Cryptanalysis and Efficient Dynamic ID Based Remote User Authentication Scheme in Multi-server Environment Using Smart Card

Ruhul Amin
Computer Science and Technology, Jakir Hossain Institute of Polytechnic
Aurangabad, Murshidabad, West Bengal, India
(Email: amin_ruhul@live.com)

(Received Sept. 17, 2013; revised and accepted July 5, 2014 & Nov. 6, 2014)

Abstract

Resembling the single server environment, if the multi-server environment using smart card provides the users to access the different servers after registering once with the registration center and uses the same password and identity for all the service provider's servers, then security would be the matter of great concern. So, remote user authentication scheme becomes necessary to provide a better security. In this regard, many dynamic ID-based remote user authentication schemes in multi-server environment using smart card have been proposed in the literature. In 2012, Sood proposed Dynamic Identity Based Authentication Protocol for Two-Server Architecture and claimed that his scheme is more efficient in terms of security. But it is pointed out that Sood's scheme is insecure against off-line identity guessing attack, off-line password guessing attack, privileged insider attack, user impersonation attack, session key recovery attack and many logged in users' attack. In 2012, Li et al.'s proposed a scheme for providing better performance than Sood's scheme. But unfortunately Li et al.'s scheme also is insecure against off-line identity guessing attack, off-line password guessing attack, user impersonation attack and many logged in users' attack. To overcome the above mention attacks for both the schemes and related attacks on remote user authentication like (identity and password guessing attack, user impersonation attack, server masquerading attack, insider attack, session key discloser attack, smart card stolen attack, replay attack, many logged in users' attack and stolen verifier attack etc.), we proposed an efficient dynamic ID-Based remote user authentication scheme in multi-server environment using smart card. After performance analysis, the proposed scheme has lower computation complexity, better communication cost and higher security that makes the authentication system more secure and efficient than both Sood's and Li et al.'s schemes published earlier.

Keywords: Authentication, dynamic ID, multi-server

1 Introduction

It is terribly inefficient and difficult for the users to remember different identities and passwords for accessing various remote servers repetitively, when users used many single-server environments. However, users can login the control server only once and then access numerous different remote service providing servers, if they use the multi-server authentication scheme [1, 4, 5, 7, 14]. In 2000, first Ford and Kaliski [6] proposed password based multi-server authentication protocol that splits password among different servers but the protocol has high computation due to use of public keys by the servers. Then in 2001, Jablon [10] improved Ford and Kaliski's protocol, which do not use public keys. In 2003, Lin et. al.'s [16] proposed a multi-server authentication protocol based on the ElGamal digital signature scheme. But the use of public keys makes this protocol computation intensive. In 2004, Juang [11] proposed a smart card based multi-server authentication protocol using asymmetric encryption algorithm without using any verification table. In the same year, Chang and Lee [3] proposed an improved scheme over Juang [11] scheme. In 2007, Hu et al. [9] proposed an efficient password authentication key agreement protocol for multi-server architecture in which user can access multiple servers using smart card and one weak password and also provides mutual authentication and secret session key for secure communication. In 2008, Tsai [21] proposed a multi-server authentication protocol using smart cards based on the nonce and one-way hash function that does not require storing any verification table on the server and the registration center. This protocol does not use any symmetric key or asymmetric key algorithm for implementation. In 2009, Liao and Wang [15] proposed a dynamic identity based remote user authentication protocol using smart cards to achieve users' anonymity. This protocol uses only cryptographic one-way hash function for the implementation. In the same year, Hsiang and Shih [8] found that Liao and Wangs protocol is susceptible to insider attack, masquerade attack, server spoofing attack, registration center spoofing attack and does not provide mutual authentication as well. To overcome these drawbacks, they proposed an improved scheme over Liao and Wang's [15] scheme. Then in 2010, Sood et al. [20] showed that Hsiang and Shih's [8] scheme is insecure against replay attack, impersonation attack and stolen smart card attack.

The remainder of this paper is organized as follows: Section 2. briefly reviews the Sood's [19] scheme. Section 3. shows cryptanalysis of Sood's [19] scheme. Section 4. briefly reviews the Li et el.'s scheme. Section 5. describes cryptanalysis of Li et al.'s [13] scheme. Section 6. describes the proposed scheme. Section 7. shows the cryptanalysis of the proposed scheme. Section 8. compares the performance analysis with related schemes published earlier. We conclude the paper in Section 9. Finally, references are given in Section 10.

1.1 Contribution

In this paper, we have briefly reviewed Sood's and Li et al.'s authentication protocol for multi-server environment. Then, we demonstrated that both schemes suffers from several attacks described in Section 3 and Section 5 respectively. Afterward, we proposed a remote user authentication protocol for multi-server environment. After cryptanalysis of the proposed protocol, it can be claimed that the proposed protocol has no security weaknesses and takes minimum computational and communication cost than related scheme.

1.2 Preliminaries

In this section, a briefly review the basic concepts of cryptographic one-way hash function and a related mathematical problem are introduced.

Cryptographic One-way Hash Function: A cryptographic hash function maps a string of arbitrary length to a string of fixed length called the hashed value. It can be symbolized as: $h : X \rightarrow Y$, where $X = \{0,1\}^*$, and $Y = \{0,1\}^n$. X is binary string of arbitrary length and Y is a binary string of fixed length n. It is used in many cryptographic applications such as digital signature, random sequence generators in key agreement, authentication protocols and so on. Cryptographic one-way hash function satisfies the following properties:

1) *Preimage Resistant:* It is hard to find m from given y, where $h(m) = y$.
2) *Second-Preimage Resistant:* It is hard to find input $m' \in X$ such that $h(m) = h(m')$ for given input $m \in X$ and $m' \neq m$.
3) *Collision Resistant:* It is hard to find a pair $(m, m') \in X \times X$ such that $h(m) = h(m')$, where $m \neq m'$.
4) *Mixing-Transformation:* On any input $m \in X$, the hashed value $y = h(m)$ is computationally indistinguishable from a uniform binary string in the interval $\{0, 2^n\}$, where n is the output length of hash $h(\cdot)$.

