An Improved and Secure Smart Card Based Dynamic Identity Authentication Protocol

Sandeep Kumar Sood

Department of Computer Science & Engineering, G.N.D.U Regional Campus, Gurdaspur, India
(Email: san1198@gmail.com)

(Received Nov. 1, 2010; revised and accepted Dec. 27, 2010)

Abstract

Password is the most commonly used authentication technique in smart card based authentication protocols. During communication, the static identity based authentication protocols leaks out the user's authentication messages corresponding to static identity to the attacker. Therefore, the attacker can trace and identify the different requests belonging to the same user. On the other hand, the dynamic identity based authentication protocols provide multi-factor authentication based on the identity, password, smart card and hence more suitable to e-commerce applications. In 2008, Liu et al. proposed a nonce based mutual authentication scheme using smart cards. In 2009, Sun et al. demonstrated man-in-the-middle attack on Liu et al.'s scheme. However, we found that Liu et al.'s scheme is also vulnerable to stolen smart card attack. This paper presents a new dynamic identity based authentication scheme that uses the nonce and timestamp at the same time to resolve the aforementioned problems, while keeping the merits of Liu et al.'s scheme. The aim of this paper is to provide a dynamic identity based secure and computational efficient authentication protocol with user's anonymity using smart cards. It protects the user's identity in insecure communication channel and hence can be applied directly to e-economic applications. Security analysis proved that the proposed protocol is secure and practical.

Keywords: Authentication protocol, cryptography, dynamic identity, password, smart card, stolen smart card attack

1 Introduction

Smart cards have been widely used in many e-commerce applications and network security protocols due to their low cost, portability, efficiency and the cryptographic properties. Smart card stores some sensitive data corresponding to the user that assist in user authentication. The user (card holder) inserts his smart card into a card reader machine and submits his identity and password. Then smart card and card reader machine perform some cryptographic operations using submitted arguments and the data stored inside the memory of smart card to verify the authenticity of the user.

A number of static identity based remote user authentication protocols have been proposed to improve security, efficiency and cost. The user may change his password but can not change his identity in password authentication protocols. During communication, the static identity leaks out partial information about the user's authentication messages to the attacker. Most of the password authentication protocols are based on static identity and the attacker can use this information to trace and identify the different requests belonging to the same user. On the other hand, the dynamic identity based authentication protocols provide multi-factor authentication based on the identity, password, smart card and hence more suitable to e-commerce applications. The aim of this paper is to provide a dynamic identity based secure and computational efficient authentication protocol with user's anonymity using smart cards.

2 Literature Review

In 1981, Lamport [4] proposed a password based authentication scheme that authenticates the remote users over an insecure communication channel. Lamport's scheme eliminates the problems of password table disclosure and communication eavesdropping. Since then, a number of approaches to remote user authentication have been proposed to improve security, efficiency and cost.

In 1999, Yang and Shieh [15] proposed a timestamp based password authentication scheme using smart card. In their scheme, the users are allowed to choose and change their passwords freely and the remote server does not require to keeps the password table or verification table. The verification and authentication data pertaining to the user is generated and provided by the user to the server during authentication phase. However, many researchers [1, 2, 7] demonstrated vulnerability of Yang and Shieh's scheme to forged login attack and other attacks. In 2003, Shen *et al.* [7] proposed an improved scheme to preclude the weaknesses of Yang and Shieh's scheme that can resist the forged login attack and also provides mutual authentication between the client and the server to protect it from the forged server attack. In 2005, Yoon *et al.* [16] demonstrated that Shen *et al.*'s scheme was still vulnerable to forged login attack. The attacker can intercept the legitimate user's login request message and register the new smart card with the server using the computed identity from intercepted login request messages to carry out the forged login attack. In 2008, Liu *et al.* [5] proposed a nonce based mutual authentication scheme using smart cards and claimed that their scheme can withstand the existing forged attacks. In 2009, Sun et al. [13] demonstrated the man-in-the-middle attack on Liu *et al.*'s scheme. In this paper, we found that the Liu *et al.*'s scheme is also vulnerable to stolen smart card attack. In 2009, Xu *et al.* [14] proposed an exponential based smart card authentication scheme and claimed that it can resist the various feasible attacks. In 2010, Sood *et al.* [9] found that Xu *et al.*'s scheme is also found to be vulnerable to forgery attack and proposed an improved scheme. In 2009, Sood et al. [8] proposed a dynamic identity based single password anti-phishing protocol that is secure against different possible attacks. In this protocol, the user machine's browser generates a dynamic identity and a dynamic password for each login request to the server. The dynamic identity and dynamic password will be different for the same user in different sessions of the SSL protocol. The user can use a single password for different online accounts and that password cannot be detected by any of the malicious server or the attacker. In 2010, Sood *et al.* also proposed dynamic identity based authentication protocols for single server [10] and for multi-server [11] architecture. In 2011, Sood *et al.* [12] proposed an inverse cookie and dynamic identity based virtual authentication protocol in which the cookies are not being stored on the trustworthy machines instead the cookies are being stored on those machines from where the user failed to login.

