©2010 International Journal of Computer Applications (0975 – 8887)
Volume 1 – No. 8

An Improvement of Liou et al.'s Authentication Scheme using Smart Cards

Sandeep K. Sood
Electronics & Computer Engineering
Indian Institute of Technology
Roorkee, India

Anil K. Sarje
Electronics & Computer Engineering
Indian Institute of Technology
Roorkee, India

Kuldip Singh
Electronics & Computer Engineering
Indian Institute of Technology
Roorkee, India

ABSTRACT
In 2004, Das et al. proposed a dynamic identity based remote user authentication scheme. This scheme allows the users to choose and change their passwords freely and the server does not maintain any verification table. Das et al. claimed that their scheme is secure against stolen verifier attack, replay attack, forgery attack, dictionary attack, insider attack and identity theft. Unfortunately, many researchers demonstrated that Das et al.'s scheme is susceptible to various attacks. Furthermore, this scheme does not achieve mutual authentication and thus can not resist malicious server attack. In 2006, Liou et al. improved Das et al.'s scheme and claimed that the improved scheme achieves mutual authentication and is secure against aforementioned attacks. However, we found that Liou et al.'s scheme is susceptible to impersonation attack, malicious user attack, offline password guessing attack and man-in-the-middle attack. This paper presents a secure dynamic identity based authentication scheme using smart cards to resolve the aforementioned problems, while keeping the merits of different dynamic identity based authentication schemes.

Categories and Subject Descriptors
K.6.5 [**Computing Milieux**]: Management of Computing and Information System- Security and Protection, Authentication.

General Terms
Security, Algorithms, Verification, Reliability, Human Factors.

Keywords
Network Security; Cryptography; Password; Authentication Protocol; Smart Card; Hash Function.

1. INTRODUCTION
Most of password authentication schemes use smart cards to support users for their authentication. Smart card is used for storing some sensitive information and performing different operations securely. Authorization of reading and writing the data to smart cards is privileged to the server, the card issuer authority and the card reader machine. The user (card holder) submits his identity and password to his smart card. Then smart card performs some operations using submitted arguments and the data stored inside its memory to authenticate the user. Smart cards have been extensively used in many e-commerce applications and network security protocols due to their low cost, portability, efficiency and the cryptographic properties.

In 1981, Lamport [1] proposed a password based authentication scheme that authenticates 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 static and dynamic identity based remote user authentication schemes have been proposed to improve security, efficiency and cost. The static identity leaks out partial information about the user's authentication messages to the attacker. On the other hand, the dynamic identity based authentication schemes provide two-factor authentication based on the identity and password and hence more suitable to e-commerce applications. Therefore in 2004, Das et al. [2] proposed a dynamic identity based remote user authentication scheme to authenticate users that preserves the user's anonymity. Their scheme uses dynamic identity to achieve this purpose and the user's identity is dynamically changed during each new authentication process. The server does not require to keep any verification table and the users can choose and change their passwords without the server's help. Das et al. claimed that their scheme is secure against stolen verifier attack, replay attack, forgery attack, guessing attack, insider attack and identity theft. However, many researchers [3-8] demonstrated susceptibility of Das et al.'s scheme to different attacks. In 2005, Chien and Chen [3] pointed out that Das et al.'s scheme fails to preserve the user anonymity effectively because the authentication messages belonging to the same user can be identified. They proposed an authentication scheme and claimed that the proposed scheme preserves the user's anonymity more efficiently. Though their scheme preserves the user's anonymity and secure against various attacks but it is highly computation intensive. In 2005, Liao et al. [4] proposed an improved scheme that enhances the security of Das et al.'s scheme and achieves mutual authentication. In 2006, Yoon and Yoo [5] demonstrated a reflection attack on Liao et al.'s scheme that breaks the mutual authentication. They also proposed an improved dynamic identity based mutual authentication scheme that eliminates the security flaws of Liao et al.'s scheme.

In 2006, Liou et al. [6] suggested a new dynamic identity based remote user authentication scheme using smart cards that achieves mutual authentication. They claimed that their scheme preserves the advantages of Das et al.'s scheme and overcomes the weaknesses of Das et al.'s scheme. In 2008, Shih [7] demonstrated that Liou et al.'s scheme fails to achieve mutual authentication. In this paper, we found that the Liou et al.'s scheme is susceptible to Ku and Chang's impersonation attack [8], malicious user attack, offline password guessing attack and man-in-the-middle attack. To remedy these pitfalls, this paper presents an efficient scheme that inherits the merits of different

dynamic identity based authentication schemes and resolves the aforementioned problems.

The rest of this paper is organized as follows. In Section 2, a brief review of Liou et al.'s scheme [6] is given. Section 3 describes the susceptibility of Liou et al.'s scheme to different attacks. In Section 4, a new dynamic identity based authentication scheme is proposed. The security analysis of the proposed 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.

2. REVIEW OF LIOU ET AL.'S SCHEME

In this section, we examine the remote user authentication scheme proposed by Liou et al. in 2006. Liou et al.'s scheme consists of four phases viz. registration phase, login phase, verification phase and password change phase as summarized in Figure 1. The notations used in this section are listed in Table 1.