Factorization Problem [18]: It is computationally infeasible to find two large primes p and q each of length at least 1024-bits from given n $(= p \times q)$.

2 Brief Review of Sood's Scheme

This section presents a brief description of Sood's [19] dynamic ID-based remote user authentication scheme in multi-server environment using smart card. The notations used throughout this paper are summarized in Table 1.

Table 1: Notation used

CS	$\longrightarrow$	Control Server
S_k	$\longrightarrow$	$k-th$ Service Provider Server
U_i	$\longrightarrow$	$i-th$ user
ID_i	$\longrightarrow$	Identity of U_i
PW_i	$\longrightarrow$	Password chosen by U_i
x	$\longrightarrow$	Secret key of Control Server CS
$H(\cdot)$	$\longrightarrow$	Cryptographic one-way hash function
SK	$\longrightarrow$	Shared secret session key
$\oplus$	$\longrightarrow$	Bitwise xor operation
$\parallel$	$\longrightarrow$	Concatenate operation
$(\cdot)$	$\longrightarrow$	Multiplication operation

Sood's [19] scheme consists of the following phases: Registration Phase, Login Phase, Authentication and Session Key Agreement Phase and Password Change Phase.

2.1 Registration Phase

In this phase, user U_i submits identity ID_i and password PW_i to the Control server over secure channel for registration. After receiving ID_i and PW_i, Control server computes $Z_i = H(ID_i \parallel PW_i) \oplus H^2(x)$, $V_i = y_i \oplus ID_i \oplus H(x)$, $B_i = H(ID_i, PW_i) \oplus PW_i \oplus y_i$ and $C_i = H(y_i) \oplus ID_i \oplus x$, where x is the secret key of the control server and y_i is the random number chosen by the CS such that $y_i \oplus x$ will be unique for each user. Then, Control server CS stores $y_i \oplus x$ in its database corresponding to C_i and issues a smart card for the user U_i by storing the security parameter $Z_i, V_i, B_i, H(\cdot)$ into the memory of smart card.

All service provider servers have to register themselves with the control server CS and CS agrees on unique secret key SK_k with each service provider S_k. Then, S_k remembers secret key SK_k and CS store SK_k by computing $SK_k \oplus H(x \parallel SID_k)$ corresponding service provider identity SID_k. The CS sends ID_i and $H(y_i)$ corresponding to newly registered user U_i to all service provider. Then, all the service provider store ID_i and $H(y_i)$ in the database for further use.

2.2 Login Phase

In the login phase, user U_i insert his/her smart card into cardreader and submits ID_i^* and password PW_i^* and choose the identity of service provider server SID_k. Then, smart card computes $y_i = B_i \oplus H(ID_i^* \parallel PW_i^*) \oplus PW_i^*$, $H(x) = V_i \oplus y_i \oplus ID_i^*$, $Z_i^* = H(ID_i^* \parallel PW_i^*) \oplus H^2(x)$ and verifies whether computed Z_i^* is equals with stored Z_i. If the verification holds, smart card generates random nonce N_1 and computes $CID_i = V_i \oplus y_i \oplus H(y_i) \oplus N_1$, $M_i = H^2(x) \oplus N_1$ and $E_i = H(y_i \parallel H(x) \parallel N_1 \parallel ID_i \parallel SID_k)$. Then, smart card sends login message $\{SID_k, CID_i, M_i, E_i\}$ to the service provider S_k through public channel.

2.3 Authentication and Session Key Agreement Phase

After receiving login request message $\{SID_k$, CID_i, M_i, $E_i\}$, server S_k generates random nonce N_2 and computes $G_i = SK_k \cdot N_2$. Then, service provider server S_k sends login request message $\{SID_k, CID_i, M_i, E_i, G_i\}$ to the control server. After receiving it, control server computes $N_1 = M_i \oplus H^2(x)$, $N_2 = G_i \oplus SK_k$ and $C_i^* = CID_i \oplus N_1 \oplus H(x) \oplus x$. Then, CS checks the condition whether computed C_i^* is identical with the stored C_i in its database or not. If the condition does not hold, control server rejects the login request otherwise extract y_i from $y_i \oplus x$ stored in the database. Then, control server further computes $ID_i = C_i \oplus H(y_i) \oplus x$, $E_i^* = H(y_i \parallel H(x) \parallel N_1 \parallel ID_i \parallel SID_k)$ and compares E_i^* with the received E_i to verify the legitimacy of the user U_i and service provider S_k. If the condition holds, control server extracts SK_k from $SK_k \oplus H(x \parallel SID_k)$ stored in the database. Then, control server generates random nonce N_3 and computes $A_i = N_1 \oplus N_3 \oplus H(SK_k)$, $D_i = ID_i \oplus H(N_1 \oplus N_2 \oplus N_3)$, $F_i = H[H(N_1 \oplus N_2 \oplus N_3) \parallel ID_i \parallel H(y_i)]$, $T_i = N_2 \oplus N_3 \oplus H(y_i \parallel ID_i \parallel H(x) \parallel N_1)$ and sends message $\{A_i, D_i, F_i, T_i\}$ to the service provider server S_k.

Service provider server S_k then computes $N_1 \oplus N_3 = A_i \oplus H(SK_k)$, $ID_i = D_i \oplus H(N_1 \oplus N_2 \oplus N_3)$ and extracts $H(y_i)$ corresponding ID_i from its database. Afterward, server S_k computes $F_i^* = H[H(N_1 \oplus N_2 \oplus N_3) \parallel ID_i \parallel H(y_i)]$ and compares F_i^* with F_i to verify the legitimacy of control server. If the above condition holds, server S_k sends F_i and T_i to smart card of user U_i.

After receiving F_i and T_i, smart card computes $N_2 \oplus N_3 = T_i \oplus H(y_i \parallel ID_i \parallel H(x) \parallel N_1)$ and $F_i^* = H[H(N_1 \oplus N_2 \oplus N_3) \parallel ID_i \parallel H(y_i)]$. Then, compares computed F_i^* with received F_i to verify the legitimacy of control server CS and service provider server S_k. If the above condition holds, then login request is accepted, otherwise rejects the session. Finally, user U_i, control server CS and service provider server S_k agree on the common secret session key as $SK = H(ID_i \parallel (N_1 \oplus N_2 \oplus N_3) \parallel H(y_i))$.

