Journal of Information Security, **2012, 3, 326-334**
http://dx.doi.org/10.4236/jis.2012.34040 Published Online October 2012 (http://www.SciRP.org/journal/jis)

Dynamic Identity Based Authentication Protocol for Two-Server Architecture

Sandeep K. Sood
Department of Computer Science & Engineering, Regional Campus Gurdaspur, Gurdaspur, India
Email: san1198@gmail.com

Received July 17, 2012; revised August 23, 2012; accepted September 3, 2012

ABSTRACT

Most of the password based authentication protocols make use of the single authentication server for user's authentication. User's verifier information stored on the single server is a main point of susceptibility and remains an attractive target for the attacker. On the other hand, multi-server architecture based authentication protocols make it difficult for the attacker to find out any significant authentication information related to the legitimate users. In 2009, Liao and Wang proposed a dynamic identity based remote user authentication protocol for multi-server environment. However, we found that Liao and Wang's protocol is susceptible to malicious server attack and malicious user attack. This paper presents a novel dynamic identity based authentication protocol for multi-server architecture using smart cards that resolves the aforementioned flaws, while keeping the merits of Liao and Wang's protocol. It uses two-server paradigm by imposing different levels of trust upon the two servers and the user's verifier information is distributed between these two servers known as the service provider server and the control server. The proposed protocol is practical and computational efficient because only nonce, one-way hash function and XOR operations are used in its implementation. It provides a secure method to change the user's password without the server's help. In e-commerce, the number of servers providing the services to the user is usually more than one and hence secure authentication protocols for multi-server environment are required.

Keywords: Authentication Protocol; Smart Card; Dynamic Identity; Multi-Server Architecture; Password

1. Introduction

Most of the existing password authentication protocols are based on single-server model in which the server stores the user's password verifier information in its database. Password verifier information stored on the single server is mainly susceptible to stolen verifier attack. The concept of multi-server model removes this common point of susceptibility. The proposed protocol uses multi-server model consisting of two servers at the server side that work together to authenticate the users. Different levels of trust are assigned to the servers and the service provider server is more exposed to the clients than that of the control server. The back-end control server is not directly accessible to the clients and thus it is less likely to be attacked. Two-server model provides the flexibility to distribute user passwords and the authentication functionality into two servers to eliminate the main point of vulnerability of the single-server model. Therefore, two-server model appears to be a genuine choice for practical applications.

In a single server environment, the issue of remote login authentication with smart cards has already been solved by a variety of schemes. These conventional single-server password authentication protocols can not be directly applied to multi-server environment because each user needs to remember different sets of identities and passwords. Different protocols have been suggested to access the resources of multi-server environment. A secure and efficient remote user authentication protocol for multi-server environment should provide mutual authentication, key agreement, secure password update, low computation requirements and resistance to different feasible attacks.

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 for multi-server environment 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 two-factor authentication based on the identity and password 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 for multi-server environment using smart cards. It protects the user's identity in insecure communication channel and hence can be applied directly to e-economic applications.

This paper is organized as follows. In Section 2, we explore the literature on existing authentication protocols for multi-server environment. Section 3 reviews the dynamic identity based remote user authentication protocol for multi-server environment proposed by Liao and Wang. Section 4 describes the susceptibility of Liao and Wang's protocol to malicious server attack and malicious user attack. In Section 5, we present dynamic identity based authentication protocol for multi-server architecture using smart cards. Section 6 discusses the security analysis of the proposed protocol. The comparison of the cost and functionality of the proposed protocol with other related protocols is shown in Section 7. Section 8 concludes the paper.

2. Related Work

A number of smart card based remote user authentication protocols have been proposed due to the convenience and secure computation provided by the smart cards. However, most of these protocols do not protect the user's identities in authentication process. User's anonymity is an important issue in many e-commerce applications.

In 2000, Ford and Kaliski [1] proposed the first multi-server password based authentication protocol that splits a password among multiple servers. This protocol generates a strong secret using password based on the communications exchanges with two or more independent servers. The attacker can not compute the strong secret unless all the servers are compromised. This protocol is highly computation intensive due to the use of public keys by the servers. Moreover, the user requires a prior secure authentication channel with the server. Therefore in 2001, Jablon [2] improved this protocol and proposed multi-server password authentication protocol in which the servers do not use public keys and the user does not require prior secure communication channels with the servers.

In 2003, Lin *et al*. [3] proposed a multi-server authentication protocol based on the ElGamal digital signature scheme that uses simple geometric properties of the Euclidean and discrete logarithm problem concept. The server does not require keeping any verification table but the use of public keys makes this protocol computation intensive. In 2004, Juang [4] proposed a smart card based multi-server authentication protocol using symmetric encryption algorithm without maintaining any verification table on the server. In 2004, Chang and Lee [5] improved Juang's protocol and proposed a smart card based multi-server authentication protocol using symmetric encryption algorithm without any verification table. Their protocol is more efficient than the multi-server authentication protocol of Juang [4]. In 2007, Hu *et al*. [6] 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. The client and the server authenticate each other and agree on a common secret session key. The proposed protocol is more efficient and more user friendly than that of Chang and Lee [5] protocol.

