Journal of Network and Computer Applications 34 (2011) 609–618

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A secure dynamic identity based authentication protocol for multi-server architecture

Sandeep K. Sood *, Anil K. Sarje, Kuldip Singh

Department of Electronics & Computer Engineering, Indian Institute of Technology, Roorkee, India

ARTICLE INFO

Article history:
Received 13 June 2010
Received in revised form
22 October 2010
Accepted 18 November 2010
Available online 30 November 2010

Keywords:
Authentication protocol
Smart card
Dynamic identity
Password
Multi-server architecture

ABSTRACT

Most of the password based authentication protocols rely on single authentication server for the user's authentication. User's verification information stored on the single server is a main point of susceptibility and remains an attractive target for the attacker. In 2009, Hsiang and Shih improved Liao and Wang's dynamic identity based smart card authentication protocol for multi-server environment. However, we found that Hsiang and Shih's protocol is susceptible to replay attack, impersonation attack and stolen smart card attack. Moreover, the password change phase of Hsiang and Shih's protocol is incorrect. This paper presents a secure dynamic identity based authentication protocol for multi-server architecture using smart cards that resolves the aforementioned security flaws, while keeping the merits of Hsiang and Shih's protocol. It uses two-server paradigm in which different levels of trust are assigned to the servers and the user's verifier information is distributed between these two servers known as the service provider server and the control server. The service provider server is more exposed to the clients than the control server. The back-end control server is not directly accessible to the clients and thus it is less likely to be attacked. The user's smart card uses stored information in it and random nonce value to generate dynamic identity. The proposed protocol is practical and computationally efficient because only nonce, one-way hash functions 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.

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.

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 Protected Extensible Authentication Protocol jointly developed by Cisco Systems, Microsoft and RSA Security is the most widely used authentication protocol. It encapsulates the Extensible Authentication Protocol within an encrypted and authenticated Transport Layer Security (TLS) tunnel. This protocol is included with Microsoft Windows XP and Windows 7 operating systems. It is based on the single server authentication concept. On the other hand, Kerberos is the multi-server authentication protocol. The limitation of Kerberos protocol is that all the servers are equally exposed to the user. The proposed protocol uses multi-server model consisting of two servers that work together to authenticate the users. Yang et al. (2006) also suggested similar kind of two-server model for user's authentication. In the proposed protocol, 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 reasonable 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

* Corresponding author. Tel.: +91 1894276861.
E-mail addresses: san1198@gmail.com, ssooddec@iitr.ernet.in (S.K. Sood), sarjefec@iitr.ernet.in (A.K. Sarje), ksconfcn@iitr.ernet.in (K. Singh).

1084-8045/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2010.11.011

schemes (Das et al., 2004; Chien and Chen, 2005; Liao et al., 2005; Yoon and Yoo, 2006; Liou et al., 2006). These conventional single-server password authentication protocols cannot 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 (Yang et al., 2006; Ford and Kaliski, 2000; Jablon, 2001; Lee and Chang, 2000; Li et al., 2001; Lin et al., 2003; Raimondo and Gennaro, 2003; Brainard et al., 2003; Juang, 2004; Chang and Lee, 2004; Hu et al., 2007; Tsaur et al., 2004; Yang et al., 2005; Mackenzie et al., 2006; Tsai, 2008; Liao and Wang, 2009; Hsiang and Shih (2009)). 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.

Password is the most commonly used authentication technique in authentication protocols. Low entropy password makes system susceptible to dictionary attack. 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 cannot change his identity in password authentication protocols. During communication, the static identity leaks out partial information about 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 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 dynamic identity based authentication protocols using smart cards and authentication protocols for multi-server environment. Section 3 reviews the dynamic identity based remote user authentication protocol for multi-server environment proposed by Hsiang and Shih (2009). Section 4 describes the susceptibility of Hsiang and Shih's protocol to replay attack, impersonation attack and stolen smart card 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. Therefore in 2004, Das et al. proposed a dynamic identity based remote user authentication protocol to authenticate the users that preserves the user's anonymity. Their protocol uses dynamic identity to achieve this purpose and 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 server's help. Das et al. claimed that their protocol is secure against stolen verifier attack, replay attack, forgery attack, guessing attack, insider attack and identity theft. However, many researchers Chien and Chen (2005); Liao et al. (2005); Yoon and Yoo (2006); Liou et al. (2006); Shih (2008) demonstrated susceptibility of Das et al.'s protocol to different attacks. In 2005, Chien and Chen pointed out that Das et al.'s protocol fails to preserve the user anonymity effectively because the authentication messages belonging to the same user can be identified. They proposed an authentication protocol and claimed that the proposed protocol preserves user's anonymity more efficiently. Though their protocol preserves user's anonymity and secure against various attacks but it is highly computation intensive. In 2005, Liao et al. proposed an improved protocol that enhances the security of Das et al.'s protocol and achieves mutual authentication. In 2006, Yoon and Yoo demonstrated a reflection attack on Liao et al.'s protocol that breaks the mutual authentication. They also proposed an improved dynamic identity based mutual authentication protocol that eliminates the security flaws of Liao et al.'s protocol. In 2006, Liou et al. suggested a new dynamic identity based remote user authentication protocol using smart cards that achieves mutual authentication. They claimed that their protocol preserves the advantages of Das et al.'s protocol and overcomes the weaknesses of Das et al.'s protocol. In 2008, Shih demonstrated that Liou et al.'s protocol fails to achieve mutual authentication.

In 2000, Ford and Kaliski proposed the first multi-server password based authentication protocol that splits a password among multiple servers. This protocol generates a strong secret with the help of password based on the communications exchanges with two or more independent servers. The attacker cannot 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 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 2000, Lee and Chang proposed a user identification and key distribution protocol for multi-server environment based on the hash function and difficulty of factorization. In 2001, Li et al. proposed a remote password authentication protocol for multi-server environment. This password authentication system is a pattern classification system based on an artificial neural network. The user has to register with registration center once and then can obtain services from multiple servers without needing to register individually with each server. The users can choose their passwords freely and the server does not require to keep any verification table. This protocol can withstand the replay attack effectively but it requires intensive communication and computation efforts.