2.4 Password Change Phase

This phase is invokes 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 $y_i = B_i \oplus H(ID_i^* \parallel PW_i^*) \oplus PW_i^*$, $H(x) = V_i \oplus y_i \oplus ID_i^*$, $Z_i^* = H(ID_i^* \parallel PW_i^*) \oplus H^2(x)$ and compares the computed value of Z_i^* with stored value of Z_i. If identifies, U_i enters a new password PW_i^{new}. Then, card reader computes $Z_i^{new} = Z_i \oplus H(ID_i \parallel PW_i) \oplus H(ID_i \parallel PW_i^{new})$ and $B_i^{new} = B_i \oplus H(ID_i \parallel PW_i) \oplus PW_i \oplus H(ID_i \parallel PW_i^{new}) \oplus PW_i^{new}$. Then, stores Z_i^{new} and B_i^{new} instead of Z_i and B_i into memory of smart card.

3 Security Analysis of Sood's Scheme

In this section, the cryptanalysis of Sood's [19] scheme is presented. To analyze the security weaknesses of Sood's scheme, our assumptions are given as follows.

Assumption 1. *It can be assumed that an attacker could obtain the secret values stored in the smart card by monitoring the power consumption [12, 17] and an attacker can intercept all communicating messages between the users, the service provider servers S_k and control server CS.*

Assumption 2. *Due to the low entropy of ID_i and PW_i selected by U_i, we assume an adversary is able to off-line guess U_i's identity ID_i and password PW_i individually. However, he/she cannot off-line guess ID_i and PW_i simultaneously in polynomial time as pointed out by Sood et al. [20].*

Assumption 3. *It can also be assumed that a valid user can provide secret information of the control server CS to an attacker or a valid user can acts as an attacker after deriving secret information of the control server.*

Under these assumptions, it can be explained various attacks on Sood's [19] scheme such as off-line identity guessing attack, off-line password guessing attack, privileged insider attack, user impersonation attack, session key recovery attack and many logged in users' attack.

3.1 Off-line Identity Guessing Attack

User's identity can be either name, phone number, birthday or some meaningful text which can be easily guessed because of the low entropy. To successfully launch off-line identity guessing attack, an attacker has to keep control server's secret information $H^2(x)$ and $H(x)$ which can easily obtain under Assumption 3. After that, off-line identity guessing attack can be launched successfully as follows.

Step 1. From login phase of the protocol, an attacker can derives $N_{1a} = M_i \oplus H^2(x)$;

Step 2. Attacker computes $T_a = V_i \oplus H(x) = y_i \oplus ID_i$. So $y_i = T_a \oplus ID_i$;

Step 3. Then, Attacker computes $Z_a = CID_i \oplus V_i \oplus N_{1a} = y_i \oplus H(y_i) = T_a \oplus ID_i \oplus H(T_a \oplus ID_i)$;

Step 4. Now, attacker guess user's identity ID_i^{guess} separately and verifies the correctness $Z_a = T_a \oplus ID_i^{guess} \oplus H(T_a \oplus ID_i^{guess})$;

Step 5. Continue the above step until correct identity is obtained. After some guessing attacker can easily find the correct user identity. Thus, an attacker can successfully launch off-line identity guessing attack.

3.2 Off-line Password Guessing Attack

After launching successfully off-line identity guessing attack, an attacker can easily guess user's password from the smart card parameters Z_i in following steps:

Step 1. Attacker chooses PW_i^{guess} for the user U_i to find the correct password PW_i.

Step 2. Attacker then verifies the correctness of $Z_i = H(ID_i \parallel PW_i^{guess}) \oplus H^2(x)$ where $H^2(x)$ is known parameter to the attacker.

Step 3. The above steps will continue until the correct password obtained. After some guessing the attacker can easily find out the correct password. Thus, Sood's scheme can not resists off-line password guessing attack.

3.3 Privileged Insider Attack

Generally, many users use the 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. In Sood's scheme, user U_i provides his/her password PW_i to the remote server. As a result, Sood's scheme is insecure against insider attack, because system manager or privileged insider of the server may try to access the user's other accounts in other server using password PW_i.

3.4 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. Under our assumption, Sood's scheme can not resist user impersonation attack as follows.

Step 1. Attacker can compute $y_i = V_i \oplus ID_i \oplus H(x)$. Then, attacker can easily compute $H(y_i)$.

Step 2. Now, Attacker generates a random number N_a and can easily compute login message $CID_i^* = V_i \oplus y_i \oplus N_a$, $M_i^* = H^2(x) \oplus N_a$ and $E_i^* = H(y_i \parallel H(x) \parallel N_a \parallel ID_i \parallel SID_k)$. Then attacker sends $\{CID_i^*, M_i^*, E_i^*\}$ to the service provider server S_k to proof himself as a valid user.

Step 3. It can be easily proved that the sending login message by an attacker is valid to the service provider server S_k. Then service provider server sends reply messages F_i and T_i to the attacker by computing $F_i = H[H(N_a \oplus N_2 \oplus N_3) \parallel ID_i \parallel H(y_i)]$ and $T_i = N_2 \oplus N_3 \oplus H(y_i \parallel ID_i \parallel H(x) \parallel N_a)$, where N_2 and N_3 is random number chosen by service provider server S_k and control server CS respectively.

Step 4. After receiving reply message from service provider server S_k, attacker computes $N_2 \oplus N_3 = T_i \oplus H(y_i \parallel ID_i \parallel H(x) \parallel N_a)$. Then, attacker and service provider server agree on the valid session key by computing $SK = H(ID_i \parallel (N_1 \oplus N_2 \oplus N_3) \parallel H(y_i))$ which is used for secure communication.

3.5 Many Logged In Users' Attack

Many logged in users' attack can be successfully launched after successful performance of off-line identity guessing attack and off-line password guessing attack as described in Section 3. After getting correct password of user U_i, an attacker can easily compute the value y_i which is different for all users. Then, attackers or non-registered user can successfully access the service of the service provider server S_k as follows.