In 2006, Yang *et al*. [7] proposed a password based user authentication and key exchange protocol using two-server architecture in which only a front-end server communicates directly with the users and a control server does not interact with the users directly. The concept of distributing the password verification information and authentication functionality into two servers requires additional efforts from an attacker to compromise two servers to launch successful offline dictionary attack. In 2008, Tsai [8] 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. The proposed authentication protocol is efficient as compared to other such related protocols because it does not use any symmetric and asymmetric encryption algorithm for its implementation. In 2009, Liao and Wang [9] proposed a dynamic identity based remote user authentication protocol using smart cards to achieve user's anonymity. This protocol uses only hash function to implement a strong authentication for the multi-server environment. It provides a secure method to update the user's password without the help of trusted third party. In their paper, they claimed that suggested protocol can resist various known attacks. However, we show in Section 4 that their protocol is insecure in the presence of an active attacker. In 2009, Hsiang and Shih [10] also found that Liao and Wang's protocol is susceptible to insider attack, masquerade attack, server spoofing attack, registration center spoofing attack and is not reparable. Furthermore, it fails to provide mutual authentication. To remedy these flaws, Hsiang and Shih proposed an improvement over Liao and Wang's protocol. In 2010, Sood *et al*. [11] found that Hsiang and Shih protocol is also found to be flawed for replay attack, impersonation attack and stolen smart card attack.

3. Review of Liao and Wang's Protocol

In this section, we describe the dynamic identity based remote user authentication protocol for multi-server environment proposed by Liao and Wang [9]. The notations used in this section are listed in **Table 1** and the protocol

is shown in **Figure 1**.

3.1. Registration Phase

The user U_i has to submit his identity ID_i and password P_i to registration center RC so that he can access the resources of the service provider server S_J. The RC computes

$$T_i = H(ID_i|x),\ V_i = T_i \oplus H(ID_i|P_i),\ B_i = H(P_i) \oplus H(x)$$

and $D_i = H(T_i)$. Then RC issues the smart card with secret parameters (V_i, B_i, D_i, H ( ), y) to the user U_i through a secure communication channel.

3.2. Login Phase

The user U_i submits his identity ID_i^*, password P_i^* and the server identity SID_J to smart card in order to login on to the service provider server S_J. The smart card computes $T_i^* = V_i \oplus H(ID_i^* | P_i^*)$, $D_i^* = H(T_i^*)$ and then verifies the equality of calculated value of D_i^* with the stored value of D_i in its memory. If both values of D_i match, the legitimacy of the user is assured and smart card proceeds to the next step. Otherwise the login request from the user U_i is rejected. Then smart card generates nonce value N_i and computes

$$CID_i = H(P_i) \oplus H(T_i|y|N_i),\ P_{iJ} = T_i \oplus H(y|N_i|SID_J)$$

and $Q_i = H(B_i|y|N_i)$. Afterwards, smart card sends the login request message (CID_i, P_{iJ}, Q_i, N_i) to the server S_J.

3.3. Mutual Verification and Session Key Agreement Phase

The server S_J computes

$$T_i = P_{iJ} \oplus H(y|N_i|SID_J),\ H(P_i) = CID_i \oplus H(T_i|y|N_i),$$
$$B_i = H(P_i) \oplus H(x)$$

and $Q_i^* = H(B_i|y|N_i)$, and then compares the computed

Table 1. Notations.

U_i	i^{th} User
S_J	J^{th} Server
RC	Registration Center
ID_i	Unique Identification of User U_i
P_i	Password of User U_i
SID_J	Unique Identification of Server S_J
CID_i	Dynamic Identity of User U_i
H ( )	One-Way Hash Function
x	Master Secret of Registration Center
y	Shared Secret Key of Registration Center & All Servers
$\oplus$	XOR Operation
|	Concatenation