In 2003, Lin et al. 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 to keep any verification table but the use of public keys makes this protocol computation intensive. In 2003, Raimondo and Gennaro proposed two multi-server password authentication protocols in which the user has to communicate in parallel with all authentication servers. They proved that these protocols are provable secure in the standard model. The attacker has to compromise minimum threshold number of servers to gain any meaningful information regarding the password of the user. These two protocols differ in the way the client interacts with the different servers. In these protocols, the servers are equally exposed to the user as well as to the attacker. In 2003, Brainard et al. proposed a password based two-server authentication protocol in which only one server was exposed to the users. The use of public keys makes this system computationally intensive. Moreover, it uses Secure Socket Layer (SSL) to establish a session key between a user and the front-end server to provide authentication but it provides only unilateral authentication.

In 2004, Juang 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 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 (2004). In 2007, Hu et al. 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 (2004) protocol. In 2004, Tsaur et al. proposed a smart card based multi-server authentication protocol that uses the RSA cryptosystem and Lagrange interpolating polynomial without using any password verification table. This protocol involves high communication and computation costs. In 2005, Yang et al. proposed two-server based password authentication and key exchange protocol in which the back-end control server is managed by an enterprise head quarter and each affiliating organization operates a front-end external server. The back-end control server requires public key for its operations. The attacker has to compromise both the servers simultaneously to launch offline dictionary attack.

In 2006, Yang et al. 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 2006, Mackenzie et al. proposed an efficient password-authenticated key exchange protocol that uses a set of servers with known public keys so that a certain threshold number of servers must participate to authenticate a user. Therefore, the attacker has to compromise the minimum threshold number of servers to launch offline dictionary attack. The use of public key makes this protocol computation intensive. In 2008, Tsai proposed a multi-server authentication protocol using smart cards based on the nonce and one-way hash function that does not require to store 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 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. However, Liao and Wang's protocol is found to be susceptible to malicious server attack and malicious user attack. In 2009, Hsiang and Shih 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. However, we show in Section 4 that their protocol is insecure in the presence of an active attacker.

3. Review of Hsiang and Shih protocol (2009)

In this section, we describe the dynamic identity based remote user authentication protocol for multi-server environment proposed by Hsiang and Shih (2009). Their protocol includes four phases (registration, login, mutual verification & session key agreement and password change). The notations used in this section are listed in Table 1 and the protocol is shown in Fig. 1.

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 & r	Secret number known to registration center
$\oplus$	XOR operation
$\|$	Concatenation

3.1. Registration phase

The user U_i selects a random number b, computes $E_i = H(b \oplus P_i)$ and submits ID_i and E_i to the registration center RC for registration over a secure communication channel.

Step 1: $U_i \to \text{RC}$: ID_i, E_i
The RC computes the security parameters $T_i = H(\text{ID}_i | x)$, $V_i = T_i \oplus H(\text{ID}_i \oplus H(b \oplus P_i))$, $A_i = H(H(b \oplus P_i)|r) \oplus H(x \oplus r)$, $B_i = A_i \oplus H(b \oplus P_i)$, $R_i = H(H(b \oplus P_i)|r)$ and $H_i = H(T_i)$. Then the RC issues the smart card containing security parameters (V_i, B_i, H_i, R_i, $H(\ )$) to the user U_i through a secure communication channel.
Step 2: $\text{RC} \to U_i$: Smart card
After that, user U_i enters the value of b in his smart card. Finally, the smart card contains security parameters as (V_i, B_i, H_i, R_i, $H(\ )$, b) stored in its memory.
Step 3: $U_i \to$ Smart card: b
All service provider servers register themselves with RC. The RC computes $H(\text{SID}_J|y)$ for service provider server S_J and sends this information to the server S_J over a secure communication channel. Similarly RC computes these server specific keys for all service provider servers and sends to them over a secure communication channel.

3.2. Login phase

The user U_i inserts his smart card into a card reader to login on to the server S_J and submits his identity ID_i^*, password P_i^* and server identity SID_J. The smart card computes $T_i^* = V_i \oplus H(\text{ID}_i^* \oplus H(b \oplus P_i^*))$, $H_i^* = H(T_i^*)$ and compares H_i^* with the stored value of H_i in its memory to verify the legitimacy of the user U_i.

Step 1: Smart card checks $H_i^* ?= H_i$
After verification, smart card generates random nonce value N_i and computes $A_i = B_i \oplus H(b \oplus P_i)$, $\text{CID}_i = H(b \oplus P_i) \oplus H(T_i|A_i|N_i)$, $P_{iJ} = T_i \oplus H(A_i|N_i|\text{SID}_J)$, $Q_i = H(B_i|A_i|N_i)$, $D_i = R_i \oplus \text{SID}_J \oplus N_i$ and $C_0 = H(A_i|N_i+1|\text{SID}_J)$. Afterwards, Smart card sends the login request message (CID_i, P_{iJ}, Q_i, D_i, C_0, N_i) to the service provider server S_J.
Step 2: Smart card $\to S_J$: CID_i, P_{iJ}, Q_i, D_i, C_0, N_i

3.3. Mutual verification and session key agreement phase

The service provider server S_J generates random nonce value N_{Jr}, computes $M_{\text{Jr}} = H(\text{SID}_J|y) \oplus N_{\text{Jr}}$ and then sends the message (M_{Jr}, SID_J, D_i, C_0, N_i) to the registration center RC.