The rest of this paper is organized as follows. In Section 3, a brief review and cryptanalysis of Liu *et al.*'s scheme is given. In Section 4, the improved scheme is proposed. The security analysis of the proposed improved scheme is presented in Section 5. The comparison of the cost and functionality of the proposed scheme with the other related schemes is shown in Section 6. Section 7 concludes the paper.

3 Liu *et al.*'s Scheme

3.1 Main phases of Liu *et al.*'s scheme

In this section, we examine the remote user authentication scheme proposed by Liu *et al.* [5] in 2008. Liu *et al.*'s scheme consists of three phases viz. initialization phase, registration phase, login and authentication phase.

3.1.1 Initialization Phase

Key Information Center (KIC) generates secret parameters corresponding to the user, store them on the smart card and issue the smart card to the user. KIC is also responsible to change the passwords of registered users. It generates two large prime numbers p and q and computes $n = p.q$. Then it chooses a public key e and finds a corresponding secret key d that satisfies $e.d \equiv 1 \bmod (p-1).(q-1)$. The secret key d is sent to the server S over a secure communication channel. Afterwards, KIC finds an integer g that is a primitive element in $GF(p)$ and $GF(q)$, where g is the public parameter of KIC. Finally, KIC sends the parameters n, e and g to the server S.

3.1.2 Registration Phase

A user U_i has to submit his password P_i to KIC for registration over a secure communication channel. KIC selects an identity ID_i corresponding to the user U_i. Then KIC computes $CID_i = H(ID_i \oplus d)$, $S_i \equiv ID_i^d \bmod n$ and $h_i \equiv g^{P_i.d} \bmod n$, where $H()$ is a one-way hash function. Afterwards, KIC issues the smart card containing secret parameters $(n, e, g, ID_i, CID_i, S_i, h_i)$ to the user U_i through a secure communication channel.

3.1.3 Login and Authentication Phase

The user U_i inserts his smart card into a card reader to login on to the server S and submits his identity ID_i^* and password P_i^*. The smart card compares the identity ID_i^* with the stored value of ID_i in its memory to verify the legitimacy of the user. Then the smart card computes $SID_i = H(CID_i)$ and sends the login request message $M1 = \{ID_i, SID_i\}$ to the service provider server S. The service provider server S computes $CID_i = H(ID_i \oplus d)$ and compares $H(CID_i)$ with the received value of SID_i. If they are not equal, the server S rejects the login request and terminates this session. Otherwise the server S stores the parameters ID_i, SID_i and chooses nonce value N_S as a challenge to the user U_i. The server S computes $S_N = N_S \oplus CID_i$ and sends the message $M2 = \{S_N\}$ back to the smart card of the user U_i. On receiving the message $M2$, the smart card chooses a random nonce value N_C and computes $N_S = S_N \oplus CID_i$, $X_i \equiv g^N c.^{P_i} \bmod n$ and $Y_i \equiv S_i.h_i^N c.^N s \bmod n$. Then, the smart card sends the message $M3 = \{X_i, Y_i\}$ to the server S. On receiving the message $M3$, the server S checks whether the equation $Y_i^e \equiv ID_i.\ X_i^N s \bmod n$ holds. If it holds, the server S accepts the login request and computes $Z_i \equiv (H(CID_i.X_i))^d \bmod n$ and sends the message $M4 = \{Z_i\}$ back to the smart card. On receiving the message $M4$, the smart card checks whether the equation $Z_i^e \equiv H(CID_i.X_i) \bmod n$ holds or not. This equivalency authenticates the legitimacy of the service provider server S and the login request is accepted else the connection is interrupted.

3.2 Cryptanalysis of Liu *et al.*'s Scheme

Liu *et al.* [5] claimed that their protocol can resist various known attacks. Sun *et al.* [13] demonstrated man-in-the-middle attack on Liu *et al.*'s protocol [5]. However in this paper, we found that Liu *et al.*'s protocol [5] is also found to be flawed for stolen smart card attack.

3.2.1 Stolen Smart Card Attack

A user U_i may lose his smart card, which is found by an attacker or an attacker steals the user's smart card. An attacker can extract the stored values through some technique like by monitoring their power consumption and reverse engineering techniques as pointed out by Kocher *et al.* [3] and Messerges *et al.* [6].

1) The attacker can extract the (n, e, g, ID_i, CID_i, S_i, h_i) parameters from the memory of a smart card.

2) Now the attacker computes $SID_i = H(CID_i)$ and sends the login request message $M1 = \{ID_i, SID_i\}$ to the service provider server S.

3) The service provider server S computes and verifies the received value of SID_i.

4) Then the service provider server S chooses random nonce value N_S as a challenge to the smart card of the user U_i, computes $S_N = N_S \oplus CID_i$ and sends the message $M2 = \{S_N\}$ back to the smart card of the user U_i.