Table 1. Notations

U_i	i^{th} User
S	Server
P_i	Password of User U_i
x	Master Secret of Registration Server S
y	Remote Server's Secret Number
H ()	One-Way Hash Function
⊕	XOR Operation
|	Concatenation

2.1 Registration Phase

A user U_i has to submit his password P_i to the server S for registration over a secure communication channel. The server S computes $M_i = H(P_i) \oplus H(y)$, $N_i = H(P_i) \oplus H(x)$, where x is secret key of the remote server S. Then the server S issues the smart card with secret parameters (H ( ), M_i, N_i, y) to the user U_i through a secure communication channel, where y is the remote server's secret number stored in each registered user's smart card.

2.2 Login Phase

The user U_i inserts his smart card into a card reader to login on to the server S and then submit his password P_i^*. The smart card computes $CID_i = H(P_i^*) \oplus H(M_i \oplus N_i \oplus T)$ and $E_i = M_i \oplus H(T \oplus y)$, where T is current date and time of the input device and sends the login request message (CID_i, E_i, T) to the service provider server S.

2.3 Verification Phase

The service provider server S checks the validity of timestamp T by checking (T' – T) <= δT, where T' denotes the server's current timestamp and δT is expected time interval for a transmission delay. Afterwards, the server S computes $M_i^* = E_i \oplus H(T \oplus y)$, $N_i^* = M_i^* \oplus H(y) \oplus H(x)$, $H(P_i^*) = CID_i \oplus H(M_i^* \oplus N_i^* \oplus T)$, $H(x^*) = N_i^* \oplus H(P_i^*)$ and compares the value of $H(x^*)$ with the known value of H (x). If they are not equal, the server S rejects the login request and terminates this session. Otherwise, the server S computes $R_i = H(M_i^* \oplus N_i^* \oplus T'')$, where T'' denotes the server's current timestamp and sends the message (R_i, T'') back to the smart card of user U_i. On receiving the message (R_i, T''), smart card checks the validity of timestamp T'' by checking (T''' – T'') <= δT, where T''' denotes the client's smart card current timestamp. Then the client's smart card computes $R_i^* = H(M_i \oplus N_i \oplus T'')$ and compares it with the received value of R_i. This equivalency authenticates the legality of the service provider server S and the login request is accepted else the connection is interrupted.

2.4 Password Change Phase

The client C can change his password without the server's help. The user U_i inserts his smart card into a card reader and submits his password P_i^* corresponding to his smart card. Smart card computes $H(P_i^*)$ and extracts M_i, y from its memory to compute $H(P_i) = M_i \oplus H(y)$. Then smart card compares the value of $H(P_i^*)$ with $H(P_i)$ to verifies the legality of the user. If both values match, the legality of card holder is verified and then the client can instruct the smart card to change his password. Afterwards, the smart card asks the card holder to submit a new password P_i^{new}. Then the smart card computes the values $H(x) = H(P_i) \oplus N_i$, $M_i^{new} = H(P_i^{new}) \oplus H(y)$ and $N_i^{new} = H(P_i^{new}) \oplus H(x)$. Finally, the smart card updates the values of M_i and N_i stored in its memory with M_i^{new} and N_i^{new}.

3. WEAKNESSES OF LIOU ET AL.'S SCHEME

Liou et al. [6] claimed that their protocol can resist various known attacks. Unfortunately, this protocol is found to be flawed for impersonation attack, malicious user attack, offline password guessing attack and man-in-the-middle attack.

3.1. Impersonation Attack

Ku and Chang's [8] demonstrated impersonation attack on Das et al.'s scheme [2]. This attack is also applicable on Liou et al.'s [6] scheme. An attacker can perform impersonation attack as follows.

1. The attacker intercepts a login request message (CID_i, E_i, T) of the user U_i from the public communication channel.
2. Now the attacker gets the current time stamp T' and computes $\delta T = T \oplus T'$, $E_i' = E_i \oplus \delta T$ and $CID_i' = CID_i \oplus \delta T$.
3. Then an attacker frames the message (CID_i', E_i', T') and sends this login request message to the server S.
4. The server S checks the validity of the timestamp T' by checking (T'' – T') <= δT, where T'' denotes the server's current timestamp. Then the server S computes:

$$\begin{aligned} M_i' &= E_i' \oplus H(T' \oplus y) \\ &= E_i \oplus \delta T \oplus H(T' \oplus y) \\ &= M_i \oplus \delta T \\ N_i' &= M_i' \oplus H(y) \oplus H(x) \\ &= M_i \oplus \delta T \oplus H(y) \oplus H(x) \\ &= N_i \oplus \delta T \\ H(P_i') &= CID_i' \oplus H(M_i' \oplus N_i' \oplus T') \\ &= CID_i \oplus \delta T \oplus H(M_i \oplus \delta T \oplus N_i \oplus \delta T \oplus T') \\ &= CID_i \oplus \delta T \oplus H(M_i \oplus N_i \oplus T') \\ &= H(P_i) \oplus \delta T \\ H(x) &= N_i' \oplus H(P_i') \\ &= N_i \oplus \delta T \oplus H(P_i) \oplus \delta T \\ &= N_i \oplus H(P_i) \end{aligned}$$

The server S compares this computed value of H (x) with the known value of H (x). On this successful verification, the server S accepts the forged login authentication request. Therefore, the attacker can impersonate as the legitimate user U_i.

