 2009.

[12] Sood, S. K., Sarje, A. K., Singh, K., "A secure dynamic identity based authentication protocol for multi-server architecture", Journal of Network and Computer Applications, vol. 34, pp. 609-618, 2011.

[13] Tsai, J. L., "Efficient multi-server authentication scheme based on one-way hash function without verification table", Computers and Security, vol. 27, pp. 115-121, 2008.

[14] Zhu, H, Liu, T. and Liu, J., "Robust and simple multi-server authentication protocol without verification table", In Proceedings of the 9th International Conference on Hybrid Intelligent Systems, pp. 13-22, 2009.

[15] Denning, Dorothy E, Sacco, Giovanni Maria, "Timestamps in key distributed protocols", Communication of the ACM, vol. 24, pp. 533-535, 1981.

[16] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis", Proceedings of Advances in Cryptology, pp. 388-397, 1999.

[17] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks, IEEE Transactions on Computers, vol. 51, no. 5, pp. 541-552, 2002.

[18] Whitfield Diffie and Martin E. Hellman, "New Directions in Cryptography". IEEE Transactions on Information Theory, vol. IT-22 (6), pp. 644-654, November 1976.