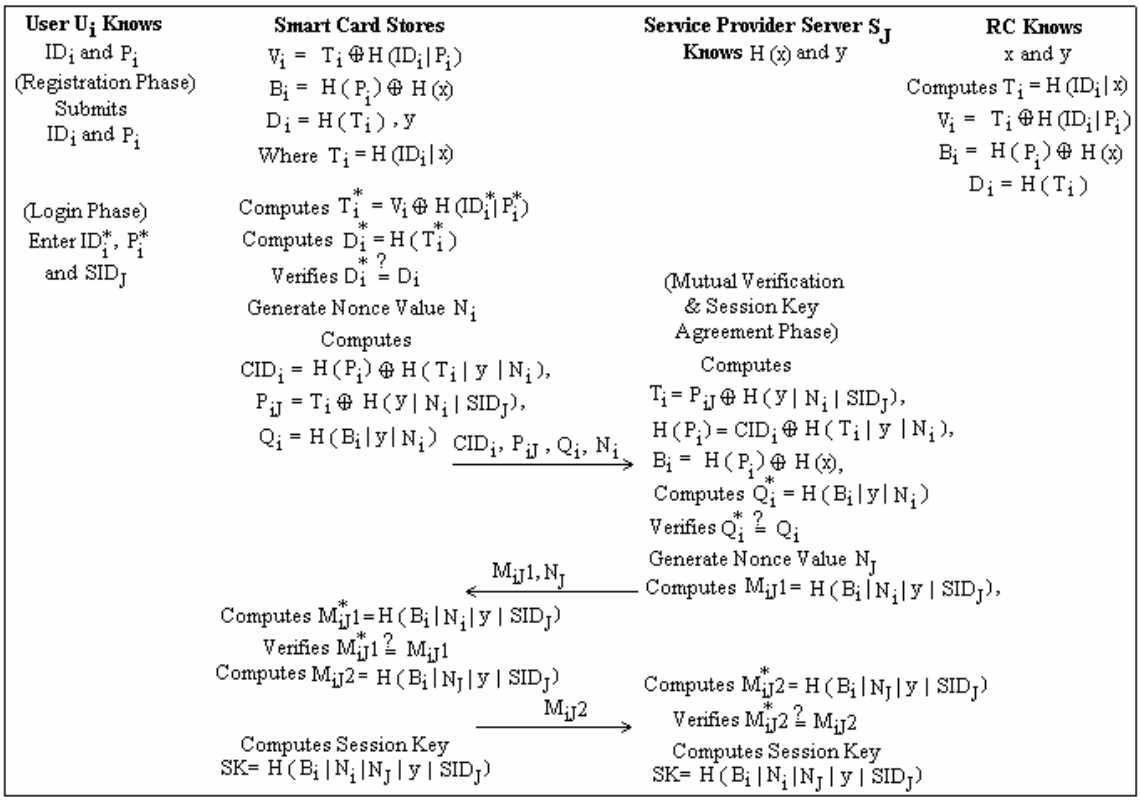

Figure 1. Liao and Wang's dynamic identity based on multi-server authentication protocol.

value of Q_i^* with the received value of Q_i. If they are not equal, the server S_J rejects the login request and terminates this session. Otherwise, the server S_J generates nonce value N_J and computes $M_{iJ}1 = H(B_i|N_i|y|SID_J)$ and sends the message ($M_{iJ}1$, N_J) back to smart card of the user U_i. On receiving the message ($M_{iJ}1$, N_J), the user U_i's smart card computes $M_{iJ}1^* = H(B_i|N_i|y|SID_J)$ and compares the computed value of $M_{iJ}1^*$ with the received value of $M_{iJ}1$. This equivalency authenticates the legitimacy of the service provider server S_J else the connection is interrupted. Then the user U_i's smart card computes $M_{iJ}2 = H(B_i|N_J|y|SID_J)$ and sends $M_{iJ}2$ back to the service provider server S_J. On receiving the message $M_{iJ}2$, the service provider server S_J computes $M_{iJ}2^* = H(B_i|N_J|y|SID_J)$ and compares the computed value of $M_{iJ}2^*$ with the received value of $M_{iJ}2$. This equivalency assures the legitimacy of the user U_i. After finishing mutual authentication, the user U_i and the service provider server S_J computes $SK = H(B_i|N_i|N_J|y|SID_J)$ as the session key.

4. Cryptanalysis of Liao and Wang's Protocol

Liao and Wang [9] claimed that their protocol provides identity privacy and can resist various known attacks. However, we found that this protocol is flawed for malicious server attack and malicious user attack.

4.1. Malicious Server Attack

The malicious legitimate server S_J can compute the value of T_i, $H(P_i)$ and B_i corresponding to the user U_i during mutual verification and session key agreement phase. This malicious server S_J also knows H ( ) function, y and H(x) because Liao and Wang mentioned that y is the shared key among the users, the servers and the registration center and H (x) is used by the legitimate server S_J to compute $B_i = H(P_i) \oplus H(x)$. The malicious server S_J can record $CID_i = H(P_i) \oplus H(T_i|y|N_i)$, $Q_i = H(B_i|y|N_i)$, N_i during login request message from the user U_i and computes $P_{ik} = T_i \oplus H(y|N_i|SID_k)$ corresponding to the user U_i. Afterwards, the malicious server S_J sends the login request message (CID_i, P_{ik}, Q_i, N_i) to the service provider server S_k by masquerading as the user U_i. The service provider server S_k authenticates the received messages by calculating Q_i^* from the received messages and checks its equivalency with the received value of Q_i. After that, the server S_k generates a nonce value N_k and computes $M_{ik}1 = H(B_i|N_i|y|SID_k)$ and sends the message ($M_{ik}1$, N_k) back to the malicious server S_J who is masquerading as the user U_i. On receiving the message ($M_{ik}1$, N_k), the malicious server S_J computes $M_{ik}2 = H(B_i|N_k|y|SID_k)$ and sends $M_{ik}2$ back to the service provider server S_k. On receiving the message $M_{ik}2$, the service provider server S_k computes $M_{ik}2^* = H(B_i|N_k|y|SID_k)$ and compares it with the received value of $M_{ik}2$. This equivalency assures the legitimacy of the user U_i. After the completion of mutual authentication phase, the malicious server masquerading as the user U_i and the service provider S_k computes $SK = H(B_i|N_i|N_k|y|SID_k)$ as the session key.