Step 1. Attacker or non-registered user choose his/her desired password PW_i^a and computes $Z_i^a = H(ID_i \parallel PW_i^a) \oplus H^2(x)$, $V_i^a = y_i \oplus ID_i \oplus H(x)$ and $B_i^a = H(ID_i \parallel PW_i^a) \oplus P_i \oplus y_i$, where ID_i is the valid user's identity which remains unchanged.

Step 2. Then, attacker or non-registered users stores $\{Z_i^a, V_i^a, B_i^a, H(\cdot)\}$ into memory of new smart card and it can be used by many attackers or non-registered users as a valid user.

The above attack proves that Sood's scheme can not be used for practical implementation in terms of security because without stealing user's smart card, many non-registered users can act as a valid user.

3.6 Session Key Recovery Attack

In Sood's scheme, user's U_i, service provider server S_k and control server CS agree on the common session key SK which is based on the difficulty of cryptographic one-way hash function. The Common secret session key SK depends on the secret parameter ID_i, y_i and random nonce N_1, N_2, N_3. In the user impersonation attack in Section 3 shows that an attacker can easily obtain ID_i, y_i and random nonce N_1, N_2, N_3. So, after obtaining all these secret parameters attacker can compute secret session key for every transaction of user U_i. As a result, Sood's scheme is insecure against session key recovery attack.

4 Brief Review of Li et al.'s Scheme

This section presents brief description of Li et al.'s [13] dynamic ID based remote user authentication scheme in multi-server environment using smart card. Li et al.'s [13] scheme consists of following phases: Registration phase, Login phase, Authentication and Session Key Agreement phase.

4.1 Registration Phase

Whenever a new user wants to get services from the remote server, he/she must have to register with control server CS as follows:

User U_i chose his/her desired identity ID_i and password PW_i and generates a random nonce b. Then, U_i computes $PWB_i = H((ID_i \parallel PW_i) \oplus b)$ and sends ID_i, PWB_i to the control server CS through secure channel. After receiving registration messages from user U_i, CS first verifies U_i's personal information and credit and if it is valid then computes $TID_i = (T_i \parallel ID_i)$, $\sigma_i = H(TID_i \parallel x) \oplus H((ID_i \parallel PW_i) \oplus b)$, where T_i is the registration time of user U_i. Afterward, CS stores $\{\sigma_i, H(TID_i), T_i, H(\cdot)\}$ into memory of the smart card and issues it for the user U_i. After getting smart card, user U_i stores b into memory of the smart card and keeps it secret for personal use.

4.2 Login Phase

Whenever user U_i wants to get service from server S_k, then user U_i inserts his/her smart card into the card reader and submits ID_i and password PW_i and chooses the identity of service provider server SID_k. Then, smart card computes $TID_i^* = (T_i \parallel ID_i)$, $PWB_i = H((ID_i \parallel PW_i) \oplus b)$ and checks whether $TID_i^* = TID_i$ or not. If it does not hold, the smart card terminates this login, otherwise generates a random number N_1 and computes $\alpha_1 = \sigma_i \oplus PWB_i \oplus N_1$, $\alpha_2 = H((TID_i \parallel SID_k) \oplus N_1)$. Then, sends $\{TID_i,\ \alpha_1,\ \alpha_2\}$ to the service provider server S_k.

After receiving login messages $\{TID_i,\ \alpha_1,\ \alpha_2\}$ from user U_i, server S_k computes $\beta_1 = H(SID_k \parallel x) \oplus N_2$ and $\beta_2 = H((SID_k \parallel TID_i) \oplus N_2)$, where N_2 is the random number generated by service provider server S_k. Then, S_k sends $\{TID_i,\ \alpha_1,\ \alpha_2,\ SID_k,\ \beta_1,\ \beta_2\}$ to the control server CS through public channel.

4.3 Authentication and Session Key Agreement Phase

After receiving login request messages $\{TID_i,\ \alpha_1,\ \alpha_2,\ SID_k,\ \beta_1,\ \beta_2\}$ from server S_k, control server CS checks the validity of user's TID_i and server's SID_k. If both does not hold, rejects the connection, otherwise CS computes $N_1^* = \alpha_1 \oplus H(TID_i \parallel x)$ and verifies the freshness of N_1^*. If it does hold, CS further computes $\alpha_2^* = H((TID_i \parallel SID_k) \oplus N_1^*)$ and further compares with computed α_2^* equals with received α_2. if it is holds, then CS believes that user U_i is authentic, otherwise terminates the connection. Then, CS computes $N_2^* = \beta_1 \oplus H(SID_k \parallel x)$ and checks the freshness of N_2^*. If it does hold, CS further computes $\beta_2^* = H((SID_k \parallel TID_i) \oplus N_2^*)$ and further compares with computed β_2^* equals with received β_2. If it holds, then CS believes that service provider server S_k is authentic, otherwise terminates the connection. CS generates a random number N_3 and computes $\alpha^{'} = H(N_1^*) \oplus N_2^* \oplus N_3$, $\gamma_u = H(H(TID_i \parallel x) \oplus SK)$, $\beta^{'} = H(N_2^*) \oplus N_1^* \oplus N_3$ and $\gamma_s = H(H(SID_k \parallel x) \oplus SK)$, where SK is a common secret session key which is constructed by computing $SK = H(N_1^* \oplus N_2^* \oplus N_3)$. Finally, CS sends $\{\alpha^{'},\ \gamma_u,\ \beta^{'},\ \gamma_s\}$ to service provider server S_k.

After receiving the message from CS, server S_k computes $\beta^{''} = \beta^{'} \oplus H(N_2)$, $SK_s = H(\beta^{''} \oplus N_2)$, $\gamma_s^{'} = H(H(SID_k \parallel x) \oplus SK_s)$ and compares computed $\gamma_s^{'}$ with received γ_s. If it is invalid, server S_k terminates the connection, otherwise server S_k believes that control server CS is authentic and sends $\{\alpha^{'},\ \gamma_u\}$ to the smart card user U_i. It can be easily shown that $SK_s = SK$ common secret session key between user U_i, server S_k and CS.