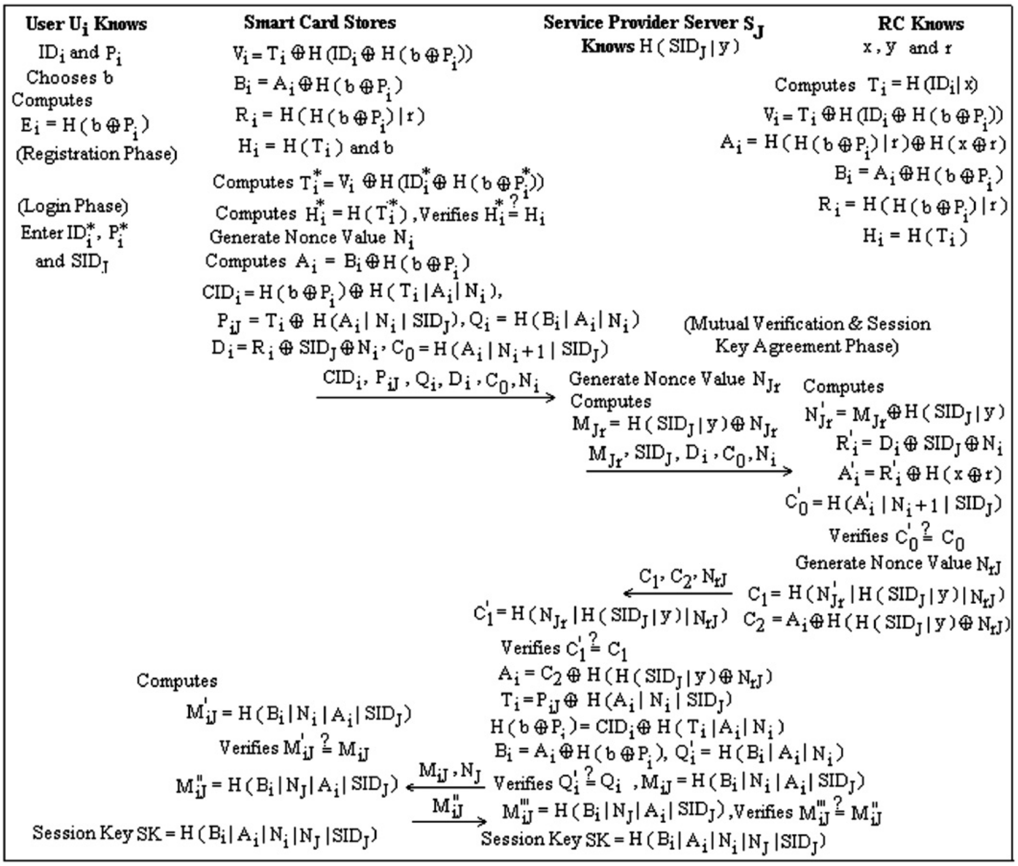

Fig. 1. Hsiang and Shih's dynamic identity based multi-server authentication protocol.

Step 1: $S_J \rightarrow$ RC: M_{Jr}, SID_J, D_i, C_0, N_i

On receiving the message (M_{Jr}, SID_J, D_i, C_0, N_i), the RC computes $N_{Jr}'=M_{Jr}\oplus H(\mathrm{SID}_J|y)$, $R_i'=D_i\oplus \mathrm{SID}_J\oplus N_i$, $A_i'=R_i'\oplus H(x\oplus r)$, $C_0'=H(A_i'|N_i+1|\mathrm{SID}_J)$ and compares the computed value of C_0' with the received value of C_0. If they are not equal, the registration center RC rejects the login request and terminates this session.

Step 2: Registration center checks $C_0'\ ?=C_0$

Otherwise the RC generates nonce value N_{rJ} and computes $C_1=H(N_{Jr}'|H(\mathrm{SID}_J|y)|N_{rJ})$, $C_2=A_i\oplus H(\mathrm{H}(\mathrm{SID}_J|y)\oplus N_{rJ})$ and sends the message (C_1, C_2, N_{rJ}) back to the server S_J. On receiving the message (C_1, C_2, N_{rJ}), the service provider server S_J computes $C_1'=H(N_{Jr}|H(\mathrm{SID}_J|y)|N_{rJ})$ and compares the computed value of C_1' with the received value of C_1. If they are not equal, the service provider server S_J rejects the login request and terminates this session.

Step 3: Service provider server S_J checks $C_i'?=C_i$

Then the server S_J computes $A_i=C_2\oplus H(H(\mathrm{SID}_J|y)\oplus N_{rJ})$, $T_i=P_{iJ}\oplus H(A_i|N_i|\ \mathrm{SID}_J)$, $H(b\oplus P_i)=\mathrm{CID}_i\oplus H(T_i|A_i|N_i)$, $B_i=A_i\oplus H(b\oplus P_i)$, $Q_i'=H(B_i|A_i|N_i)$ and compares the computed value of Q_i' with the value of Q_i received in login request message. If they are not equal, the server S_J rejects the login request and terminates this session.

Step 4: Service provider server S_J checks $Q_i'?=Q_i$

Otherwise the server S_J generates random nonce value N_J, computes $M_{iJ}=H(B_i|N_i|A_i|\mathrm{SID}_J)$ and sends the message (M_{iJ}, N_J) back to smart card of the user U_i. On receiving the message (M_{iJ}, N_J), the user U_i's smart card computes $M_{iJ}'=H(B_i|N_i|A_i|\mathrm{SID}_J)$ and compares it with the received value of M_{iJ}. If they are not equal, the user U_i's smart card rejects the login request and terminates this session.

Step 5: Smart card checks $M_{iJ}'?=M_{iJ}$

Otherwise the user U_i's smart card computes $M_{iJ}''=H(B_i|N_J|A_i|\mathrm{SID}_J)$ and sends the message M_{iJ}''' back to the service provider server S_J. Then the server S_J computes $M_{iJ}'''=H(B_i|N_J|A_i|\mathrm{SID}_J)$ and compares it with the received value of $M_{iJ''}$. If they are not equal, the server S_J rejects the login request and terminates this session.

Step 6: Service provider server S_J checks $M_{iJ}'''?= M_{iJ}''$

This equivalency authenticates the legitimacy of the user U_i and the login request is accepted else the connection is interrupted. Finally after mutual authentication, the user U_i's smart card and the server S_J agree on the common session key as $\mathrm{SK}=H(B_i|A_i|N_i|N_J|\mathrm{SID}_J)$.

3.4. Password change phase

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. Then smart card computes $T_i^*=V_i\oplus H(\mathrm{ID}_i^*\oplus H(b\oplus P_i^*))$, $H_i^*=H(T_i^*)$ and compares the computed value of H_i^* with the stored value of H_i in its memory to verify the legitimacy of the user U_i. Once the authenticity of card holder is verified then the user U_i can instruct smart card to change his password. Afterwards, smart card asks the card holder to resubmit a new password P_i^{new}, then $V_i=T_i\oplus H(\mathrm{ID}_i\oplus H(b\oplus P_i))$ and $B_i=H(H(b\oplus P_i)|r)\oplus H(x\oplus r)\oplus H(b\oplus P_i)$ stored in smart card can be updated with $V_i^{\mathrm{new}}=T_i\oplus H(\mathrm{ID}_i\oplus H(b\oplus P_i^{\mathrm{new}}))$, $B_i^{\mathrm{new}}=B_i\oplus H(b\oplus P_i)\oplus H(b\oplus P_i^{\mathrm{new}})$ and password gets changed.