3.2. Malicious User Attack

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. [9] and Messerges et al. [10]. Therefore, a malicious privileged user U_i can extract $M_i = H(P_i) \oplus H(y)$, $N_i = H(P_i) \oplus H(x)$ and y from his own smart card. He can find out $H(x) = N_i \oplus H(P_i)$ because the malicious user U_i knows his own password P_i corresponding to his smart card.

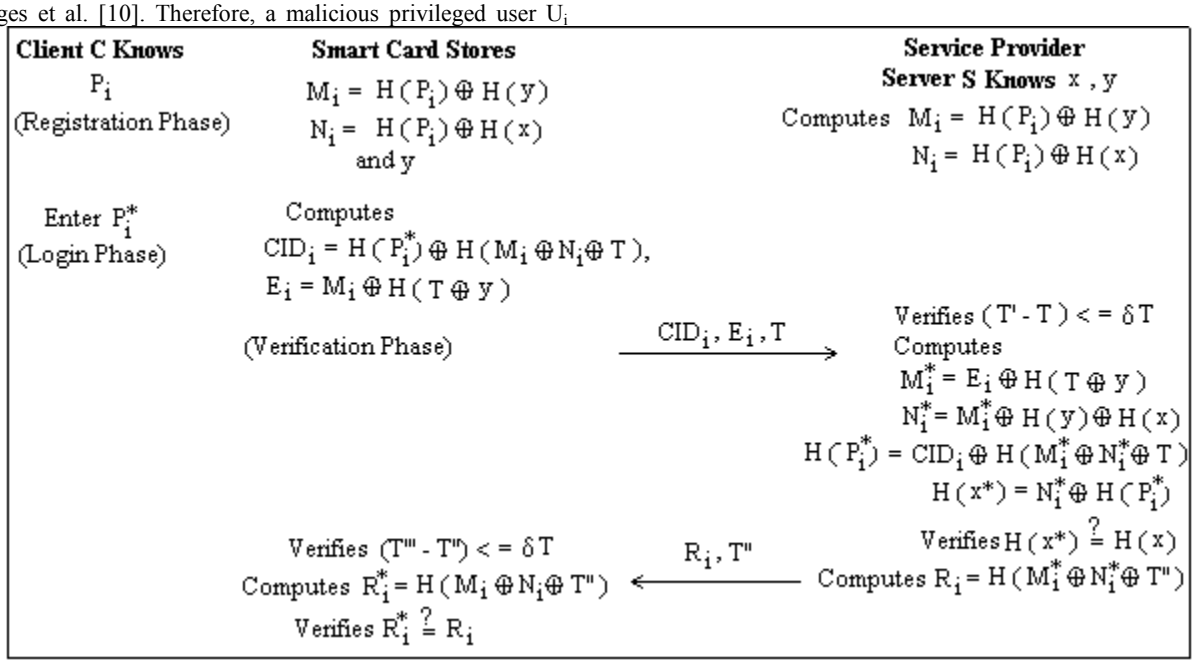

Figure 1. Liou et al.'s Scheme

1. Now this malicious privileged user U_i intercepts the login request message (CID_K, E_K, T) of the user U_K from the public communication channel.
2. This malicious user U_i can compute the password verifier information of the user U_K as $H(P_K) = CID_K \oplus H(H(y) \oplus H(x) \oplus T)$ because the malicious user knows y and H (x).
   Since $H(M_K \oplus N_K \oplus T) = H(H(P_K) \oplus H(y) \oplus H(P_K) \oplus H(x) \oplus T)$
   $= H(H(y) \oplus H(x) \oplus T)$
   Then the malicious user can compute the values of $M_K = H(P_K) \oplus H(y)$, $N_K = H(P_K) \oplus H(x)$ and hence can frame fabricated login request message (CID_K', E_K', T') corresponding to the user U_K, where $CID_K' = H(P_K) \oplus H(M_K \oplus N_K \oplus T')$ and $E_K' = M_K \oplus H(T' \oplus y)$. Afterwards, the malicious user U_i sends this fabricated login request message to the server S.
3. The service provider server S checks the validity of timestamp T' by checking (T'' – T') <= δT, where T'' denotes the server's current timestamp and δT is expected time interval for a transmission delay. Afterwards, the server S computes:
   $M_K = E_K' \oplus H(T' \oplus y)$
   $N_K = M_K \oplus H(y) \oplus H(x)$
   $H(P_K) = CID_K' \oplus H(M_K \oplus N_K \oplus T')$
   $H(x) = N_K \oplus H(P_K)$

Then the server S compares this computed value of H (x) with the known value of H (x). This equivalency authenticates the legality of the user U_K and the login request is accepted by the service provider server S.