After receiving the response message from server S_k, smart card computes $\alpha^{''} = \alpha^{'} \oplus H(N_1)$, $SK_u = H(\alpha^{''} \oplus N_1)$, $\gamma_u^{'} = H(H(TID_i \parallel x) \oplus SK_u)$ and compares computed $\gamma_u^{'}$ with received γ_u. If it is not valid, terminates the connection, otherwise user believes that server S_k and control server CS is authentic participants. Finally, three participants user, service provider server and control server agree with a common secret session key $SK = SK_s = SK_u$ which can be used in future for secure communication.

5 Cryptanalysis of Li et al.'s Scheme

In this section, the cryptanalysis of Li et al.'s [13] scheme is presented. To analyze the security weaknesses of Li et al.'s scheme, we assume Assumptions 1 and 2 which are described in Section 3 of this paper.

5.1 Off-line Identity Guessing Attack

During the registration phase, user U_i usually chooses an identity which is easily remembered for his/her convenience. These easy to remember identities are low entropy and thus attacker can easily guess user's identity. Generally user's identity is static and often confined to a predefined format, so it is more easily guessed by the attacker than the password. Li et al.'s scheme suffers from identity guessing attack as follows:

Step 1. An attacker extracts information $H(TID_i)$, T_i from the valid user's smart card by monitoring power consumption.

Step 2. Then, attacker chooses user's identity ID_i^a and verifies the correctness $H(TID_i) = H(T_i \parallel ID_i^a)$.

Step 3. Continue step 2 until correct identity is obtained. After some guessing, an attacker can find out the correct user's identity ID_i.

Thus, Li et al.'s scheme is insecure against off-line identity guessing attack.

5.2 Off-line Password Guessing Attack

In remote user authentication schemes, for the sake of user-friendliness, a user is often allow to select his/her desired password during the registration phase. Generally, the user chooses his/her password which is easy to remember. But these easy to remember passwords are of low entropy and an attacker can guess the user's password. After launching successfully off-line identity guessing attack, an attacker can easily guess user's valid password using stored smart card's parameters σ_i, b and service provider server's login message $\{TID_i,\ \alpha_1,\ \alpha_2,\ SID_k,\ \beta_1,\ \beta_2\}$ as follows:

Step 1. Attacker computes $N_1 = \alpha_1 \oplus \sigma_i \oplus PWB_i$. Now,

$$\begin{aligned}\alpha_2 &= H((TID_i \parallel SID_k) \oplus N_1) \\ &= H((TID_i \parallel SID_k) \oplus \alpha_1 \oplus \sigma_i \oplus PWB_i) \\ &= H((TID_i \parallel SID_k) \oplus \alpha_1 \oplus \sigma_i \\ &\qquad \oplus H((ID_i \parallel PW_i) \oplus b)).\end{aligned}$$

Step 2. Now, Attacker chooses password PW_i^{guess} for user U_i to find the correct password PW_i. Then, attacker checks the correctness whether $\alpha_2 = H((TID_i \parallel SID_k) \oplus \alpha_1 \oplus \sigma_i \oplus H((ID_i \parallel PW_i^{guess}) \oplus b))$, where ID_i is the correct user identity by using identity guessing attack and all other parameters of α_2 is known to the attacker except password PW_i.

Step 3. An attacker then repeats the above process until the correct password is obtained. After some guessing, an attacker can find out the correct password. Thus, Li et al.'s scheme is vulnerable to off-line password guessing attack.

5.3 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. Under our assumption, Li et al.'s scheme can not resist user impersonation attack as follows:

Step 1. Attacker can compute $PWB_i^a = H((ID_i \parallel PW_i) \oplus b)$, where ID_i, PW_i is the user's correct identity and password by using off-line identity and password guessing attack respectively. Attacker further computes $\alpha_1^a = \sigma_i \oplus PWB_i^{'} \oplus N_1^a$ and $\alpha_2^a = H((TID_i \parallel SID_k) \oplus N_1^a)$, where N_1^a is the random number generated by the attacker and attacker knows σ_i, b from user's smart card memory by monitoring power consumption.

Step 2. Then, attacker sends forged login message $\{TID_i,\ \alpha_1^a,\ \alpha_2^a\}$ to the service provider server S_k. It can be easily proved that the sending login message by an attacker is valid to the service provider server S_k. Then, service provider server S_k sends login message $\{TID_i,\ \alpha_1^a,\ \alpha_2^a,\ SID_k,\ \beta_1,\ \beta_2\}$ to the control server CS after computing β_1, β_2.

Step 3. After receiving login message from service provider server S_k, control server checks the validity of user's TID_i and server's SID_k. If both hold, CS computes, $N_1^a = \alpha_1^a \oplus H(TID_i \parallel x)$ and checks the freshness of N_1^*. If it holds, CS further computes $\alpha_2^* = H((TID_i \parallel SID_k) \oplus N_1^a)$ and compares with computed α_2^* equals with received α_2^a. If it holds, then CS believes that the sending messages are authentic, otherwise terminates the connection. Then, CS sends $\alpha^{'}, \gamma_u$ to the smart card user U_i through service provider server S_k, where $\alpha^{'} = H(N_1^a) \oplus N_2^* \oplus N_3$ and $\gamma_u = H(H(TID_i \parallel x) \oplus SK)$.

Step 4. After receiving $\alpha^{'}, \gamma_u$ from CS through service provider server S_k, attacker derives $N_2^* \oplus N_3 = H(N_1^a) \oplus \alpha^{'}$ and computes session key $SK_a = H(N_1 \oplus N_2^* \oplus N_3)$ which is used for secure communication. Thus, Li et al.'s scheme is insecure against user impersonation attack.