4. Cryptanalysis of Hsiang and Shih protocol

Hsiang and Shih (2009) claimed that their protocol provides identity privacy and can resist various known attacks. This protocol protects the identity of the user efficiently. However, we found that this protocol is flawed for replay attack, impersonation attack and stolen smart card attack. Moreover, the password change phase of Hsiang and Shih's protocol is incorrect.

4.1. Replay attack

A malicious privileged user U_k having his own smart card can gather information (V_k, B_k, H_k, R_k, $H(\ )$, b_k) from his own smart card. He can compute the value of A_k as $A_k=B_k\oplus H(b_k\oplus P_k)$ because this malicious user U_k knows the value of b_k and his own password P_k corresponding to his smart card. Then this malicious user U_k can compute the value of $H(x\oplus r)$ as $H(x\oplus r)=A_k\oplus R_k$. Now this malicious user U_k can intercept a valid login request message (CID_i, P_{ij}, Q_i, D_i, C_0, N_i) of the user U_i from the public communication channel. Then the malicious user U_k can compute $R_i=D_i\oplus SID_J\oplus N_i$, $A_i=R_i\oplus H(x\oplus r)$, $T_i=P_{ij}\oplus H(A_i|N_i|SID_J)$, $H(b\oplus P_i)=CID_i\oplus H(T_i|A_i|N_i)$ and $B_i=A_i\oplus H(b\oplus P_i)$ corresponding to the user U_i. The malicious user U_k can replay this valid login request message (CID_i, P_{ij}, Q_i, D_i, C_0, N_i) to the server S_J by masquerading as the user U_i at some time latter. This valid login request message is verified by the registration center RC and the server S_J. After verification of login request message, the server S_J computes $M_{ij}=H(B_i|N_i|A_i|SID_J)$ and sends the message (M_{ij}, N_J) to the user U_k who is masquerading as the user U_i. The masquerading user U_k can verify the received value of M_{ij} because he knows the values of B_i, N_i, A_i and SID_J. Then the masquerading user U_k can compute $M_{ij}''=H(B_i|N_J|A_i|SID_J)$ and sends the message M_{ij}'' back to the server S_J. Then the server S_J computes $M_{ij}'''=H(B_i|N_J|A_i|SID_J)$ and verifies it with the received value of M_{ij}''. This equivalency authenticates the legitimacy of the user U_i, the service provider server S_J and the login request is accepted. Finally after mutual authentication, the malicious user U_k masquerading as the user U_i and the server S_J agree on the common session key as $SK=H(B_i|A_i|N_i|N_J|SID_J)$.

4.2. Impersonation attack

A malicious privileged user U_k having his own smart card can gather information (V_k, B_k, H_k, R_k, $H(\ )$, b_k) from his own smart card. He can compute the value of $H(x\oplus r)$ as shown in the replay attack. Now this malicious user U_k can intercept a valid login request message (CID_i, P_{ij}, Q_i, D_i, C_0, N_i) of the user U_i from the public communication channel. Then the malicious user U_k can compute $R_i=D_i\oplus SID_J\oplus N_i$, $A_i=R_i\oplus H(x\oplus r)$, $T_i=P_{ij}\oplus H(A_i|N_i|\ SID_J)$, $H(b\oplus P_i)=CID_i\oplus H(T_i|A_i|N_i)$ and $B_i=A_i\oplus H(b\oplus P_i)$ corresponding to the user U_i. This malicious user U_k can choose random nonce value N_i' and computes $CID_i=H(b\oplus P_i)\oplus H(T_i|A_i|N_i')$, $P_{im}=T_i\oplus H(A_i|N_i'|SID_m)$, $Q_i=H(B_i|A_i|N_i')$, $D_i=R_i\oplus SID_m\oplus N_i'$ and $C_0=H(A_i|N_i'+1|SID_m)$. Now this malicious user U_k can send valid login request message (CID_i, P_{im}, Q_i, D_i, C_0, N_i') by masquerading as the user U_i to the server S_m. This valid login request message is verified by the registration center RC and the server S_m. After verification of login request message, the server S_m computes $M_{im}=H(B_i|N_i'|A_i|SID_m)$ and sends the message (M_{im}, N_m) to the user U_k who is masquerading as the user U_i. The masquerading user U_k can verify the received value of M_{im} because he knows the values of B_i, N_i', A_i and SID_m. Then the masquerading user U_k can compute $M_{im}''=H(B_i|N_m|A_i|SID_m)$ and sends the message M_{im}'' back to the server S_m. Then the server S_m computes $M_{im}'''=H(B_i|N_m|A_i|SID_m)$ and verifies it with the received value of M_{im}''. This equivalency authenticates the legitimacy of the user U_i, the service provider server S_m and the login request is accepted. Finally after mutual authentication, the malicious user U_k masquerading as the user U_i and the server S_m agree on the common session key as $SK=H(B_i|A_i|N_i'|N_m|SID_m)$.

4.3. Stolen smart card attack

A malicious privileged user U_k having his own smart card can gather information (V_k, B_k, H_k, R_k, $H(\ )$, b_k) from his own smart card. He can find out the value of $H(x\oplus r)$ as shown in the replay attack. Now this malicious user U_k can intercept a valid login request message (CID_i, P_{ij}, Q_i, D_i, C_0, N_i) of the user U_i from the public communication channel. Then the malicious user U_k can compute $R_i=D_i\oplus SID_J\oplus N_i$, $A_i=R_i\oplus H(x\oplus r)$, $T_i=P_{ij}\oplus H(A_i|N_i|SID_J)$, $H(b\oplus P_i)=CID_i\oplus H(T_i|A_i|N_i)$ and $B_i=A_i\oplus H(b\oplus P_i)$ corresponding to the user U_i.