3.3. Offline Password Guessing 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 such as by monitoring their power consumption and reverse engineering techniques as pointed out by Kocher et al. [9] and Messerges et al. [10]. He can extract $M_i = H(P_i) \oplus H(y)$, $N_i = H(P_i) \oplus H(x)$ and y from the memory of smart card because smart card contains (M_i, N_i, y, H ( )). Then the attacker can find out the password information $H(P_i)$ of user U_i as $H(P_i) = M_i \oplus H(y)$. Now the attacker can guess different values of P_i and check its correctness by verifying it with the actual value of $H(P_i)$.

3.4. Man-in-the-middle Attack

In this type of attack, an attacker intercepts the messages sent between the client and the server and replay these intercepted messages with in the valid time frame window. An attacker can act as a client to the server or vice-versa with recorded messages.

1. The malicious privileged user U_i intercepts the login request message (CID_K, E_K, T) of the user U_K to the server S from the public communication channel.
2. Then this malicious privileged user U_i starts a new session with the server S by sending a login request message (CID_i, E_i, T').
3. After receiving the login request, the server S check the validity of timestamp T' by checking (T'' – T') <= δT, where T'' denotes the server's current timestamp. Then the server S computes:
   $M_i = E_i \oplus H(T \oplus y)$

$N_i = M_i \oplus H(y) \oplus H(x)$
$H(P_i) = CID_i \oplus H(M_i \oplus N_i \oplus T)$
$H(x) = N_i \oplus H(P_i)$

Then the server S compares this computed value of H (x) with the known value of H (x). This equivalency authenticates the legality of the user U_i and the login request is accepted by the service provider server S.

4. The server S computes $R_i = H(M_i \oplus N_i \oplus T'')$ and sends the message (R_i, T'') back to the user U_i.
5. Now the user U_i immediately sends the message (R_i, T'') to the user U_K.
6. The user U_K checks the validity of timestamp T'' by checking (T''' – T'') <= δT, where T''' denotes the user U_K's smart card current timestamp. The user U_K computes $R_K = H(M_K \oplus N_K \oplus T'')$ and compares it with the received value of R_i.

$R_K = H(M_K \oplus N_K \oplus T'')$
$= H(H(P_K) \oplus H(y) \oplus H(P_K) \oplus H(x) \oplus T'')$
$= H(H(y) \oplus H(x) \oplus T'')$
$R_i = H(M_i \oplus N_i \oplus T'')$
$= H(H(P_i) \oplus H(y) \oplus H(P_i) \oplus H(x) \oplus T'')$
$= H(H(y) \oplus H(x) \oplus T'')$

This equivalency authenticates the legality of the service provider server S and the login request is accepted by the user U_K. Thus the user U_i acts as middle-man between the user U_K and the server S and masquerade as the legitimate server S to the user U_K.

4. PROPOSED PROTOCOL

In this section, we describe a new remote user authentication scheme which resolves the above security flaws of Liou et al.'s [6] scheme. Figure 2 shows the entire protocol structure of the new authentication scheme. Legitimate client C can easily login on to the service provider server using his smart card, identity and password. Notations used in this section are listed in Table 2. The proposed scheme consists of four phases viz. registration phase, login phase, authentication phase and password change phase.

Table 2. Notations

U_i	i^{th} User
S	Server
ID_i	Unique Identification of U_i
P_i	Password of User U_i
x	Master Secret of Registration Server S
y_i	Random Number Selected by Server S
H ()	One-Way Hash Function
⊕	XOR Operation
|	Concatenation

1. Registration Phase
When a user U_i wants to become a legal client C, the user has to submit his identity and password to the server S via a secure communication channel. Then, the server S computes some security parameters and stores them on the smart card of the client. Then, the server S issues the smart card to the client C.

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. Smart card verifies authenticity of the client and sends client verifier information to the destination server S.

3. Authentication Phase
The service provider server S verifies the authenticity of the client and then the server S sends server verifier information to the smart card of the client to authenticate itself. Once the client and the server mutually authenticate each other then they agree on the common session key.

4. Password Change Phase
The client has to authenticate itself to the smart card before requesting the password change.

4.1. Registration Phase

A user U_i has to submit his unique identity ID_i and password P_i to the server S for registration over a secure communication channel.

Step 1: C → S: ID_i, P_i

The server S computes the security parameters $A_i = H(x \mid y_i)$, $B_i = H(ID_i \mid P_i) \oplus P_i \oplus H(x \mid y_i)$, $C_i = H(x \mid y_i) \oplus H(P_i)$ and $D_i = H(ID_i \mid P_i) \oplus H(x)$. The server S chooses the value of y_i corresponding to each user in such a way that the value of A_i must be unique for each user. The server S stores $y_i \oplus x$ and $ID_i \oplus H(x)$ corresponding to A_i in its database. Then the server S issues the smart card containing security parameters (B_i, C_i, D_i, H ( )) to the user U_i through a secure communication channel.