4.2. Malicious User Attack

The malicious privileged user U_m can extract information like y and $B_m = H(P_m) \oplus H(x)$ from his own smart card. He can also intercept the login request message (CID_i, P_{iJ}, Q_i, N_i) of the user U_i to the service provider S_J. This malicious user U_m can compute

$$H(x) = B_m \oplus H(P_m),\ T_i = P_{iJ} \oplus H(y|N_i|SID_J),$$
$$H(P_i) = CID_i \oplus H(T_i|y|N_i)$$

and $B_i = H(P_m) \oplus H(x)$. Now this malicious user U_m can choose random nonce value N_m and computes

$$CID_i = H(P_i) \oplus H(T_i|y|N_m),\ P_{iJ} = T_i \oplus H(y|N_m|SID_J)$$

and $Q_i = H(B_i|y|N_m)$ and masquerade as the legitimate user U_i by sending the login request message (CID_i, P_{iJ}, Q_i, N_m) to the service provider server S_J. The service provider server S_J computes

$$T_i = P_{iJ} \oplus H(y|N_m|SID_J),\ H(P_i) = CID_i \oplus H(T_i|y|N_m),$$
$$B_i = H(P_i) \oplus H(x),\ Q_i^* = H(B_i|y|N_m)$$

and compares the equality of calculated value of Q_i^* with the received value of Q_i to verify the legitimacy of the user U_i. Afterwards, the server S_J generates nonce value N_J, computes $M_{iJ}1 = H(B_i|N_m|y|SID_J)$ and sends the message ($M_{iJ}1$, N_J) back to the malicious user U_m who is masquerading as the user U_i. On receiving the message ($M_{iJ}1$, N_J), the malicious user U_m computes $M_{iJ}2 = H(B_i|N_J|y|SID_J)$ and sends $M_{iJ}2$ back to the service provider server S_J. On receiving the message $M_{iJ}2$, the service provider server S_J computes $M_{iJ}2^* = H(B_i|N_J|y|SID_J)$ and compares the computed value of $M_{iJ}2^*$ with the received value of $M_{iJ}2$ to verify the legitimacy of the user U_i. After finishing mutual authentication phase, the malicious user U_m masquerading as the user U_i and the service provider server S_J computes $SK = H(B_i|N_m|N_J|y|SID_J)$ as the session key.

5. Proposed Protocol

In this section, we propose a dynamic identity based authentication protocol for multi-server architecture using smart cards that is free from all the attacks considered above. The notations used in this section are listed in **Table 2** and the protocol is summarized in **Figure 2**.

Table 2. Notations.

U_i	i^{th} User
S_k	K^{th} Service Provider Server
RC	Control Server
ID_i	Unique Identity of User U_i
P_i	Password of User U_i
H ( )	One-Way Hash Function
SID_K	Unique Identity of k^{th} Service Provider Server
y_i	Random Value chosen by CS for User U_i
x	Master Secret Parameter of Server CS
N_1	Random Nonce Value Generated by User's Smart Card
N_2	Random Nonce Value Generated by Server S_k
N_3	Random Nonce Value Generated by Server CS
$\oplus$	XOR Operation
|	Concatenation

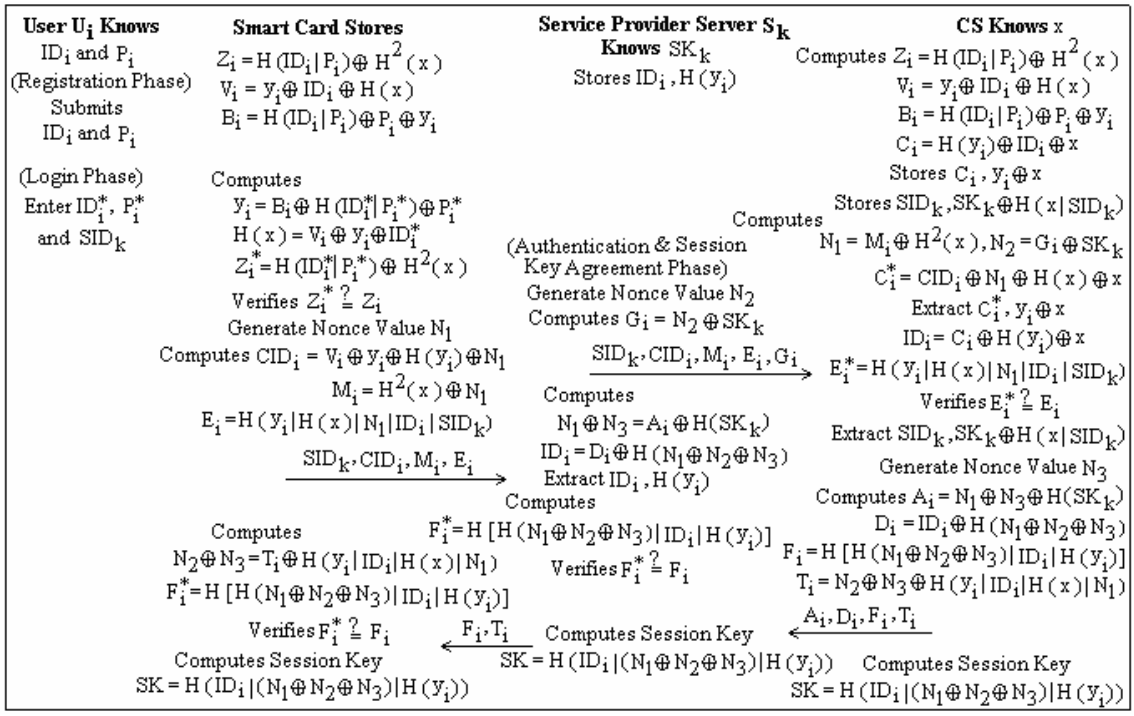

Figure 2. Dynamic identity based multi-server authentication protocol.