1. In case the user U_i's smart card is stolen by this malicious user U_k, he can extract the information (V_i, B_i, H_i, R_i, $H(\ )$,b) from the memory of smart card.
2. Then the malicious user U_k can launch offline dictionary attack on $V_i=T_i\oplus H(ID_i\oplus H(b\oplus P_i))$ to know the identity ID_i of the user U_i because the malicious user U_k knows the values of T_i and $H(b\oplus P_i))$ corresponding to the user U_i.
3. Then this malicious user U_k can launch offline dictionary attack on $H(b\oplus P_i)$ to know the password P_i of the user U_i because the malicious user U_k knows the value of b from smart card of the user U_i.

Now this malicious user U_k possesses the valid smart card of user U_i, knows the identity ID_i, password P_i corresponding to the user U_i and hence can login on to any service provider server.

4.4. Incorrect password change phase

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. Then smart card computes $T_i^*=V_i\oplus H(ID_i^*\oplus H(b\oplus P_i^*))$, $H_i^*=H(T_i^*)$ and compares H_i^* with the stored value of H_i in its memory to verify the legitimacy of the user U_i. Once the authenticity of card holder is verified then the user U_i can instruct smart card to change his password. Afterwards, smart card asks the card holder to resubmit a new password P_i^{new}, then $V_i=T_i\oplus H(ID_i\oplus H(b\oplus P_i))$ and $B_i=H(H(b\oplus P_i)|r)\oplus H(x\oplus r)\oplus H(b\oplus P_i))$ stored in the smart card can be updated with $V_i^{new}=T_i\oplus H(ID_i\oplus H(b\oplus P_i^{new}))$ and $B_i^{new}=B_i\oplus H(b\oplus P_i)\oplus H(b\oplus P_i^{new})=H(H(b\oplus P_i)|r)\oplus H(x\oplus r)\oplus H(b\oplus P_i^{new})$. The B_i^{new} value contains older password P_i in $H(H(b\oplus P_i)|r)$. Therefore, the modified B_i^{new} is not correct. Moreover, smart card of the user U_i does not know the value of r and hence cannot compute the correct value of B_i^{new}. Moreover, the value of $R_i=H(H(b\oplus P_i)|r)$ also contains password P_i, which has not been updated by smart card of the user U_i in password change phase. Smart card does not know the value of r and hence cannot compute the correct new R_i^{new} value. Therefore, the password change phase of Hsiang and Shih's protocol is incorrect.

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 legitimate user U_i can easily login on to the service provider server using his smart card, identity and password. The notations used in this section are listed in Table 2. This protocol consists of four phases (i.e. registration, login, authentication & session key agreement and password change) as summarized in Fig. 2.

1. Registration phase: When the user U_i wants to become a legal client, the user U_i has to submit his identity and password verifier information to the control server CS via a secure communication channel. Then the CS chooses and computes some security parameters and stores them on the smart card of the user U_i. Then the CS issues smart card to the user U_i. Also the user U_i computes and stores some security parameters on his smart card.
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 identity SID_k of

service provider server S_k to login on to the service provider server S_k. Smart card verifies authenticity of the user U_i and sends user's and server's verifier information to the destination server S_k.

3. Authentication and session key agreement phase: The service provider server S_k forwards user's and server's verifier information to the CS. Once CS authenticates the user U_i and the service provider server S_k then the CS sends some security parameters back to the server S_k. The server S_k verifies the authenticity of the CS using these security parameters. Then the server S_k sends some security parameters back to smart card of the user U_i. Using these security parameters, smart card of the user U_i verifies the legitimacy of the server S_k and the CS. Finally the CS, the service provider server S_k and the user U_i agree on the common session key.
4. Password change phase: The user U_i has to authenticate itself to smart card before requesting the password change.

5.1. Registration phase

The user U_i selects a random number b, computes $A_i=H(ID_i|b)$, $B_i=H(b\oplus P_i)$ and submits A_i and B_i to the control server CS for registration over a secure communication channel.

Table 2
Notations.

U_i	ith User
S_k	kth Service provider server
CS	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 kth 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
$\mid$	Concatenation

Step 1: $U_i \rightarrow$ CS: A_i, B_i
The CS computes the security parameters $F_i=A_i\oplus y_i$, $G_i=B_i\oplus H(y_i)\oplus H(x)$ and $C_i=A_i\oplus H(y_i)\oplus x$, 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. Then the CS stores $y_i\oplus x$ corresponding to C_i in its client's database. Then the CS issues smart card containing security parameters (F_i, G_i, $H(\ )$) to the user U_i through a secure communication channel.
Step 2: CS $\rightarrow U_i$: Smart card
After that, the user U_i computes security parameters $D_i=b\oplus H(ID_i \mid P_i)$, $E_i=H(ID_i \mid P_i)\oplus P_i$ and enters the value of D_i and E_i in his smart card. Finally, the smart card contains security parameters as (D_i, E_i, F_i, G_i, $H(\ )$) stored in its memory.
Step 3: $U_i \rightarrow$ Smart card: D_i, E_i
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 $SK_k\oplus H(x \mid SID_k)$ corresponding to service provider server identity SID_k in its service provider server's database.

5.2. Login phase

The user U_i inserts his smart card into a card reader to login on to the server S_k and submits his identity $ID_i{}^*$, password $P_i{}^*$ and server identity SID_k. The smart card computes $E_i{}^*=H(ID_i{}^*|P_i{}^*)\oplus P_i{}^*$ and compares it with the stored value of E_i in its memory to verify the legitimacy of the user U_i.

Step 1: Smart card checks $E_i{}^*\ ?=E_i$
After verification, smart card generates random nonce value N_1 and computes $b=D_i\oplus H(ID_i|P_i)$, $A_i=H(ID_i|b)$, $B_i=H(b\ \oplus P_i)$, $y_i=F_i\oplus A_i$, $H(x)=G_i\oplus B_i\oplus H(y_i)$, $Z_i=H^2(x)\oplus N_1$, $CID_i=A_i\oplus H(y_i)\oplus H(x)\oplus N_1$ and $M_i=H(H(x)|y_i|SID_k|N_1)$. Then smart card sends the login request message (SID_k, Z_i, CID_i, M_i) to the service provider server S_k.
Step 2: Smart card $\rightarrow S_k$: SID_k, Z_i, CID_i, M_i