5) Afterwards, the smart card chooses a random nonce value N_C, computes $N_S = S_N \oplus CID_i$, $X_i^* \equiv h_i^{N_C.e} \bmod n$ and $Y_i \equiv S_i.h_i^{N_C.\overset{N}{S}} \bmod n$.

$$\begin{aligned} X_i^* &\equiv h_i^{N_C.e} \bmod n \\ &\equiv (g^{P_i.d} \bmod n)^{N_C.e} \bmod n \textit{ because } h_i \\ &\equiv g^{P_i.d} \bmod n \\ &\equiv (g^{P_i.d.N_C.e} \bmod n) \bmod n \\ &\equiv (g^{P_i.N_C} \bmod n) \bmod n \textit{ because } g^{e.d} \bmod n \\ &\equiv 1 \\ &\equiv (g^{N_C.P_i} \bmod n) \bmod n \\ &\equiv X_i \end{aligned}$$

6) Then the smart card sends the message $M3 = \{X_i, Y_i\}$ to the server S.

7) The server S checks and verifies that the equation $Y_i^e \equiv ID_i.X_i^N s \bmod n$ holds.

8) Then the server S computes $Z_i \equiv (H(CID_i.X_i))^d \bmod n$ and sends the message $M4 = \{Z_i\}$back to the smart card of the user U_i.

9) Now the attacker masquerading as the user U_i has authenticated itself to the service provider server S without knowing the password of the user U_i.

10) That means once an attacker gets the smart card of the user U_i,he can masquerade as a legitimate user U_i by authenticating itself to the server S without knowing the password of the user U_i corresponding to his smart card.

4 Dynamic Identity Based Smart Card Authentication Protocol

In this section, we describe a new remote user authentication scheme which resolves the above security flaws of Liu *et al.*'s [5] scheme. Figure 1 shows the entire protocol structure of the new authentication scheme. The proposed protocol consists of four phases viz. registration phase, login phase, verification and session key agreement phase and password change phase.

4.1 Registration Phase

A user U_i has to submit his identity ID_i and password P_i to the server S via a secure communication channel to register itself to the server S.

Step 1: $U_i \rightarrow S : ID_i, P_i$

The server S computes the security parameters $Z_i \equiv g^{(ID_i|P_i)+H(P_i)} \bmod n$, $B_i \equiv g^{(ID_i|x|y_i)+H(P_i)} \bmod n$ and $C_i \equiv g^{x+y_i+P_i} \bmod n$, where n is large prime number and g is a primitive element in $GF(n)$. The server S chooses its secret key x and $H()$ is a one-way hash function. The server S also computes $A_i \equiv g^{(ID_i|x|y_i)+y_i} \bmod n$ for each user and stores $y_i \oplus x$ corresponding to A_i in its database. The server S chooses the value of y_i corresponding to each user in such a way so that the value of A_i must be unique for each user. Then the server S issues the smart card containing security parameters $(Z_i, B_i, C_i, n, g, H())$ to the user U_i.

Step 2: $S \rightarrow U_i$: **Smart card**

4.2 Login Phase

A user U_i inserts his smart card into a card reader to login on to the server S and submits his identity ID_i^* and password P_i^*. The smart card computes $Z_i^* \equiv g^{(ID_i^*|P_i^*)+H(P_i^*)} \bmod n$ and compares it with the stored value of Z_i in its memory to verify the legitimacy of the user U_i.

Step 1: Smart card checks $Z_i^*? = Z_i$

After verification, the smart card computes $B_i' \equiv B_i g^{-H(P_i)} \bmod n \equiv g^{(ID_i|x|y_i)} \bmod n$, $C_i' \equiv C_i g^{-P_i} \bmod n \equiv g^{x+y_i} \bmod n$, $D_i \equiv B_i'. \ C_i' \bmod n \equiv g^{(ID_i|x|y_i)+x+y_i} \bmod n$, $E_i \equiv g^{w+H(B_i'|T)} \bmod n$ and $M_i = H(B_i'|C_i'|T)$, where smart card chooses $w \in_R Z_n^*$ and T is current time stamp of the smart card. Then the smart card sends the login request message (D_i, E_i, M_i, T) to the server S.

Step 2: Smart card → S : D_i, E_i, M_i, T

4.3 Verification and Session Key Agreement Phase

After receiving the login request from the user U_i,the service provider server S checks the validity of timestamp T by checking $(T' - T) \leq \delta T$, where T' is current timestamp of the server S and is permissible time interval for a transmission delay. The server S computes $A_i' \equiv D_i g^{-x} \bmod n$ and compares A_i' with the stored values of A_i in its database.

Step 1: Server S checks $A_i'? = A_i$

If no match found, the server S rejects the login request and terminates this session. Otherwise, the server S extracts y_i from $y_i \oplus x$ corresponding to A_i from its database. Now the server S computes $B_i' \equiv A_i g^{-y_i} \bmod n \equiv g^{(ID_i|x|y_i)} \bmod n$, $C_i' \equiv g^{x+y_i} \bmod n, g^w \equiv E_i g^{-H(B_i'|T)} \bmod n$, $M_i' = H(B_i'|C_i'|T)$ and compares M_i' with the received values of M_i.