5.1. Registration Phase

The user U_i has to submit his identity ID_i and password P_i to the control server CS for its registration over a secure communication channel.

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

The control server CS computes the security parameters

$$Z_i = H(ID_i|P_i) \oplus H^2(x),\ V_i = y_i \oplus ID_i \oplus H(x),$$

$$B_i = H(ID_i P_i) \oplus P_i \oplus y_i$$

and $C_i = H(y_i) \oplus ID_i \oplus x$, where x is the secret key of the CS and y_i is the random value chosen by the CS for the user U_i. The server CS chooses the value of y_i corresponding to the user U_i in such a way so that the value of C_i must be unique for each user. The server CS stores $y_i \oplus x$ corresponding to C_i in its client's database. Then the server CS issues smart card containing security parameters (Z_i, V_i, B_i, H ( )) to the user U_i through a secure communication channel.

Step 2: CS $\rightarrow$ U_i: Smart card

All service provider servers register themselves with

CS and CS agrees on a unique secret key SK_k with each service provider server S_k. The server S_k remembers the secret key SK_k and CS stores the secret key SK_k as $SK_k \oplus H(x|SID_k)$ corresponding to service provider server identity SID_k in its service provider server's database.

Step 3: CS → S_k: ID_i, H (y_i)

The CS sends ID_i and H (y_i) corresponding to newly registered user U_i to all service provider servers. Each service provider server stores ID_i and H (y_i) in its database.

5.2. Login Phase

The user U_i inserts his smart card into a card reader and submits his identity ID_i^*, password P_i^* and the server identity SID_k to smart card in order to login on to the service provider server S_k. Then smart card computes

$$y_i = B_i \oplus H(ID_i^* | P_i^*) \oplus P_i^*, H(x) = V_i \oplus y_i \oplus ID_i^*,$$

$$Z_i^* = H(ID_i^* | P_i^*) \oplus H^2(x)$$

and compares the computed value of Z_i^* with the stored value of Z_i in its memory to verifies the legitimacy of the user U_i.

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

After verification, smart card generates random nonce value N_1 and computes

$$CID_i = V \oplus y_i \oplus H(y_i) \oplus N_1, M_i = H^2(x) \oplus N_1$$

and $E_i = H(y_i | H(x) | N_1 | ID_i | SID_k)$. Then smart card sends the login request message (SID_k, CID_i, M_i, E_i) to the service provider server S_k.

Step 2: Smart card → S_k: SID_k, CID_i, M_i, E_i

5.3. Authentication and Session Key Agreement Phase

After receiving the login request from the user U_i, the server S_k generates random nonce value N_2, computes $G_i = N_2$ SK_k and sends the login request message (SID_k, CID_i, M_i, E_i, G_i) to the control server CS.

Step 1: S_k → CS: SID_k, CID_i, M_i, E_i, G_i

The control server CS computes

$$N_1 = M_i \oplus H^2(x), N_2 = G_i \oplus SK_k,$$

$$C_i^* = CID_i \oplus N_1 \oplus H(x) \oplus x$$

and finds the matching value of C_i corresponding to C_i^* from its client database.

Step 2: Server CS checks C_i^* ?= C_i

If the value of C_i^* does not match with any value of C_i in its client database, the CS rejects the login request and terminates this session. Otherwise, the CS extracts y_i from y_i x corresponding to C_i^* from its client database. Then the CS computes

$$ID_i = C_i \oplus H(y_i) \oplus x, E_i^* = H(y_i | H(x) | N_1 | ID_i | SID_k)$$

and compares E_i^* with the received value of E_i to verifies the legitimacy of the user U_i and the service provider server S_k.

Step 3: Server CS checks E_i^* ?= E_i

If they are not equal, the CS rejects the login request and terminates this session. Otherwise, the CS extracts SK_k from $SK_k \oplus H(x|SID_k)$ corresponding to SID_k in its service provider server's database. Then the CS generates random nonce value N_3, computes

$$A_i = N_1 \oplus N_3 \oplus H(SK_k), D_i = ID_i \oplus H(N_1 \oplus N_2 \oplus N_3),$$

$$F_i = H[H(N_1 \oplus N_2 \oplus N_3) ID_i | H(y_i)],$$

$$T_i = N_2 \oplus N_3 \oplus H(y_i | ID_i | H(x) | N_1)$$

and sends the message (A_i, D_i, F_i, T_i) back to the service provider server S_k. The server S_k computes

$$N_1 \oplus N_3 = A_i \oplus H(SK_k) \text{ from } A_i$$

and $$ID_i = D_i \oplus H(N_1 \oplus N_2 \oplus N_3) \text{ from } D_i.$$

Then the server S_k extracts H(y_i) corresponding to ID_i from its database. Afterwards, the server S_k computes

$$F_i^* = H[H(N_1 \oplus N_2 \oplus N_3) | ID_i | H(y_i)]$$

and compares F_i^* with the received value of F_i to verifies the legitimacy of the control server CS.