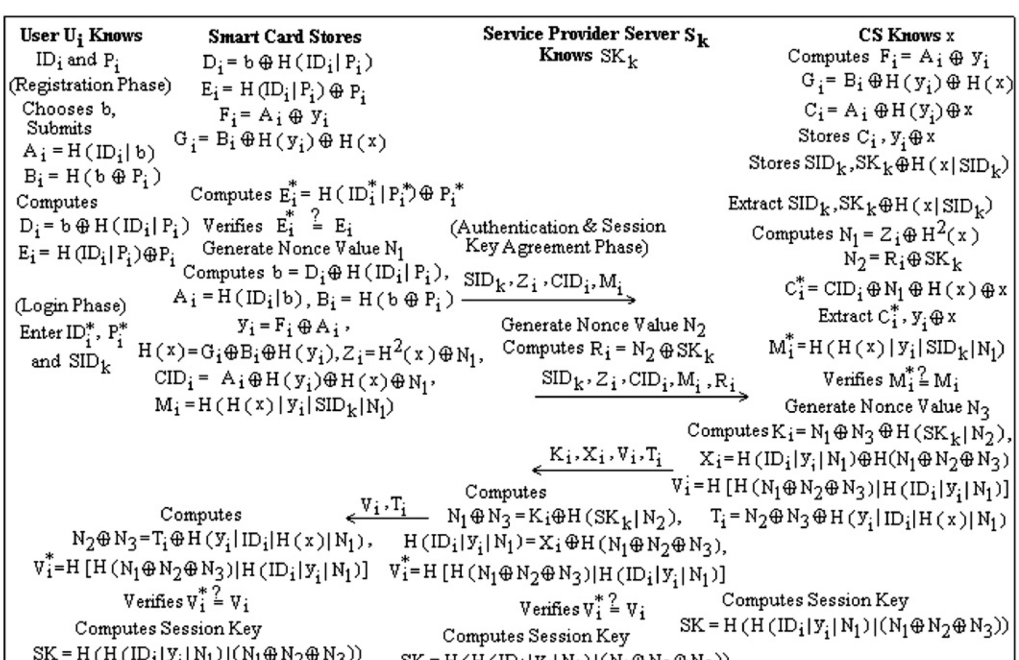

Fig. 2. Dynamic identity based multi-server authentication protocol.

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 $R_i = N_2 \oplus SK_k$ and sends the login request message (SID_k, Z_i, CID_i, M_i, R_i) to the CS.

Step 1: $S_k \rightarrow$ CS: SID_k, Z_i, CID_i, M_i, R_i
The CS extracts SK_k from $SK_k \oplus H(x | SID_k)$ corresponding to SID_k in its service provider server's database. The CS computes $N_1 = Z_i \oplus H^2(x)$, $N_2 = R_i \oplus SK_k$, $C_i^* = CID_i \oplus N_1 \oplus H(x) \oplus x$ 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 $y_i \oplus x$ corresponding to C_i^* from its client database. Then the CS computes $M_i^* = H(H(x) | y_i | SID_k | N_1)$ and compares M_i^* with the received value of M_i to verify the legitimacy of the user U_i and the service provider server S_k.
Step 3: Control server CS checks $M_i^* ?= M_i$
If they are not equal, the control server CS rejects the login request and terminates this session. Otherwise, the CS generates random nonce value N_3, computes $K_i = N_1 \oplus N_3 \oplus H(SK_k | N_2)$, $X_i = H(ID_i | y_i | N_1) \oplus H(N_1 \oplus N_2 \oplus N_3)$, $V_i = H[H(N_1 \oplus N_2 \oplus N_3) | H(ID_i | y_i | N_1)]$, $T_i = N_2 \oplus N_3 \oplus H(y_i | ID_i | H(x) | N_1)$ and sends the message (K_i, X_i, V_i, T_i) back to the service provider server S_k. The server S_k computes $N_1 \oplus N_3 = K_i \oplus H(SK_k | N_2)$ from K_i and $H(ID_i | y_i | N_1) = X_i \oplus H(N_1 \oplus N_2 \oplus N_3)$ from X_i. Then the server S_k computes $V_i^* = H[H(N_1 \oplus N_2 \oplus N_3) | H(ID_i | y_i | N_1)]$ and compares the computed value of V_i^* with the received value of V_i to verify the legitimacy of the control server CS.
Step 4: Server S_k checks $V_i^* ?= V_i$
Then the server S_k sends (V_i, T_i) to smart card of the user U_i. Then smart card computes $N_2 \oplus N_3 = T_i \oplus H(y_i | ID_i | H(x) | N_1)$, $V_i^* = H[H(N_1 \oplus N_2 \oplus N_3) | H(ID_i | y_i | N_1)]$ and compares the computed value of V_i^* with the received value of V_i.
Step 5: Smart card checks $V_i^* ?= V_i$
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 $SK = H(H(ID_i | y_i | N_1) | (N_1 \oplus N_2 \oplus N_3))$. Afterwards, all the subsequent messages between the user U_i, the server S_k and the CS are XOR^{ed} with the session key. Therefore, either the user U_i or the server S_k or the server CS can retrieve the original message because all of them know the common session key.

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 $E_i^* = H(ID_i^* | P_i^*) \oplus P_i^*$ and compares the computed value of E_i^* with the stored value of E_i in its memory to verify the legitimacy of the user U_i. Once the authenticity of card holder is verified, smart card computes the values of b, $H(y_i)$, $H(x)$ and then asks the card holder to resubmit a new password P_i^{new}. Finally, the value of $D_i = b \oplus H(ID_i | P_i)$, $E_i = H(ID_i | P_i) \oplus P_i$ and $G_i = H(b \oplus P_i) \oplus H(y_i) \oplus H(x)$ stored in the smart card is updated with $D_i^{new} = b \oplus H(ID_i | P_i^{new})$, $E_i^{new} = H(ID_i | P_i^{new}) \oplus P_i^{new}$ and $G_i^{new} = H(b \oplus P_i^{new}) \oplus H(y_i) \oplus H(x)$ and password gets changed. Hsiang and Shih's protocol cannot update the values of B_i and R_i correctly because smart card does not know the value of r and hence cannot compute the correct new values of B_i^{new} and R_i^{new}. Therefore, the password change phase of Hsiang and Shih's protocol is incorrect. On the other hand, our proposed protocol can update the values of D_i, E_i and G_i stored in the smart card with D_i^{new}, E_i^{new} and G_i^{new} successfully and password gets changed.