Step 2: Server S checks $M_i'? = M_i$

Now the server S chooses $m \in_R Z_n^*$ and acquires the current time stamp T'' and computes $G_i \equiv g^{B_i'+m} \bmod n$, $N_i \equiv g^{H(C_i'|T'')+m} \bmod n$ and sends the message (G_i, N_i, T'') back to the smart card of the user U_i.

Step 3: S → Smart card: G_i, N_i, T''

On receiving the message (G_i, N_i, T''), the user U_i's smart card checks the validity of timestamp T'' by checking $(T''' - T'') \leq \delta T$, where T''' is current time stamp of the smart card. Then the smart card extracts $g^m \equiv G_i g^{-B_i'} \bmod n$ and computes $N_i' \equiv g^{H(C_i'|T'')}.g^m \bmod n \equiv g^{H(C_i'|T'')+m} \bmod n$ and compares it with the received value of N_i to verify the legality of the service provider server S.

Step 4: Smart card checks $N_i'? = N_i$

This equivalency authenticates the legitimacy of the service provider server S and the login request is accepted else the connection is interrupted. Finally, the user U_i and the server S agree on the common session key as $S_k = H(g^w|g^m|B_i'|C_i'|T|T'')$.

4.4 Password Change Phase

The user U_i can change his password without the help of the server S. The user U_i inserts his smart card in to a card reader and enters his identity ID_i^* and password P_i^* corresponding to his smart card. The smart card computes $Z_i^* \equiv g^{(ID_i^*|P_i^*)+H(P_i^*)} \bmod n$ and compares the calculated value of Z_i^* with the stored value of Z_i in its memory to verifies the legitimacy of the user U_i. Once the authenticity of the card holder is verified then the user U_i can instruct the smart card to change his password. Afterwards, the smart card asks the card holder to resubmit a new password P_i' and then the smart card computes $Z_i^{new} \equiv g^{(ID_i|P_i')+H(P_i')} \bmod n$, $B_i^{new} \equiv B_i g^{-H(P_i)} g^{+H(P_i')} \equiv g^{(ID_i|x|y_i)+H(P_i')} \bmod n$ and $C_i^{new} \equiv C_i g^{-P_i} g^{+P_i'} \equiv g^{x+y_i+P_i'} \bmod n$. Afterwards, the smart card updates the values of Z_i, B_i and C_i stored in its memory with Z_i^{new}, B_i^{new} and C_i^{new}.

5 Security Analysis

Smart card is a memory card that uses an embedded micro-processor from smart card reader machine to perform required operations specified in the protocol. Kocher et al. [3] and Messerges et al. [6] pointed out that all existing smart cards can not prevent the information stored in them from being extracted by techniques such as by monitoring their power consumption. Some other reverse engineering techniques are also available for extracting information from the smart cards. That means once a smart card is stolen by an attacker, he can extract the information stored in it. A good password authentication scheme should provide protection from different possible attacks relevant to that protocol.

1) **Stolen smart card attack:**
In case a user U_i's smart card is stolen by an attacker, he can extract the information stored in its memory. An attacker can extract $Z_i \equiv g^{(ID_i|P_i)} +^{H(P_i)}$ mod$n$, $B_i \equiv g^{(ID_i|x|y_i)+H(P_i)} \bmod n$ and $C_i \equiv g^{x+y_i+P_i} \bmod n$ from the memory of smart card. Even after gathering this information, the attacker has to guess out ID_i and P_i correctly at the same time. It is not possible to guess out the two parameters correctly at the same time in real polynomial time. Therefore, the proposed protocol is secure against stolen smart card attack.

2) **Man-in-the-middle attack:**
In this type of attack, the attacker intercepts the messages send between the client and the server and replay these intercepted messages with in the valid time frame window. The attacker can act as client to server or vice-versa with recorded messages. In our proposed protocol, the attacker can intercept the login request message (D_i, E_i, M_i, T) from the user U_i to the server S. Then he starts a new session with the server S by sending a login request by replaying the login request message (D_i, E_i, M_i, T) with in the valid time frame window. The attacker can authenticate itself to the server S as well as to the legitimate user U_i but can not compute the session key $S_k = H(g^w|g^m|B_i'|C_i'|T|T'')$ because the attacker does not know the value of g^w, g^m, B_i' and C_i'. Therefore, the proposed protocol is secure against man-in-the-middle attack.