5.4 Many Logged in Users' Attack

Many logged in users' attack can be successfully launched after successful performance of off-line identity guessing attack and off-line password guessing attack as described in Section 5. After getting correct password and identity of user U_i, attackers or non-registered user can successfully access the service of the server S_k as follows:

Step 1. Attacker can compute $PWB_i^a = H((ID_i \parallel PW_i) \oplus b)$, where ID_i, PW_i is the user's correct identity and password by using off-line identity and password guessing attack respectively. Then, computes $H(TID_i \parallel x) = \sigma_i \oplus PWB_i^a$.

Step 2. Now, Attacker chooses password PW_i^a and computes $\sigma_i^a = H(TID_i \parallel x) \oplus H((ID_i \parallel PW_i^a) \oplus b)$, where attacker keeps unchanged $H(TID_i), T_i$ which can be extracted from memory of smart card by monitoring power consumption.

Step 3. Then, attacker or non-registered users stores $\{\sigma_i^a, H(TID_i), T_i, H(\cdot)\}$ into memory of the smart card and it can be used by many attacker or non-registered users as a valid user.

Above attack proves that Li et al.'s scheme can not be used for practical implementation in terms of security. It is because, without stealing user's smart card, many non-registered users can acts as valid users.

6 Proposed Scheme

In this paper, We have shown that Sood's scheme and Li et al.'s scheme are insecure against various attacks. To overcome these weaknesses, in this section, we proposed an efficient dynamic identity based remote user authentication in multi-server environment using smart card. It can be assumed that control server CS is a trusted authority. The proposed scheme consists of four phases namely registration phase, login phase, authentication and session key agreement phase and password change phase. All these proposed phases are discussed as below:

6.1 Registration Phase

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

Server Registration Phase. In this phase, service provider server S_k selects his/her desired identity SID_k and submits it to control server CS over a secure channel. After receiving SID_k from S_k, CS computes $P_k = H(SID_k \parallel x)$ and sends it to the server S_k through secure channel and S_k keeps it as secret, where x is the secret key of control server CS.

User Registration Phase. Whenever a new user wants to get services from the service provider server, first he/she has to register with the control server CS. So, the user chooses his/her desired identity ID_i and password PW_i and generates a random nonce b. Afterwards, user computes $PWR_i = H(PW_i \oplus b)$, where $H(\cdot)$ is the secure one-way hash function like secure MD5 and sends ID_i, PWR_i to control server through secure channel for the registration. After receiving a registration message from user U_i, CS generates a random nonce y_i for each user U_i and computes $CID_i = H(ID_i \oplus y_i \oplus x)$ such that CID_i will be unique for each user U_i like bank account number. So, after computing CID_i for user U_i, CS checks whether the value of CID_i is exist in CS's database or not. If exists, CS chooses another random nonce y_i^* and computes again $CID_i = H(ID_i \oplus y_i^* \oplus x)$. CS again verifies whether CID_i is exist or not in the database. If exists, then again computes CID_i with the new random nonce until CID_i will be unique, otherwise control server CS computes $REG_i = H(ID_i \parallel PWR_i \parallel CID_i)$ and $T_i = H(CID_i \parallel x) \oplus PWR_i$ and issues a smart card for U_i after storing $\{CID_i,\ REG_i,\ T_i,\ y_i,\ H(\cdot)\}$ into memory of user's smart card. After getting smart card, user U_i stores b into memory of smart card and uses it securely for taking services from S_k.

6.2 Login Phase

Whenever an existing user U_i wants to get the service(s) from the server S_k, first inserts his/her smart card into the card reader and submits ID_i^* and PW_i^*; and chooses server identity SID_k. Then, card reader computes $PWR_i^* = H(PW_i^* \oplus b)$ and $REG_i^* = H(ID_i^* \parallel PW_i^* \parallel CID_i)$; and checks whether REG_i^* equals stored REG_i holds or not. If the verification holds, it implies $ID_i^* = ID_i$ and $PW_i^* = PW_i$. Then, smart card derives $L_1 = T_i \oplus PWR_i^*$ and generates random numbers N_1, N_2 and further computes $N_3 = N_1 \oplus N_2$, $L_2 = N_2 \oplus PWR_i^*$ and $L_3 = H(L_1 \parallel SID_k \parallel N_1 \parallel L_2 \parallel N_3)$ and sends login request message $\{CID_i,\ SID_k,\ T_i,\ L_3,\ L_2,\ N_3\}$ to control server CS.

6.3 Authentication Phase

After receiving the login request message $\{CID_i,\ SID_k,\ T_i,\ L_3,\ L_2,\ N_3\}$, control server first checks the format of CID_i and SID_k. If it is valid then computes $A_1 = H(CID_i \parallel x)$ and derives $PWR_i^{'} = T_i \oplus A_1$, $N_2^{'} = L_2 \oplus PWR_i^{'}$ and $N_1^{'} = N_3 \oplus N_2^{'}$. Further computes $L_3^{'} = H(A_1 \parallel SID_k \parallel N_1^{'} \parallel L_2 \parallel N_3)$ and verifies whether computed $L_3^{'}$ equals with received L_3. If it does not hold, CS terminates the session, otherwise CS believes that the user U_i is authentic and also believes that SID_k is the registered identity of service provider server S_k. Then, control server generates a random number N_4 and computes $A_2 = H(SID_k \parallel x)$, $A_3 = A_2 \oplus N_4$, $N_5 = N_1{'} \oplus N_4$ and $A_4 = H(A_2 \parallel N_4 \parallel N_1 \parallel CID_i)$. Then, CS sends $\{CID_i,\ A_4,\ A_3,\ N_5\}$ to the service provider server S_k of the corresponding identity SID_k through public channel.

After receiving messages $\{CID_i,\ A_4,\ A_3,\ N_5\}$ from CS, server S_k derives $N_4^{'} = P_k \oplus A_3$ and $N_1^{'} = N_4^{'} \oplus N_5$ and computes $A_4^{'} = H(P_k \parallel N_4^{'} \parallel N_1^{'} \parallel CID_i)$. Then, server S_k compares $A_4^{'}$ with received A_4. This equivalency authenticates the legitimacy of the control server CS and user U_i. Further, server S_k generates random number N_6 and computes $N_7 = N_1^{'} \oplus N_6$, $SK_s = H(SID_k \parallel CID_i \parallel N_6 \parallel N_1^{'})$, $A_5 = H(SK_s \parallel N_6)$ and sends $\{SID_k,\ A_5,\ N_7\}$ to the smart card user U_i through public channel.