Step 2: S → C: 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 $H(x \mid y_i) = B_i \oplus H(ID_i^* \mid P_i^*) \oplus P_i^*$, $C_i^* = H(x \mid y_i) \oplus H(P_i^*)$ and compares C_i^* with the stored value of C_i in its memory to verifies the legality of the user.

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

After verification, the smart card computes $H(x) = D_i \oplus H(ID_i \mid P_i)$, $CID_i = H(x \mid y_i) \oplus H(H(x) \mid T)$ and $M_i = H(H(x) \mid H(x \mid y_i) \mid T)$, where T is current date and time of the input device. Then the smart card sends login request message (CID_i, M_i, T) to the service provider server S.

Step 2: Smart card → S: CID_i, M_i, T

4.3. Authentication Phase

After receiving the login request message from the client C, the service provider server S checks the validity of timestamp T by checking (T' –T) <= δT, where T' is current date and time of the server S and δT is expected time interval for a transmission delay. The server S computes $A_i^* = CID_i \oplus H(H(x) \mid T)$ and finds A_i corresponding to A_i^* in its database and then extracts $y_i \oplus x$ and $ID_i \oplus H(x)$ corresponding to A_i from its database. Now the server S computes y_i from $y_i \oplus x$ and ID_i from $ID_i \oplus H(x)$ because the server S knows the value of x. Then the server S computes $M_i^* = H(H(x) \mid A_i \mid T)$ and compares M_i^* with the received value of M_i.

Step 1: Server S checks $M_i^* ?= M_i$

If they are not equal, the server S rejects the login request and terminates this session. Otherwise, the server S acquires the current time stamp T'' and computes $V_i = H(A_i \mid H(x) \mid T \mid T'')$ and sends the message (V_i, T'') back to the smart card of the user U_i.

Step 2: S → Smart card: V_i, T''

On receiving the message (V_i, T''), the user U_i's smart card checks the validity of timestamp T'' by checking (T''' –T'') <= δT, where T''' is current date and time of the smart card. Then the smart card computes $V_i^* = H(H(x \mid y_i) \mid H(x) \mid T \mid T'')$ and compares it with the received value of V_i.

Step 3: Smart card checks $V_i^* ?= V_i$

This equivalency authenticates the legality of the service provider server S and the login request is accepted else the connection is interrupted. Finally, the client C and the server S agree on the common session key as $H(ID_i \mid H(x \mid y_i) \mid H(x) \mid T \mid T'')$.