3) **Impersonation attack:**
In this type of attack, the attacker impersonates as the legitimate user and forges the authentication

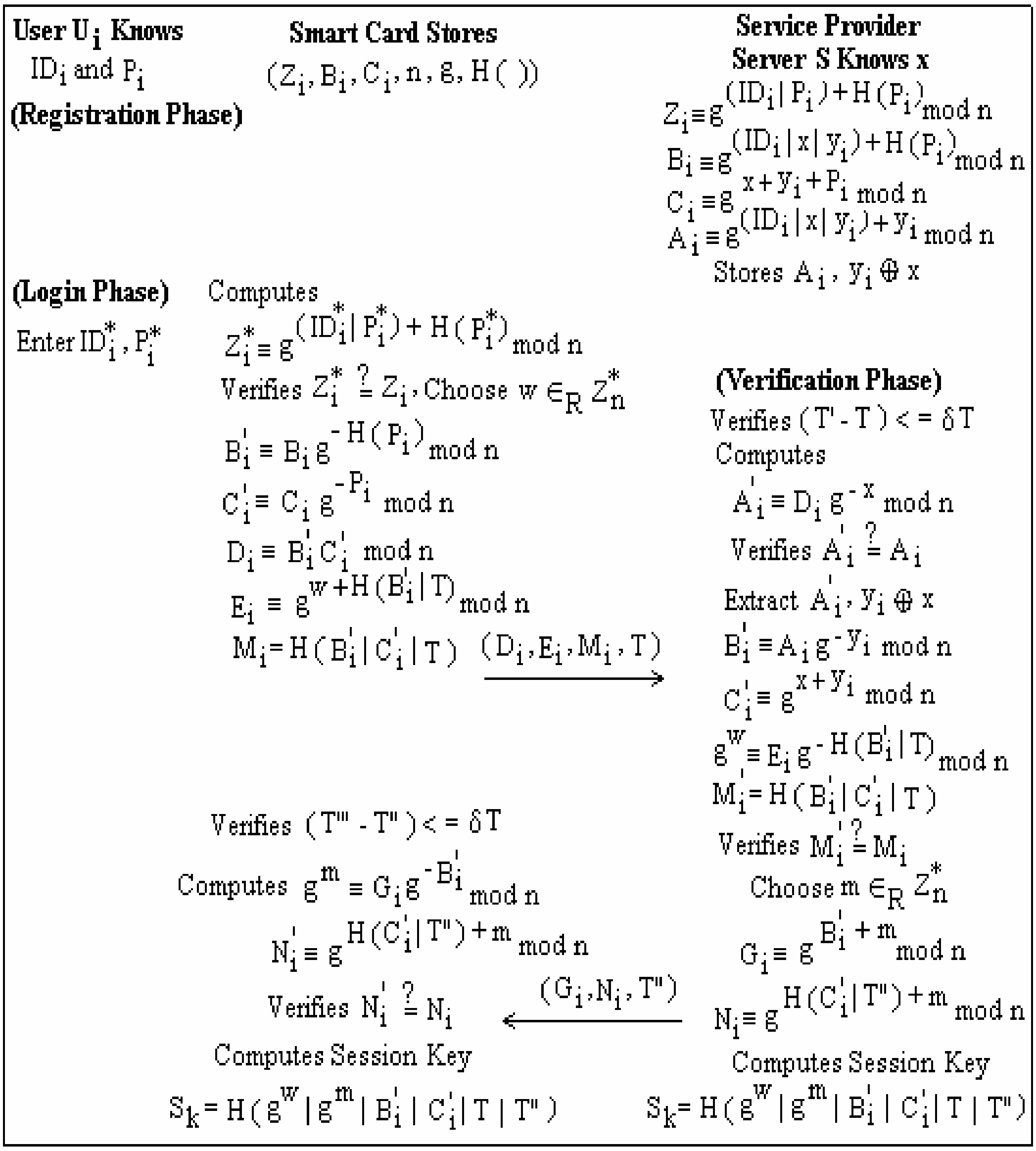
User U_i Knows
ID_i and P_i
(Registration Phase)
Smart Card Stores
$(Z_i, B_i, C_i, n, g, H(\ ))$
Service Provider Server S Knows x
$Z_i \equiv g^{(ID_i|P_i)+H(P_i)} \bmod n$
$B_i \equiv g^{(ID_i|x|y_i)+H(P_i)} \bmod n$
$C_i \equiv g^{x+y_i+P_i} \bmod n$
$A_i \equiv g^{(ID_i|x|y_i)+y_i} \bmod n$
Stores A_i, $y_i \oplus x$
(Login Phase)
Enter ID_i^*, P_i^*
Computes
$Z_i^* \equiv g^{(ID_i^*|P_i^*)+H(P_i^*)} \bmod n$
Verifies $Z_i^* \overset{?}{=} Z_i$, Choose $w \in_R Z_n^*$
$B_i' \equiv B_i g^{-H(P_i)} \bmod n$
$C_i' \equiv C_i g^{-P_i} \bmod n$
$D_i \equiv B_i' C_i' \bmod n$
$E_i \equiv g^{w+H(B_i'|T)} \bmod n$
$M_i = H(B_i'|C_i'|T)$
$\xrightarrow{(D_i, E_i, M_i, T)}$
(Verification Phase)
Verifies $(T' - T) <= \delta T$
Computes
$A_i' \equiv D_i g^{-x} \bmod n$
Verifies $A_i' \overset{?}{=} A_i$
Extract A_i', $y_i \oplus x$
$B_i' \equiv A_i g^{-y_i} \bmod n$
$C_i' \equiv g^{x+y_i} \bmod n$
$g^w \equiv E_i g^{-H(B_i'|T)} \bmod n$
$M_i' = H(B_i'|C_i'|T)$
Verifies $M_i' \overset{?}{=} M_i$
Choose $m \in_R Z_n^*$
$G_i \equiv g^{B_i'+m} \bmod n$
$N_i \equiv g^{H(C_i'|T'')+m} \bmod n$
$\xleftarrow{(G_i, N_i, T'')}$
Verifies $(T''' - T'') <= \delta T$
Computes $g^m \equiv G_i g^{-B_i'} \bmod n$
$N_i' \equiv g^{H(C_i'|T'')+m} \bmod n$
Verifies $N_i' \overset{?}{=} N_i$
Computes Session Key
$S_k = H(g^w|g^m|B_i'|C_i'|T|T'')$
Computes Session Key
$S_k = H(g^w|g^m|B_i'|C_i'|T|T'')$