Step 4: Server S_k checks F_i^* ?= F_i

Then the server S_k sends (F_i, T_i) to smart card of the user U_i. Then smart card computes

$$N_2 \oplus N_3 = T_i \oplus H(y_i | ID_i | H(x) | N_1),$$

$$F_i^* = H[H(N_1 \oplus N_2 \oplus N_3) | ID_i | H(y_i)]$$

and compares the computed value of F_i^* with the received value of F_i.

Step 5: Smart card checks F_i^* ?= F_i

This equivalency authenticates the legitimacy of the control server CS, the server S_k and the login request is accepted else the connection is interrupted. Finally, the user U_i's smart card, the server S_k and the control server CS agree on the common session key as $SK = H(ID_i | (N_1 \oplus N_2 \oplus N_3) | H(y_i))$.

5.4. Password Change Phase

The user U_i can change his password without the help of control server CS. 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. Smart card computes

$$y_i = B_i \oplus H\left(ID_i^* \| P_i^*\right) \oplus P_i^*,\ H(x) = V_i \oplus y_i \oplus ID_i^*,$$
$$Z_i^* = H\left(ID_i^* \| P_i^*\right) \oplus H^2(x)$$

and compares the computed 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 card holder is verified, the smart card asks the card holder to resubmit a new password P_i^{new}. Finally, the value of

$$Z_i = H\left(ID_i \| P_i\right) \oplus H^2(x) \text{ and } B_i = H\left(ID_i \| P_i\right) \oplus P_i \oplus y_i$$

stored in the smart card is updated with

$$Z_i^{new} = Z_i \oplus H\left(ID_i \| P_i\right) \oplus H\left(ID_i \| P_i^{new}\right)$$

and $B_i^{new} = B_i \oplus H\left(ID_i \| P_i\right) \oplus P_i \oplus H\left(ID_i \| P_i^{new}\right) \oplus P_i^{new}$.

6. 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*. [12] and Messerges *et al*. [13] pointed out that all existing smart cards can not prevent the information stored in them from being extracted like by 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 the 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) Malicious server attack: A malicious privileged server S_k can monitor the authentication process of the user U_i and can gather information related to the user U_i. The malicious server S_k can gather information

$$CID_i = V_i \oplus y_i \oplus H(y_i) \oplus N_1,\ M_i = H^2(x) \oplus N_1$$

and $E_i = H\left(y_i \| H(x) \| N_1 \| ID_i \| SID_k\right)$ during login phase corresponding to the legitimate user U_i. This malicious server S_k can not compute ID_i, y_i and x from this information. This malicious server S_k can compute the identity ID_i from D_i and can extract $H(y_i)$ corresponding to ID_i from its database corresponding to the user U_i during authentication and session key agreement phase. To masquerade as the legitimate user U_i, this malicious server S_k who knows the identity ID_i has to guess y_i and H(x) correctly at the same time. It is not possible to guess out two parameters correctly at the same time in real polynomial time. In another option, this malicious server S_k has to get smart card of the user U_i and has to guess the correct password P_i in order to login on to the server S_m. It is not possible to guess the password P_i correctly in real polynomial time even after getting the smart card of legitimate user U_i and after knowing the identity ID_i of the user U_i. Therefore, the proposed protocol is secure against malicious server attack.

2) Malicious user attack: A malicious privileged user U_i having his own smart card can gather information like

$$Z_i = H\left(ID_i \| P_i\right) \oplus H^2(x),\ V_i = y_i \oplus ID_i \oplus H(x)$$

and $B_i = H\left(ID_i \| P_i\right) \oplus P_i \oplus y_i$ from the memory of smart card. The malicious user U_i can compute the value of H(x) from this information. The value of CID_m, M_m and E_m is smart card specific and the malicious user U_i requires to know the values of H(x), y_m and ID_m to masquerade as the legitimate user U_m. Therefore, this malicious user U_i has to guess y_m and ID_m correctly at the same time. It is not possible to guess out two parameters correctly at the same time in real polynomial time. Therefore, the proposed protocol is secure against malicious user attack.

3) 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 the smart card. An attacker can extract

$$Z_i = H\left(ID_i \| P_i\right) \oplus H^2(x),\ V_i = y_i \oplus ID_i \oplus H(x)$$

and $B_i = H\left(ID_i \| P_i\right) \oplus P_i \oplus y_i$ from the memory of smart card. Even after gathering this information, an attacker has to guess minimum two parameters out of ID_i, H(x), y_i and P_i correctly at the same time. It is not possible to guess out two parameters correctly at the same time in real polynomial time. Therefore, the proposed protocol is secure against stolen smart card attack.

4) Identity protection: Our approach provides identity protection in the sense that instead of sending the real identity ID_i of the user U_i in authentication, the pseudo identification $CID_i = V_i \oplus y_i \oplus H(y_i) \oplus N_1$ is generated by smart card corresponding to the legitimate user U_i for its authentication to the service provider server S_k and the control server CS. There is no real identity information about the user during the login and authentication & session key agreement phase. This approach provides the privacy and unlinkability among different login requests belonging to the same user. The attacker can not link different sessions belonging to the same user.