After receiving messages $\{SID_k,\ A_5,\ N_7\}$ from the server S_k, the smart card derives $N_6^{'} = N_7 \oplus N_1$ and computes $SK_u = H(SID_k \parallel CID_i \parallel N_6^{'} \parallel N_1)$, $A_5^{'} = H(SK_u \parallel N_6^{'})$. Then, smart card compares computed $A_5^{'}$ equals with received A_5. This equivalency authenticates the legitimacy of the service provider server S_k. It can be easily shown that $SK_u = SK_s$ which is the common secret session key between user U_i and service provider server S_k.

6.4 Password Change Phase

This phase invokes when user U_i wants to change his/her password. U_i inserts the smart card into the card reader and submits ID_i^* and PW_i^*. Then, card reader computes $PWR_i^* = H(PW_i^* \oplus b)$ and $REG_i^* = H(ID_i^* \parallel PW_i^* \parallel CID_i)$; and checks whether REG_i^* equals stored REG_i holds or not. If it holds positively, U_i enters a new password PW_i^{new}. Then card reader computes $PWR_i^{new} = H(PW_i^{new} \oplus b)$, $REG_i^{new} = H(ID_i^* \parallel PWR_i^{new} \parallel CID_i)$ and $T_i^{new} = T_i \oplus PWR_i^* \oplus PWR_i^{new}$ and stores REG_i^{new}

and T_i^{new} instead of REG_i and T_i respectively into the memory of the smart card. Thus, U_i can change the password without taking any assistance from control server or service provider server S_k.

7 Cryptanalysis of The Proposed Scheme

This section describes cryptanalysis of the proposed scheme. To cryptanalyze the proposed scheme, it can be assumed that an attacker could obtain the secret values stored in the smart card by monitoring the power consumption [12, 17] and can intercept all communicating messages between the user, service providing server and the control server. Under these assumption, we will show that the proposed scheme resists different possible attacks related to remote user authentication.

7.1 Off-line Identity Guessing Attack

After getting secret values $\{CID_i,\ REG_i,\ T_i,\ y_i,\ H(\cdot)\}$ from user's smart card memory and login request message $\{CID_i,\ SID_k,\ T_i,\ L_3,\ L_2,\ N_3\}$, an attacker attempts to derive or guess user's identity ID_i. To obtain user's correct identity ID_i from CID_i, attacker has to guess x and ID_i simultaneously which is not possible in polynomial time, where x is the secret key of the control server. So, the proposed scheme resists off-line identity guessing attack.

7.2 Off-line Password Guessing Attack

After getting secret values $\{CID_i,\ REG_i,\ T_i,\ y_i,\ H(\cdot)\}$ from user's smart card memory and login request message $\{CID_i,\ SID_k,\ T_i,\ L_3,\ L_2,\ N_3\}$, an attacker attempts to derive or guess user's password PW_i in off-line mode. To get user's correct password, attackers has to guess either two secret parameters at a time which is not possible in polynomial time or has to solve inversion of cryptographic hash function which is also computationally hard. So, the proposed scheme is secure against off-line password guessing attack.

7.3 Privileged Insider Attack

The proposed scheme is secure against privileged insider attack because, user U_i provides PWR_i which equals with $H(PW_i \oplus y)$ instead of PW_i to the control server CS. As a result, system manager or privileged insider of the server can not derive valid user's password. So, the proposed scheme resists privileged insider attack.

7.4 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 cannot impersonate as the legitimate user by forging the login request message even if the attacker can extract the secret values $\{CID_i,\ REG_i,\ T_i,\ y_i,\ H(\cdot)\}$ stored in the users smart card, because the attacker cannot compute the valid login request message $\{CID_i,\ SID_k,\ T_i,\ L_3,\ L_2,\ N_3\}$ without knowing the secret password PW_i of valid user U_i, control server secret key x and valid user identity ID_i. If the attacker wants to get these secret parameters, he/she must have to solve the inversion of cryptographic hash function which is computationally hard. So, the proposed scheme is secure against user impersonation attack.

7.5 Many Logged-in Users' Attack

The proposed scheme is secure against many logged-in users' attack because even if an attacker gets user's smart card then he/she has no way to derive or guess user's correct password PW_i, user's identity ID_i and server secret key x as described in Section 7. If the attacker wants to get the control server secret key x, user's password PW_i and user's identity ID_i, he/she must have to solve the inversion of cryptographic one-way hash function which is computationally hard. So the proposed scheme resists many logged-in users' attack.

7.6 Smart Card Stolen Attack

We assume 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 $\{CID_i,\ REG_i,\ T_i,\ y_i,\ H(\cdot)\}$ from the user's smart card. We also assume that attacker stores the $i-th$ login message $\{CID_i,\ SID_k,\ T_i,\ L_3,\ L_2,\ N_3\}$ of the user U_i. After getting all these parameters such as login message and smart card parameters, it is hard to derive user's password PW_i, identity ID_i and server secret key x by the attacker. As a result, attacker can not create the valid login message even after getting the valid user's smart card parameters. So, the proposed scheme is secure against smart card stolen attack.

7.7 Session Key Recovery Attack

In the proposed scheme, session key depends upon the difficulty of cryptographic one-way hash function and the random number N_1 and N_6. There is no way for an attacker to compute random number N_1 and N_6 from the known parameters that is from all communicating message of the proposed scheme. So the proposed scheme resists session key recovery attack.