Figure 1: Dynamic identity based smart card authentication protocol

message using the information obtained from the authentication scheme. The attacker can attempt to modify a login request message (D_i, E_i, M_i, T) into (D_i, E_i^*, M_i^*, T^*) so as to succeed in the authentication, where T^* is the attacker's current date and time. However, such a modification will fail in Step 2 of the verification and session key agreement phase because the attacker has no way of obtaining the value ID_i, P_i, x and y_i to compute the valid parameters E_i^* and M_i^*. Therefore, the proposed protocol is secure against impersonation attack.

4) **Malicious user attack:**
A malicious privileged user U_i having his own smart card can gather information like $Z_i \equiv g^{(ID_i|P_i)+H(Pi)} \bmod n$, $B_i \equiv g^{(ID_i|x|y_i)+H(P_i)} \bmod n$ and $C_i \equiv g^{x+y_i+P_i} \bmod n$ from the memory of smart card. This malicious user can not generate smart card specific value of $B_k \equiv g^{(ID_k|x|y_k)+H(P_k)} \bmod n$ and $C_k \equiv g^{x+y_k+P_k} \bmod n$ to masquerade as other legitimate user U_k to the service provider server S because the value of B_k and C_k is smart card specific and depends upon the value of ID_k, P_k, x and y_k. The malicious user does not have any method to calculate the value of ID_k, P_k, x and y_k. Therefore, the proposed protocol is secure against malicious user attack.

5) **Offline dictionary attack:**
In offline dictionary attack, the attacker can record messages and attempts to guess the user's identity ID_i and password P_i from recorded messages. The attacker first tries to obtain some user or server verification information such as $D_i \equiv B_i'.\ C_i' \bmod$

$n \equiv g^{(ID_i|x|y_i)+x+y_i} \bmod n$, $E_i \equiv g^{w+H(B_i'|T)} \bmod n$, $M_i = H(B_i'|C_i'|T)$, T, $G_i \equiv g^{B_i'+m} \bmod n$, $N_i \equiv g^{H(C_i'|T'')+m} \bmod n$, T'' and then tries to guess the ID_i, P_i, x and y_i by offline guessing. Even after gathering this information, the attacker has to guess ID_i, P_i, x and y_i correctly at the same time. In another option, the attacker requires valid smart card and then has to guess the identity ID_i and password P_i correctly at the same time. It is not possible to guess out two parameters correctly at same time. Therefore, the proposed protocol is secure against offline dictionary attack.

6) **Denial of service attack:**
In denial of service attack, the attacker updates password verification information from the memory of smart card to some arbitrary value so that the legitimate user can not login successfully in subsequent login request to the server. In the proposed protocol, the smart card checks the validity of user identity ID_i and password P_i before password update procedure. The attacker inserts the smart card into the smart card reader and has to guess the identity ID_i and password P_i correctly corresponding to the user U_i. Since the smart card computes $Z_i^* \equiv g^{(ID_i^*|P_i^*)+H(P_i^*)} \bmod n$ and compares it with the stored value of Z_i in its memory to verify the legality of the user before the smart card accepts the password update request. It is not possible to guess out identity ID_i and password P_i correctly at the same time in real polynomial time even after getting the smart card of the user. Therefore, the proposed protocol is secure against denial of service attack.

7) **Replay attack:**
In this type of attack, the attacker first listens to communication between the legitimate user and the server and then tries to imitate user to login on to the server by resending the captured messages transmitted between the legitimate user and the server. Replaying a message of one session into another session is useless because the user U_i's smart card and the server S uses current time stamp values T and T'' in each new session, which make the values of E_i, M_i and N_i dynamic and valid for small interval of time. Hence replaying old messages is useless and the proposed protocol is secure against message replay attack.

8) **Leak of verifier attack:**
In this type of attack, the attacker may be able to steal verification table from the server. If the attacker steals the verification table from the server, he can use the stolen verifiers to impersonate as a participant of the scheme. In the proposed protocol, the service provider server S knows secret x and stores $y_i \oplus x$ corresponding to the user's A_i value in its database. The attacker does not have any way to find out the value of x and hence can not calculate y_i from $y_i \oplus x$. Also the attacker can not calculate ID_i, x and y_i from $A_i \equiv g^{(ID_i|x|y_i)+y_i} \bmod n$. In case verifier is stolen by breaking into smart card database, the attacker does not have sufficient information to calculate the user's identity ID_i and password Pi. Therefore, the proposed protocol is secure against leak of verifier attack.