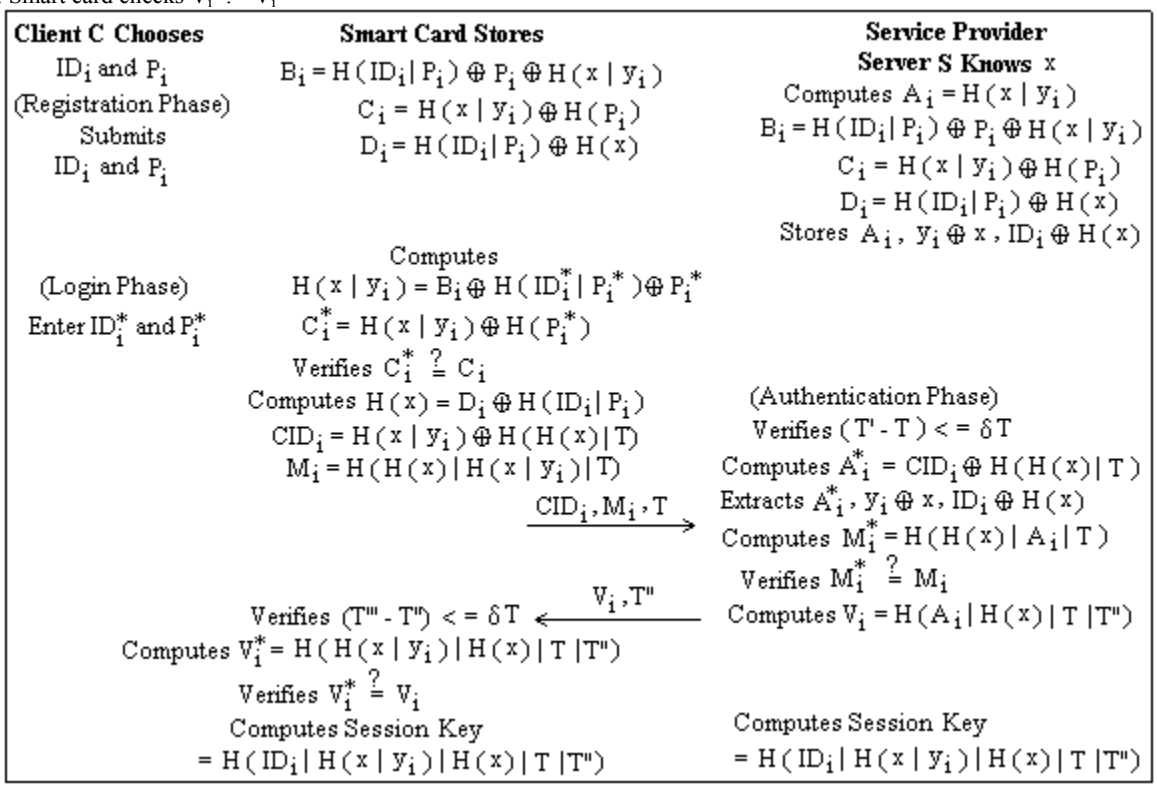

Figure 2. Proposed Protocol

4.4. Password Change Phase

The client C can change his password without the server's help. The user U_i inserts his smart card into a card reader and enters his identity ID_i^* and password P_i^* corresponding to his smart card. The smart card computes $H(x \mid y_i) = B_i \oplus H(ID_i^* \mid P_i^*) \oplus P_i^*$, $C_i^* = H(x \mid y_i) \oplus H(P_i^*)$ and compares C_i^* with the stored value of C_i in its memory to verifies the legality of the user. Once the legality of card holder is verified then the client can instruct the smart card to change his password. Afterwards, the smart card asks the card holder to resubmit a new password P_i^{new} and then smart card computes $B_i^{new} = H(ID_i \mid P_i^{new}) \oplus P_i^{new} \oplus H(x \mid y_i)$, $C_i^{new} = H(x \mid y_i) \oplus H(P_i^{new})$ and $D_i^{new} = D_i \oplus H(ID_i \mid P_i) \oplus H(ID_i \mid P_i^{new})$. Thereafter, smart card updates the values of B_i, C_i and D_i stored in its memory with B_i^{new}, C_i^{new} and D_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. [9] and Messerges et al. [10] pointed out that all existing smart cards can not prevent the information stored in them from being extracted by techniques such as monitoring their power consumption. Some other reverse engineering techniques are also available for extracting information from 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 impersonation attack, malicious user attack, stolen smart card attack, password guessing and other feasible attacks.

1. **Impersonation Attack:** In this type of attack, an attacker impersonates as a legitimate client by forging the authentication messages using the information obtained from the authentication scheme. An attacker can attempt to modify a login request message (CID_i, M_i, T) into (CID_i^*, M_i^*, T^*), where T^* is the attacker's current date and time, so as to succeed in the authentication phase. However, such a modification will fail in Step 1 of the authentication phase because the attacker require to know the values A_i and $H(x)$ to compute the valid parameters CID_i^* and M_i^*. Moreover, the attacker should know ID_i to compute the session key. Therefore, the proposed protocol is secure against impersonation attack.
2. **Malicious User Attack:** A malicious privileged user having his own smart card can gather information like B_i =

$H(ID_i \mid P_i) \oplus P_i \oplus H(x \mid y_i)$, $C_i = H(x \mid y_i) \oplus H(P_i)$ and $D_i = H(ID_i \mid P_i) \oplus H(x)$ from the memory of smart card. This malicious user can not generate smart card specific value of $CID_K = H(x \mid y_K) \oplus H(H(x) \mid T)$ and $M_K = H(H(x) \mid H(x \mid y_K) \mid T)$ to masquerade as other legitimate user U_K to the service provider server S because the values of CID_K and M_K is smart card specific and depend upon the values of x and y_K. Although, the malicious user can extract H (x) from his own smart card but he does not have any method to calculate the value of x and y_K. Moreover, the malicious user should know ID_K to compute the session key. Therefore, the proposed protocol is secure against malicious user attack.