5) Offline dictionary attack: In offline dictionary attack, the attacker can record messages and attempts to guess user's identity ID_i and password P_i from recorded messages. An attacker first tries to obtains identity and password verification information such as

$$Z_i = H\left(ID_i \| P_i\right) \oplus H^2(x),\ B_i = H\left(ID_i \| P_i\right) \oplus P_i \oplus y_i$$

and then try to guess the identity ID_i and password P_i by offline guessing. Here an attacker has to guess the identity ID_i and password 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 offline dictionary attack.

6) Replay attack: In this type of attack, the attacker first listens to communication between the user and the server and then tries to imitate the user to login on to the server by resending the captured messages transmitted between the user and the server. Replaying a message of one session into another session is useless because the user's smart card, the server S_k and the control server CS choose different nonce values (N_1, N_2, N_3) in each new session, which make all messages dynamic and valid for that session only. Therefore, replaying old dynamic identity and user's verifier information is useless. Moreover, the attacker can not compute the session key

$$SK = H\left(ID_i \| \left(N_1 \oplus N_2 \oplus N_3\right) \| H\left(y_i\right)\right)$$

because the user U_i's smart card, the server S_k and the control server CS contributes different nonce values (N_1, N_2, N_3) in each new session and the attacker does not know the value of ID_i, N_1, N_2, N_3 and $H(y_i)$. Therefore, the proposed protocol is secure against replay attack.

7) Mutual authentication: The goal of mutual authentication is to establish an agreed session key among the user U_i, the service provider server S_k and the control server CS. All three parties contribute their random nonce values as N_1, N_2 and N_3 for the derivation of session key $SK = H\left(ID_i \| \left(N_1 \oplus N_2 \oplus N_3\right) \| H\left(y_i\right)\right)$. The control server CS authenticates the user U_i using verifier information as $E_i^* = H\left(y_i \| H(x) \| N_1 \| ID_i \| SID_k\right)$, the service provider server S_k authenticates the server CS using

$$F_i^* = H\left[H\left(N_1 \oplus N_2 \oplus N_3\right) \| ID_i \| H\left(y_i\right)\right]$$

and the user U_i authenticates the server S_k and the server CS using $F_i^* = H\left[H\left(N_1 \oplus N_2 \oplus N_3\right) \| ID_i \| H\left(y_i\right)\right]$. The proposed protocol satisfies strong mutual authentication.

7. Cost and Functionality Analysis

An efficient authentication protocol must take communication and computation cost into consideration during user's authentication. The cost comparison of the proposed protocol with the relevant smart card based authentication protocols is summarized in **Table 3**. Assume that the identity ID_i, password P_i, x, y_i, nonce values (N_1, N_2, N_3) are all 128 bit long and prime modular operation is 1024 bits long as in most of practical implementations. Moreover, we assume that the output of secure one-way hash function and the block size of secure symmetric cryptosystem are 128 bits. Let T_H, T_{SYM} and T_{EXP} are defined as the time complexity for hash function, symmetric encryption/decryption and exponential operation respectively. Typically, time complexity associated with these operations can be roughly expressed as $T_{EXP} \gg T_{SYM} > T_H$. In the proposed protocol, the parameters stored in the smart card are Z_i, V_i, B_i and the memory needed (E1) in the smart card is $384\ (=3*128)$ bits. The communication cost of authentication (E2) includes the number of communication parameters involved in the authentication protocol. The number of communication parameters is $\{SID_k, CID_i, M_i, E_i, G_i, A_i, D_i, F_i, T_i\}$ and hence the communication cost of authentication (E2) is $1152\ (=9*128)$ bits. The computation cost of registration (E3) is the total time of all operations executed by the user U_i in the registration phase. The computation cost of registration (E3) is $4T_H$. The computation cost of the user (E4) is the time spent by the user during the process of authentication. Therefore, the computation cost of the user (E4) is $8T_H$. The computation cost of the service provider server and the control server (E5) is the time spent by the service provider server and the control server during the process of authentication. Therefore, the computation cost of the service provider server and the control server (E5) is $12T_H$.

The proposed protocol uses the control server CS and the service provider server S_k for the user's authentication that is why the computation cost of the servers (E5) is high as compared to Liao and Wang protocol [9]. On the other hand, the protocol proposed by Liao and Wang in 2009 totally relies on the service provider server S_k for the user's authentication and hence susceptible to malicious server attack and malicious user attack. The proposed protocol maintains the user's anonymity by generating dynamic identity and free from different attacks. The proposed protocol requires very less computation as compared to other related protocols and also highly secure as compared to these related protocols. The functionality comparison of the proposed protocol with the relevant smart card based authentication protocols is summarized in **Table 4**.

Table 3. Cost comparison among related smart card based authentication protocols.