9) **Server spoofing attack:**
In server spoofing attack, the attacker can manipulate the sensitive data of legitimate users via setting up fake servers. The proposed protocol provides mutual authentication to withstand the server spoofing attack. Malicious server can not generate the valid value of $G_i \equiv g^{B_i'+m} \bmod n$ and $N_i \equiv g^{H(C_i'|T'')+m} \bmod n$ meant for the smart card because malicious server has to know the value of B_i' and C_i' to generate the valid values of G_i and N_i corresponding to user U_i's smart card. Therefore, the proposed protocol is secure against server spoofing attack.

10) **Online dictionary attack:**
In this type of attack, the attacker pretends to be legitimate client and attempts to login on to the server by guessing different words as password from a dictionary. In the proposed protocol, the attacker has to get the valid smart card and then has to guess the identity ID_i and password P_i corresponding to user U_i. Even after getting the valid smart card by any means, the attacker gets very few chances (maximum 3) to guess the identity ID_i and password P_i because the smart card gets locked after certain number of unsuccessful attempts. Moreover, it is not possible to guess out identity ID_i and password P_i correctly at the same time. Therefore, the proposed protocol is secure against online dictionary attack.

11) **Parallel session attack:**
In this type of attack, the attacker first listens to communication between the user and the server. After that, he initiates a parallel session to imitate legitimate user to login on to the server by resending the captured messages transmitted between the client and the server with in the valid time frame window. He can masquerade as the legitimate user U_i by replaying a login request message (D_i, E_i, M_i, T) with in the valid time frame window but can not compute the agreed session key $S_k = H(g^w|g^m|B_i'|C_i'|T|T'')$ because the attacker does not know the values of g^w, g^m, B_i' and C_i'. Therefore, the proposed protocol is secure against parallel session attack.

6 Cost and Functionality Analysis

An efficient authentication scheme must take communication and computation cost into consideration during user's authentication. The cost comparison of the proposed scheme with the most related smart card based authentication schemes is summarized in Table 1. Assuming

Table 1: Cost comparison among related smart card based authentication schemes

	Proposed Scheme	**Xu *et al.*[14]**	**Liu *et al.*[5]**	**Shen *et al.*[7]**	**Yang-Shieh *et al.*[15]**
E1	640 bits	512 bits	896 bits	896 bits	896 bits
E2	7*128 bits	8*128 bits	6*128 bits	10*128 bits	8*128 bits
E3	$4T_E + 1T_H + 1T_X$	$1T_E + 2T_H$	$2T_E + 1T_H + 1T_X$	$2T_E + 1T_H + 1T_X$	$2T_E$
E4	$6T_E + 5T_H$	$3T_E + 5T_H$	$3T_E + 2T_H + 1T_X$	$3T_E + 2T_H$	$2T_E + 1T_H$
E5	$6T_E + 4T_H + 1T_X$	$3T_E + 4T_H$	$2T_E + 3T_H + 2T_X$	$3T_E + 3T_H + 1T_X$	$2T_E + 1T_H$

Table 2: Functionality comparison among related smart card based authentication schemes

	Proposed Scheme	**Xu *et al.*[14]**	**Liu *et al.*[5]**	**Shen *et al.*[7]**	**Yang-Shieh *et al.*[15]**
Stolen Smart Card Attack	No	No	Yes	Yes	Yes
Man-in-the-Middle Attack	No	Yes	Yes	Yes	Yes
Forgery Attack	No	Yes	No	Yes	Yes
Identity Protection	Yes	No	No	No	No
Offline Dictionary Attack	No	Yes	No	No	No
Mutual Authentication	Yes	Yes	Yes	Yes	No
Session Key Agreement	Yes	Yes	No	No	No

that the identity ID_i, password P_i, random number (w or m), x, y_i, timestamp, nonce values are all 128-bit long and prime modular operation is 1024-bit length as in most of practical implementations. Moreover, we assume that the output of secure one-way hash function is 128-bit. Let T_H, T_E and T_X denote the time complexity for hash function, exponential operation and XOR operation respectively. Typically, time complexity associated with these operations can be roughly expressed as $T_E >> T_H >> T_X$. In the proposed protocol, the parameters stored in the smart card are Z_i, B_i, C_i, n, g and the memory needed (E1) in the smart card is $640(= 5 * 128)$ bits.