3. **Stolen Smart Card Attack:** In case a user's smart card is stolen by an attacker, he can extract the information stored in the smart card. An attacker extracts $B_i = H(ID_i \mid P_i) \oplus P_i \oplus H(x \mid y_i)$, $C_i = H(x \mid y_i) \oplus H(P_i)$ and $D_i = H(ID_i \mid P_i) \oplus H(x)$ from the memory of smart card. Even after gathering this information, an attacker has to guess ID_i and P_i correctly at the same time. It is not possible to guess two parameters correctly at the same time in real polynomial time. Therefore, the proposed protocol is secure against stolen smart card attack.
4. **Offline Dictionary Attack:** In offline dictionary attack, an attacker can record messages and attempts to guess the user's identity ID_i and password P_i from the recorded messages. An attacker first tries to obtains some client or server verification information such as $CID_i = H(x \mid y_i) \oplus H(H(x) \mid T)$, $M_i = H(H(x) \mid H(x \mid y_i) \mid T)$, $V_i = H(A_i \mid H(x) \mid T \mid T'')$ and then tries to guess x and y_i by offline dictionary attack. Even after gathering this information, the attacker has to guess both these parameters correctly at the same time. It is not possible to guess both parameters correctly at the same time. Therefore, the proposed protocol is secure against offline dictionary attack.
5. **Man-in-the-middle Attack:** In this type of attack, an attacker intercepts the messages sent between the client and the server and replay these intercepted messages with in the valid time frame window. An attacker can act as a client to the server or vice-versa with recorded messages. In our proposed protocol, an attacker can intercept the login request message (CID_i, M_i, T) from the client to the server S, which is sent by a valid user U_i to S. Then he starts a new session with the server S by replaying the login request message (CID_i, M_i, T) with in the valid time frame window. An attacker can authenticate itself to the server S as well as to the legitimate client but can not compute the session key $H(ID_i \mid H(x \mid y_i) \mid H(x) \mid T \mid T'')$ because the attacker does not know the value of ID_i, x and y_i. Therefore, the proposed protocol is secure against man-in-the-middle attack.
6. **Denial of Service Attack:** In denial of service attack, an attacker updates password verification information on the smart card to some arbitrary value and hence legal user can not login successfully in subsequent login request to the server. In our proposed protocol, smart card checks the validity of user identity ID_i and password P_i before password update procedure. An 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 $H(x \mid y_i) = B_i \oplus H(ID_i^* \mid P_i^*) \oplus P_i^*$, $C_i^* = H(x \mid y_i) \oplus H(P_i^*)$ and compares C_i^* with the stored value of C_i in its memory to verifies the legality of the user before the smart card accepts the password update request. It is not possible to guess identity ID_i and password P_i correctly at the same time even after getting the smart card of the legitimate user. Therefore, the proposed protocol is secure against denial of service attack.
7. **Replay Attack:** In this type of attack, an attacker first listens to communication between the client and the server and then tries to imitate user to login on to the server by resending the captured messages transmitted between the client and the server. Replaying a message of one session into another session is useless because the client's smart card and the server S uses current time stamp values as T and T'' in each new session, which make all the messages CID_i, M_i and V_i dynamic and valid for small interval of time. Old replayed messages are not valid in current session and hence proposed protocol is secure against message replay attack.
8. **Leak of Verifier Attack:** In this type of attack, the attacker may able to steal verification table from the server. If an attacker steals the verification table from the server, he can use the stolen verifiers to impersonate a participant of the scheme. In our proposed scheme, the service provider server S knows the secret x and stores $y_i \oplus x$ and $ID_i \oplus H(x)$ corresponding to user's $A_i = H(x \mid y_i)$ value in its database. An attacker does not have any technique to find out the value of x and hence can not calculate y_i from $y_i \oplus x$ and ID_i from $ID_i \oplus H(x)$. Therefore, the proposed protocol is secure against leak of verifier attack.
9. **Server Spoofing Attack:** In server spoofing attack, an attacker can manipulate the sensitive data of legitimate users via setting up fake servers. Malicious server can not generate the valid value of $V_i = H(A_i \mid H(x) \mid T \mid T'')$ meant for the smart card because the malicious server has to know the values of x and y_i to generate the valid value of V_i corresponding to that client's smart card. Moreover, the malicious server should know ID_i to compute the session key. The proposed scheme provides mutual authentication to withstand the server spoofing attack. Therefore, the proposed protocol is secure against server spoofing attack.
10. **Online Dictionary Attack:** In this type of attack, an attacker pretends to be the legitimate client and attempts to login on to the server by guessing different words as password from a dictionary. In our scheme, an attacker has to get the valid smart card and then has to guess the identity ID_i and password P_i corresponding to that client. Even after getting the valid smart card by any mean, an attacker gets a very few chances to guess the identity ID_i and password P_i because smart card gets locked after certain number of unsuccessful attempts. Moreover, it is not possible to guess identity ID_i and password P_i correctly at the same time in real polynomial time. Therefore, the proposed protocol is secure against online dictionary attack.
11. **Parallel Session Attack:** In this type of attack, an attacker first listens to communication between the client 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 legitimate user U_i by replaying a login

request message (CID_i, M_i, T) with in the valid time frame window. However, an attacker can not compute the agreed session key H (ID_i | H (x | y_i) | H (x) | T | T'') because the attacker does not know the values of ID_i, x and y_i. Therefore, the proposed protocol is secure against parallel session attack.

12. **Message Modification or Insertion Attack:** In this type of attack, an attacker modifies or inserts some messages on the communication channel with the hope of discovering the client's password or gaining unauthorized access. Modifying or inserting messages in the proposed protocol can only cause authentication between the client and the server to fail but can not allow the attacker to gain any information about the client's identity ID_i and password P_i or gain unauthorized access. Therefore, the proposed protocol is secure against message modification or insertion attack.