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. (1999) and Messerges et al. (2002) pointed out that all existing smart cards cannot 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. In our proposed protocol, the password verification information is distributed between service provider server and control server. Therefore, the attacker cannot launch an attack even by capturing one of the servers out of service provider server and control sever. Moreover, practically it is not possible for the attacker to capture both servers (service provider server and control server). A good password authentication scheme should provide protection from different possible attacks relevant to that protocol.

1. *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 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 cannot compute the session key $SK = H(H(ID_i | y_i | N_1) | (N_1 \oplus N_2 \oplus N_3))$ 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 values of ID_i, y_i, N_1, N_2 and N_3. Therefore, the proposed protocol is secure against replay attack.
2. *Impersonation attack:* In this type of attack, the attacker impersonates as the legitimate user and forges the authentication messages using the information obtained from the authentication protocol. An attacker has to guess A_i, $H(x)$ and y_i to masquerades as a legitimate user U_i to login on to the service provider server S_k to access the resources of the server S_k. It is not possible to guess all these parameters correctly at the same time in real polynomial time. Moreover, the attacker cannot compute A_i, $H(x)$ and y_i from intercepted communication parameters Z_i, CID_i, M_i, R_i, K_i, X_i, V_i, T_i over insecure communication channel. Therefore, the proposed protocol is secure against impersonation 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 $D_i = b \oplus H(ID_i | P_i)$, $E_i = H(ID_i | P_i) \oplus P_i$, $F_i = A_i \oplus y_i$ and $G_i = B_i \oplus H(y_i) \oplus H(x)$ 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. *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 $Z_i = H^2(x) \oplus N_1$, $CID_i = A_i \oplus H(y_i) \oplus H(x) \oplus N_1$ and $M_i = H(H(x) \mid y_i \mid SID_k \mid N_1)$ during login phase corresponding to the legitimate user U_i. This malicious server S_k cannot compute ID_i, y_i and x from this information. The malicious server S_k cannot compute ID_i, y_i and x from $K_i = N_1 \oplus N_3 \oplus H(SK_k \mid N_2)$, $X_i = H(ID_i \mid y_i \mid N_1) \oplus H(N_1 \oplus N_2 \oplus N_3)$, $V_i = H[H(N_1 \oplus N_2 \oplus N_3) \mid H(ID_i \mid y_i \mid N_1)]$ and $T_i = N_2 \oplus N_3 \oplus H(y_i \mid ID_i \mid H(x) \mid N_1)$. Therefore, the proposed protocol is secure against malicious server attack.

5. *Malicious user attack:* A malicious privileged user U_i having his own smart card can gather information like $D_i = b \oplus H(ID_i \mid P_i)$, $E_i = H(ID_i \mid P_i) \oplus P_i$, $F_i = A_i \oplus y_i$ and $G_i = B_i \oplus H(y_i) \oplus H(x)$ 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 is smart card specific and the malicious user U_i requires to know the values of $H(x)$, y_m and A_m to masquerades as the legitimate user U_m. Therefore, this malicious user U_i has to guess y_m and A_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.
6. *Leak of verifier attack*: In this type of attack, the attacker may able to steal the verification table from the server. If the attacker steals the verification table from the server, he can use the stolen verifiers to impersonate a participant of the scheme. In the proposed protocol, the service provider server S_k knows SK_k and does not store any information in its database. Similarly the control server CS knows the value of x, stores $y_i \oplus x$ corresponding to C_i in its client's database, $SK_k \oplus H(x \mid SID_k)$ corresponding to server identity SID_k in its service provider server's database. The attacker cannot compute the values of x and y_i from the verifier information stored on the control server. In case verifier is stolen by breaking into smart card database, an attacker does not have sufficient information to calculate user's identity and password. Therefore, the proposed protocol is secure against leak of verifier attack.
7. *Offline dictionary attack*: In offline dictionary attack, the attacker can record messages and attempts to guess user U_i's identity ID_i and password P_i from recorded messages. An attacker first tries to obtains identity and password verification information such as $D_i = b \oplus H(ID_i \mid P_i)$, $E_i = H(ID_i \mid P_i) \oplus P_i$, $F_i = A_i \oplus y_i$ and $G_i = B_i \oplus H(y_i) \oplus H(x)$ 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. The probability of guessing two parameters correctly in the same attempt is nearly zero. Moreover, even if the attacker guesses one parameter correctly, he or she can not verify it with any password verifier information. Therefore, the proposed protocol is secure against offline dictionary attack.
8. *Online dictionary attack*: In this type of attack, the attacker pretends to be legitimate user 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 of the user U_i and then has to guess the identity ID_i and password P_i corresponding to the user U_i. Even after getting the valid smart card of user U_i by any mean, an attacker gets a very few chances (normally a maximum of 3) to guess the identity and password 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.
9. *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 $CID_i = A_i \oplus H(y_i) \oplus H(x) \oplus N_1$ 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 cannot link different sessions belonging to the same user.
10. *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 $SK = H(H(ID_i \mid y_i \mid N_1) \mid (N_1 \oplus N_2 \oplus N_3))$. The control server CS authenticates the user U_i using verifier information as $M_i^* = H(H(x) \mid y_i \mid SID_k \mid N_1)$, the service provider server S_k authenticates the server CS using $V_i^* = H[H(N_1 \oplus N_2 \oplus N_3) \mid H(ID_i \mid y_i \mid N_1)]$ and the user U_i authenticates the server S_k and the server CS using $V_i^* = H[H(N_1 \oplus N_2 \oplus N_3) \mid H(ID_i \mid y_i \mid N_1)]$. The proposed protocol satisfies strong mutual authentication.
11. *Denial of service attack*: In this type of attack, an attacker updates identity and password verification information on smart card to some arbitrary value and hence legitimate user can not login successfully in subsequent login request to the server. In the proposed protocol, smart card checks the validity of user U_i's identity ID_i and password P_i before password update procedure. An attacker can insert the stolen smart card of the user U_i into 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 $E_i^* = H(ID_i^* \mid P_i^*) \oplus P_i^*$ and compares it with the stored value of E_i in its memory to verify the legitimacy of the user U_i before smart card accepts password update request. It is not possible to guess 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 U_i. Therefore, the proposed protocol is secure against denial of service attack.
12. *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 (SID_k, Z_i, CID_i, M_i) but cannot compute the agreed session key $SK = H(H(ID_i \mid y_i \mid N_1) \mid (N_1 \oplus N_2 \oplus N_3))$ because an attacker does not know the values of ID_i, y_i, N_1, N_2 and N_3. Therefore, the proposed protocol is secure against parallel session attack.
13. *Man-in-the-middle attack*: In this type of attack, the attacker intercepts the messages sent between the client and the server and replay these intercepted messages. An attacker can act as client to server or vice-versa with recorded messages. In the proposed protocol, an attacker can intercept the login request message (SID_k, Z_i, CID_i, M_i) from the user U_i to the server S_k. Then he starts a new session with the server S_k by sending a login request by replaying the login request message (SID_k, Z_i, CID_i, M_i). An attacker can authenticate itself to the control server CS but cannot compute the session key $SK = H(H(ID_i \mid y_i \mid N_1) \mid (N_1 \oplus N_2 \oplus N_3))$ because an attacker does not know the values of ID_i, y_i, N_1, N_2 and N_3. Therefore, the proposed protocol is secure against man-in-the-middle attack.
14. *Message modification or insertion attack*: In this type of attack, the attacker modifies or inserts some messages on the communication channel with the hope of discovering the user's password or gaining unauthorized access. Modifying or inserting messages in proposed protocol can only cause authentication between the client and the server to fail but cannot allow the attacker to gain