In the proposed protocol, the communication cost of authentication (E2) includes the capacity of transmitting message involved in the authentication scheme. The capacity of transmitting message $\{D_i, E_i, M_i, T\}$ and $\{G_i, N_i, T''\}$ is $896(= 7 * 128)$ bits. The computation cost of registration (E3) is the total time of all operations executed in the registration phase. The computation cost of registration (E3) is $4T_E + 1T_H + 1T_X$. The computation cost of the user (E4) and the service provider server (E5) is the time spent by the user and the service provider server during the process of authentication. Therefore, the computation cost of the user (E4) is $6T_E + 5T_H$ and that of the service provider server (E5) is $6T_E + 4T_H + 1T_X$. The functionality comparison of the proposed scheme with the related smart card based authentication schemes is summarized in Table 2. The proposed scheme has less computation cost of authentication (E2) as compared to latest scheme proposed by Xu *et al.* [14] in 2009. However, the proposed scheme requires some additional computation (E3, E4, E5) but it is highly secure as compared to the related schemes.

7 Conclusions

Corporate network and e-commerce applications require secure and practical smart card based remote user authentication solutions. In this paper, we presented a cryptanalysis of Liu *et al.*'s scheme and showed that their scheme is vulnerable to stolen smart card attack. Also Sun et al. demonstrated man-in-the-middle attack on Liu *et al.*'s scheme. An improved protocol was proposed that inherits the merits of Shen *et al.* and Liu *et al.*'s schemes and resists different possible attacks. The proposed protocol allows the user to choose and change the password at their choice and provides mutual authentication between the user and the server to protect it from forgery attack. It withstands the password guessing attack even if the attacker obtains the smart card of the user. The security of proposed protocol depends upon the discrete logarithm problem and one way hash function. The proposed protocol is highly secure as compared to related protocols but its computation cost is on higher side. Future scope in this work is to decrease the computation cost as well as to keep the security high like that of proposed dynamic identity based smart card authentication protocol.

References

[1] C. K. Chan and L.M. Cheng, "Cryptanalysis of timestamp-based password authentication scheme," *Computers and Security*, vol. 21, no. 1, pp. 74-76, 2002.

[2] L. Fan, J. H. Li, and H. W. Zhu, "An enhancement of timestamp-based password authentication scheme,"

Computers and Security, vol. 21, no. 7, pp. 665-667, 2002.

[3] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” *Proceedings of Crypto ’99*, pp. 388-397, Springer-Verlag, 1999.

[4] L. Lamport, “Password authentication with insecure communication,” *Communications of the ACM*, vol. 24, no. 11, pp. 770-772, 1981.

[5] J. Y. Liu, A. M. Zhou, and M. X. Gao, “A new mutual authentication scheme based on nonce, and smart cards,” *Computer Communications*, vol. 31, no. 10, pp. 2205-2209, 2008.

[6] 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.

[7] J. J. Shen, C. W. Lin, and M. S. Hwang, “Security enhancement for the timestamp-based password authentication scheme using smart cards,” *Computers and Security*, vol. 22, no. 7, pp. 591-595, 2003.

[8] S. K. Sood, A. K. Sarje, and K. Singh, “Dynamic identity based single password anti-phishing protocol,” *Security and Communication Networks*, Accepted, doi.wiley.com/10.1002/sec.169, Oct. 2009.

[9] S. K. Sood, A. K. Sarje, and K. Singh, “An improvement of Xu et al.’s authentication scheme using smart cards,” *Proceedings of the Third Annual ACM Bangalore Conference* , no. 15, pp. 1-5, Bangalore, India, 2010.

[10] S. K. Sood, A. K. Sarje, and K. Singh, “Secure dynamic identity based remote user authentication scheme,” *Sixth International Conference on Distributed Computing and Internet Technology*, LNCS 5966, pp. 224-235, Springer-Verlag, 2010.

[11] 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*, http://dx.doi.org/10.1016/j.jnca.2010.11.011, 2010.

[12] S. K. Sood, A.K. Sarje, and K. Singh, “Inverse cookie based virtual password authentication protocol,” *International Journal of Network Security*, vol. 12, no. 3, pp. 292-302, 2011.

[13] D. Z. Sun, J. P. Huai, J. Z. Sun, and J. X. Li, “Cryptanalysis of a mutual authentication scheme based on nonce, and smart cards,” *Computer Communications*, vol. 32, no. 6, pp. 1015-1017, 2009.

[14] J. Xu, W. T. Zhu, and D. G. Feng, “An improved smart card based password authentication scheme with provable security,” *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723-728, 2009.

[15] W. H. Yang, and S. P. Shieh, “Password authentication scheme with smart cards,” *Computers and Security*, vol. 18, no. 8, pp. 727-733, 1999.

[16] E. J. Yoon, E. K. Ryu, and K. Y. Yoo, “Attacks on the Shen et al.’s timestamp-based password authentication scheme using smart cards,” *IEICE Transactions on Fundamentals*, vol. E88-A, no. 1, pp. 319-321, 2005.

Sandeep Kumar Sood received his M.Tech (Computer Science & Engineering) in 1999 from the Guru Jambheshwar University Hisar (Haryana), India. He has completed his Ph.D in the Department of Electronics and Computer Engineering at Indian Institute of Technology Roorkee, India. He is working as a lecturer in Guru Nanak Dev University Regional Campus, Gurdaspur (Punjab), India. His research interests include Authentication Protocols, Computer and Network Security, Cryptography and Computer Networks.