	Proposed Protocol	Liao & Wang [9]	Hsiang & Shih [10]	Chang & Lee [5]	Juang [4]	Lin *et al*. [3]
E1	384 bits (0.375 |n|)	512 bits (0.5 |n|)	640 bits (0.625 |n|)	256 bits (0.25 |n|)	256 bits (0.25 |n|)	(4t + 1) |n| bits
E2	9*128 bits (1.125 |n|)	7*128 bits (0.875 |n|)	14*128 bits (1.75 |n|)	5*128 bits (0.625 |n|)	9*128 bits (1.125 |n|)	7*1024 bits (7 |n|)
E3	$4T_H \ll T$	$5T_H \ll T$	$6T_H \ll T$	$2T_H \ll < T$	$T_H << T$	5tT
E4	$8T_H \ll T$	$9T_H \ll T$	$10T_H \ll T$	$4T_H + 3T_{SYM} \ll T$	$3T_H + 3T_{SYM} \ll T$	2T
E5	$12T_H \ll T$	$6T_H \ll T$	$13T_H \ll T$	$4T_H + 3T_{SYM} \ll T$	$4T_H + 8T_{SYM} \ll T$	7T

t: Number of servers; T: Time complexity of a modular exponential communication in Z_n^* : | n | = 1024 bits.

Table 4. Functionality comparison among related smart card based authentication protocols.

	Proposed protocol	Liao & Wang [9]	Hsiang & Shih [10]	Chang & Lee [5]	Juang [4]	Lin *et al*. [3]
User's anonymity	Yes	Yes	Yes	No	No	No
Computation cost	Low	Low	Low	Low	Low	High
Single registration	Yes	Yes	Yes	Yes	Yes	No
Session key agreement	Yes	Yes	Yes	Yes	Yes	No
Correct password update	Yes	Yes	No	No	No	No
No time synchronization	Yes	Yes	Yes	Yes	Yes	No
Mutual authentication	Yes	Yes	Yes	Yes	Yes	No
Two factor security	Yes	Yes	Yes	No	No	No
Malicious server attack	No	Yes	No	Yes	Yes	No
Malicious user attack	No	Yes	Yes	Yes	Yes	No

8. Conclusion

We presented a cryptanalysis of a recently proposed Liao and Wang's protocol and showed that their protocol is susceptible to malicious server attack and malicious user attack. An improved protocol is proposed that inherits the merits of Liao and Wang's protocol and resists different possible attacks. We have specified and analyzed a dynamic identity based authentication protocol for multi-server architecture using smart cards which is very effective to thwart different attacks. The proposed protocol helps the service provider servers and the control server to recognize the user's completely by computing their static identity and at the same time keeps the identity of the user dynamic in communication channel. The proposed protocol is practical and efficient because only one-way hash function and XOR operations are used in its implementation. Security analysis proved that the proposed protocol is more secure and practical.

REFERENCES

[1] W. Ford and B. S. Kaliski, "Server-Assisted Generation of a Strong Secret from a Password," *Proceedings of IEEE 9th International Workshop Enabling Technologies*, Washington DC, June 2000, pp. 176-180.

[2] D. P. Jablon, "Password Authentication Using Multiple Servers," *Proceedings of RSA Security Conference*, London, April 2001, pp. 344-360.

[3] I. C. Lin, M. S. Hwang and L. H. Li, "A New Remote User Authentication Scheme for Multi-Server Architecture," *Future Generation Computer System*, Vol. 19, No. 1, 2003, pp. 13-22. doi:10.1016/S0167-739X(02)00093-6

[4] W. S. Juang, "Efficient Multi-Server Password Authenticated Key Agreement Using Smart Cards," *IEEE Transactions on Consumer Electronics*, Vol. 50, No. 1, 2004, pp. 251-255. doi:10.1109/TCE.2004.1277870

[5] C. C. Chang and J. S. Lee, "An Efficient and Secure Multi-Server Password Authentication Scheme Using Smart Cards," *Proceedings of International Conference on Cyber Worlds*, Washington DC, November 2004, pp. 417-422. doi:10.1109/CW.2004.17

[6] L. Hu, X. Niu and Y. Yang, "An Efficient Multi-Server Password Authenticated Key Agreement Scheme Using Smart Cards," *Proceedings of International Conference on Multimedia and Ubiquitous Engineering* (*MUE*'07), April 2007, pp. 903-907. doi:10.1109/MUE.2007.70

[7] Y. Yang, R. H. Deng and F. Bao, "A Practical Password-Based Two-Server Authentication and Key Exchange System," *IEEE Transactions on Dependable and Secure Computing*, Vol. 3, No. 2, 2006, pp. 105-114. doi:10.1109/TDSC.2006.16

[8] J. L. Tsai, "Efficient Multi-Server Authentication Scheme Based on One-Way Hash Function without Verification Table," *Computers & Security*, Vol. 27, No. 3-4, 2008, pp. 115-121. doi:10.1016/j.cose.2008.04.001

[9] Y. P. Liao and S. S. Wang, "A Secure Dynamic ID-Based Remote User Authentication Scheme for Multi-Server Environment," *Computer Standards & Interface*, Vol. 31, No. 1, 2009, pp. 24-29. doi:10.1016/j.csi.2007.10.007

[10] H. C. Hsiang and W. K. Shih, "Improvement of the Secure Dynamic ID Based Remote User Authentication Scheme for Multi-Server Environment," *Computer Standards & Interface*, Vol. 31, No. 6, 2009, pp. 1118-1123. doi:10.1016/j.csi.2008.11.002

[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*, Vol. 34, No. 2, 2011, pp. 609-618. doi:10.1016/j.jnca.2010.11.011

[12] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis," *Proceedings of CRYPTO* 99, Springer-Verlag, August 1999, pp. 388-397.

[13] 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, 2002, pp. 541-552. doi:10.1109/TC.2002.1004593