Table 3. Cost Comparison among Related Smart Card based Authentication Schemes

	Our Scheme	Liou et al. [6]	Yoon-Yoo [5]	Liao et al. [4]	Chien-Chen [3]	Das et al. [2]
E1	384 bits	384 bits	384 bits	256 bits	384 bits	256 bits
E2	5 * 128 bits	5 *128 bits	6 *128 bits	6 *128 bits	4 *128 bits	4 *128 bits
E3	$4T_H$	$3T_H$	$3T_H$	$2T_H$	$2T_H$	$2T_H$
E4	$6T_H$	$4T_H$	$6T_H$	$5T_H$	$2T_E + 2T_S$	$4T_H$
E5	$5T_H$	$5T_H$	$4T_H$	$4T_H$	$2T_H + 2T_E + 2T_S$	$3T_H$

Table 4. Functionality Comparison among Related Smart Card based Authentication Schemes

	Our Scheme	Liou et al. [6]	Yoon-Yoo [5]	Liao et al. [4]	Chien-Chen [3]	Das et al. [2]
User's Anonymity	Yes	Yes	Yes	Yes	Yes	Yes
Session Key Agreement	Yes	No	No	No	Yes	No
Impersonation Attack	No	Yes	No	Yes	No	Yes
Malicious User Attack	No	Yes	No	Yes	No	Yes
Offline Dictionary Attack	No	Yes	Yes	Yes	No	Yes
Man-in-the-middle Attack	No	Yes	No	Yes	No	No
Mutual Authentication	Yes	Yes	Yes	Yes	Yes	No

6. COST AND FUNCTIONALITY ANALYSIS

An efficient authentication scheme must take communication and computation cost into consideration during user's authentication. The performance comparison of the proposed scheme with the relevant smart card based authentication schemes is summarized in Table 3. Assume that the identity ID_i, password P_i, x and y_i values are all 128-bit long. Moreover, we assume that the output of secure one-way hash function is 128-bit. Let T_H, T_E and T_S denote the time complexity for hash function, exponential operation and symmetric key encryption respectively. Typically, time complexity associated with these operations can be roughly expressed as $T_S >> T_E >> T_H$. In the proposed scheme, the parameters stored in the smart card are B_i, C_i, D_i and the memory needed in the smart card (E1) is 384 (= 3*128) bits. The communication cost of authentication (E2) includes the capacity of transmitting message involved in the authentication scheme. The capacity of transmitting message (CID_i, M_i, T) and (V_i, T'') is 640 (= 5*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 is $4T_H$. 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, both the computation cost of the user and that of the service provider server are $6T_H$ and $5T_H$ respectively. The functionality comparison of the proposed scheme with the relevant smart card based authentication schemes is summarized in Table 4. The proposed scheme requires nearly the same computation as other related schemes [2][4][5][6] and requires very less computation as compared to Chien and Chen scheme [3] but it is highly secure as compared to the related schemes.

7. CONCLUSION

Corporate network and e-commerce applications require secure and practical remote user authentication solutions. Smart card based password authentication is one of the most convenient ways to provide authentication for the communication between a client and a server because it provides inherent confidentiality, portable size and intelligent computing capability. In this paper, we presented a cryptanalysis of Liou et al.'s scheme and showed that their scheme is vulnerable to impersonation attack, malicious user attack, offline password guessing attack and man-in-the-middle attack. A secure dynamic identity based authentication scheme using smart cards is proposed to resolve the aforementioned problems, while keeping the merits of different dynamic identity based authentication schemes. The proposed protocol is simplified, fast and efficient because only one-way hash functions and XOR operations are used in its implementation. Security analysis proved that the improved scheme is more secure and practical.

8. REFERENCES

[1] L. Lamport. Password Authentication with Insecure Communication. *Communications of the ACM*, vol. 24, no. 11, pages 770-772. November 1981.

[2] M.L. Das, A. Saxena and V.P. Gulati. A Dynamic ID-Based Remote User Authentication Scheme. *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pages 629-631. May 2004.

[3] H.Y. Chien and C.H. Chen. A Remote Authentication Scheme Preserving User Anonymity. *In Proceedings of Advanced Information Networking and Applications*, vol. 2, pages 245-248. March 2005.

[4] I.E. Liao, C.C. Lee and M.S. Hwang. Security Enhancement for a Dynamic ID-Based Remote User Authentication Scheme. *In Proceeding* of *Conference on Next Generation Web Services Practice*, pages 437-440. July 2005.

[5] E.J. Yoon and K.Y. Yoo. Improving the Dynamic ID-Based Remote Mutual Authentication Scheme. *In Proceedings of OTM Workshops 2006, LNCS 4277*, pages 499-507. July 2006.

[6] Y.P. Liou, J. Lin and S.S. Wang. A New Dynamic ID-Based Remote User Authentication Scheme using Smart Cards. *In Proceedings of 16th Information Security Conference*, Taiwan, pages 198-205. July 2006.

[7] H.C. Shih. Cryptanalysis on Two Password Authentication Schemes. *Laboratory of Cryptography and Information Security, National Central University*, Taiwan, July 2008.

[8] W.C. Ku and S.T. Chang. Impersonation Attack on a Dynamic ID-based Remote User Authentication Scheme using Smart Cards. *IEICE Transactions on Communications*, vol. E88-B, no. 5, pages 2165-2167. May 2005.

[9] P. Kocher, J. Jaffe and B. Jun. Differential Power Analysis. *Proc. CRYPTO 99, Springer-Verlag*, pages 388-397. August 1999.

[10] 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, pages 541-552. May 2002.