Table 2: Comparison of computation cost of proposed scheme with related schemes

	[19]	[13]	Proposed protocol
Login Phase	$4T_h$	$3T_h$	$3T_h$
Authentication Phase	$17T_h + 1T_m$	$15T_h$	$9T_h$
Total	$21T_h + 1T_m$	$18T_h$	$12T_h$

Table 3: Comparison of communication and storage cost of proposed scheme with related schemes

	[19]	[13]	Proposed protocol
Storage Cost	512 bits	640 bits	768 bits
Communication Cost	1920 bits	1920 bits	1664 bits

Table 4: Security attack comparison of the proposed scheme with related schemes

	[19]	[13]	Proposed protocol
Off-line Identity Guessing Attack	YES	YES	NO
Off-line Password Guessing Attack	YES	YES	NO
Privileged Insider Attack	YES	NO	NO
User Impersonation Attack	YES	YES	NO
Many Logged In Users' Attack	YES	YES	NO
Session Key Recovery Attack	YES	NO	NO

8 Performance Analysis of the Proposed Scheme

In this section, we evaluated the performance of proposed scheme comparing with both the Sood's scheme and Li. et al's scheme. We have compare login and authentication phases of proposed scheme with both Sood's scheme and Li et al's scheme, because these phases are used frequently. Table 2 shows the computation over head and Table 3 shows the communication and storage cost of proposed scheme and both the related [13, 19] scheme. In Table 2, T_h is the time required for hashing operation and T_m is the time required for multiplication operation. Though, proposed scheme resists different possible attacks of both the related schemes, in spite of the proposed scheme which provides better computation cost than the related schemes.

It can be reasonably assumed that the length of ID_i, PW_i, SID_j, $h(\cdot)$ and random nonce returns 128 bits. The communication cost (capacity of transmitting message) of proposed scheme, Sood's [19] scheme and Li et al.'s [13] scheme are 1664 $bits = (13 \times 128)$, 1920 $bits = (15 \times 128)$ and 1920 $bits = (15 \times 128)$ respectively for each transaction. Also the storage cost (stored into the memory of smart card) takes almost same bits of proposed scheme and related schemes that is 768 $bits$, 512 $bits$ and 512 $bits$ respectively. Table 4 shows that their scheme is insecure against different possible attacks. Further proposed scheme provides strong authentication against different attacks described in Section 7. After resisting all possible attacks of related scheme, the proposed scheme provides low computational and communication cost than others related schemes. Hence the proposed scheme is more efficient and secure than both Sood's scheme and Li. et al's scheme.

9 Conclusion

We have shown that both Sood's [19] and Li et al.'s [13] schemes have security weaknesses described in Section 3 and Section 5 respectively. To overcome these weaknesses, we have proposed an Efficient Dynamic ID Based Remote User Authentication Scheme in Multi-server Environment using smart card. Further, we have shown that the proposed scheme using smart card which is more efficient in terms of computational and communication cost than related schemes. Additionally, the proposed scheme provides password change phase without taking any assistance of the control server and also provides strong mutual authentication. Cryptanalysis of the proposed scheme shows that the authentication system is more authentic, secure and efficient than related schemes published earlier. In future, we can incorporate biometric features with password to provide high security system and also try to analyze the security analysis of the proposed protocol using BAN logic.

References

[1] R. Amin, T. Maitra, D. Giri, "An improved efficient remote user authentication scheme in multi-server environment using smart card", *International Journal of Computer Applications*, vol. 69, no. 22, pp. 1–6, 2013.

[2] C. C. Chang and T. F. Cheng, "A robust and efficient smart card based remote login mechanism for multi-server architecture", *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 8, pp. 4589–4602, 2011.

[3] C. C. Chang, J. S. Lee, "An efficient and secure multi-server password authentication scheme using

smart cards", in *Proceedings of the International Conference on Cyberworlds*, pp. 417–422, 2004.

[4] Te-Yu Chen, C. C. Lee, M. S. Hwang, J. Ke Jan, "Towards secure and efficient user authentication scheme using smart card for multi-server environments," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1008–1032, Nov. 2013.

[5] T. H. Feng, C. H. Ling, and M. S. Hwang, "Cryptanalysis of Tan's improvement on a password authentication scheme for multi-server environments," *International Journal of Network Security*, vol. 16, no. 4, pp. 318–321, 2014.

[6] W. Ford and B. S. Kaliski, "Server-assisted generation of a strong secret from a password", in *Procedding of IEEE 9th International Workshop Enabiling Technology*, pp. 176–180, Washington, June 2000.

[7] D. He, W. Zhao, and S. Wu, "Security analysis of a dynamic id-based authentication scheme for multiserver environment using smart cards", *International Journal of Network Security*, vol. 15, no. 5, pp. 350–356, 2013.

[8] 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 Interface*, vol. 31, no. 6, pp. 1118–1123, 2009.

[9] L. Hu, X. Niu, and Y. Yang, "An efficient multiserver password authenticated key agreement scheme using smart cards", in *Proceedings of International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, pp. 903–907, Apr. 2007.

[10] D. P. Jablon, "Password authentication using multiple servers", in *Proceedings of RSA Security Conference*, pp. 344–360, London, Apr. 2001.

[11] W. S. Juang, "Efficient multi-server password key agreemented key exchange using smart cards", *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 251–255, 2004.

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

[13] C. T. Li, C. Y. Weng, and C. I Fan, "Two-factor user authentication in multi-server networks", *International Journal of Security and Its Applications*, vol. 6, no. 2, pp. 261–267, 2012.

[14] H. Li, I. C. Lin, M. S. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks", *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.

[15] Y. P. Liao and S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multiserver environment", *Computer Standard and Interface*, vol. 31, no. 1, pp. 24–29, 2009.

[16] I. C. Lin, M. S. Hwang and L. H. Li, "A new remote user authentication scheme for multi-server archicture", *Future Generation Computer System*, vol. 19, no. 1, pp. 13–22, 2003.

[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] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[19] S. K. Sood, "Dynamic identity based authentication protocol for two-server architecture", *Journal of Information Security*, vol. 3, pp. 326–334, 2012.

[20] S. K. Sood, A. K. Sarje, and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture", *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 609–618, 2011.

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

Ruhul Amin received his B.Tech and M.Tech Degree from West Bengal University of Technology in computer science and engineering department in 2009 and 2013 respectively. He has qualified GATE in 2011 in computer science. Currently, he is a lecturer of a polytechnic college. He has published three (3) international journal on the topic of remote user authentication. His research interest are remote user authentication and security in wireless sensor network.