any information about the user U_i's identity ID_i and password P_i or gain unauthorized access. Therefore, the proposed protocol is secure against message modification or insertion attack.

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 related 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 bit 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 bit. 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 $T_{SYM} > T_{EXP} > T_H$. In the proposed protocol, the parameters stored in the smart card are D_i, E_i, F_i, G_i and the memory needed (E1) in the smart card is 512 $(=4 \times 128)$ bits. The communication cost of authentication (E2) includes the number of communication parameters involved in the authentication protocol. The number of communication parameters are $\{SID_k, Z_i, CID_i, M_i, R_i, K_i, X_i, V_i, T_i\}$ and hence the communication cost of authentication (E2) is 1152 $(=9 \times 128)$ 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 $5T_H$. 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 $11T_H$. 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 $14T_H$.

The proposed protocol uses the control server CS and the service provider server S_k for user's authentication and still having less computation costs (E1, E2, E3) and nearly the same computation costs for (E4, E5) as compared to Hsiang and Shih's protocol as shown in Table 3. Moreover, 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 (Chang and Lee, 2004; Juang, 2004; Lin et al., 2003) and also highly secure as compared to these related protocols. The functionality comparison of the proposed protocol with the related smart card based authentication protocols is summarized in Table 4.

8. Conclusion

Smart card based password authentication is one of the most convenient ways to provide multi-factor authentication for the communication between a client and a server. User's privacy is an important issue in e-commerce applications. Dynamic identity based authentication protocols aim to provide privacy to the user's identity so that the users are anonymous in communication channels. Researchers have proposed different multi-server authentication protocols to eliminate main point of susceptibility of the single-server systems. We presented a cryptanalysis of a recently proposed Hsiang and Shih protocol and showed that their protocol is susceptible to replay attack, impersonation attack and stolen smart card attack. Moreover, the password change phase of Hsiang and Shih's protocol is incorrect. An improved protocol is proposed that inherits the merits of Hsiang and Shih's protocol and resists different possible attacks. We have specified and analyzed a secure 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.

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

	Proposed Protocol	Hsiang and Shih (2009)	Liao and Wang (2009)	Chang and Lee (2004)	Juang, (2004)	Lin et al. (2003)
E1	512 bits (0.5 $\|n\|$)	640 bits (0.625 $\|n\|$)	512 bits (0.5 $\|n\|$)	256 bits (0.25 $\|n\|$)	256 bits (0.25 $\|n\|$)	$(4t+1)$ $\|n\|$ bits
E2	9×128 bits (1.125 $\|n\|$)	14×128 bits (1.75 $\|n\|$)	7×128 bits (0.875 $\|n\|$)	5×128 bits (0.625 $\|n\|$)	9×128 bits (1.125 $\|n\|$)	7×1024 bits (7 $\|n\|$)
E3	$5T_H$	$6T_H$	$5T_H$	$2T_H$	T_H	$5tT_{EXP}$
E4	$11T_H$	$10T_H$	$9T_H$	$4T_H+3T_{SYM}$	$3T_H+3T_{SYM}$	$2T_{EXP}$
E5	$14T_H$	$13T_H$	$6T_H$	$4T_H+3T_{SYM}$	$4T_H+8T_{SYM}$	$7T_{EXP}$

Note: t : the number of servers; $|n|$ = 1024 bits.

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

	Proposed Protocol	Hsiang and Shih (2009)	Liao and Wang (2009)	Chang and Lee (2004)	Juang (2004)	Lin et al. (2003)
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	No	Yes	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
Replay attack	No	Yes	Yes	Yes	Yes	Yes
Impersonation attack	No	Yes	Yes	Yes	Yes	Yes
Stolen smart card attack	No	Yes	Yes	Yes	Yes	Yes

The proposed protocol is simple and fast if the user possesses valid smart card, knows correct identity and correct password for its authentication. The proposed protocol is practical and efficient because only one-way hash functions and XOR operations are used in its implementation. Security analysis proved that the proposed protocol is more secure and practical. Future scope in this work is to reduce the computational costs (E1, E2, E3, E4, E5) of authentication and to analyze this proposed work with some software tool like Java card to check out the real execution time required for the working of this protocol.

References

Brainard J, Juels A, Kaliski B, Szydlo M. A new two-server approach for authentication with short secrets. In: Proceedings of the USENIX security symposium, August 2003, p. 201–14.

Chang CC, Lee JS. An efficient and secure multi-server password authentication scheme using smart cards. In: Proceedings of the international conference on cyber worlds, November 2004, p. 417–22.

Chien HY, Chen CH. A remote authentication scheme preserving user anonymity. In: Proceedings of the advanced information networking and applications, 2, 2005. p. 245–48.

Das ML, Saxena A, Gulati VP. A dynamic id-based remote user authentication scheme. IEEE Transactions on Consumer Electronics 2004;50(2):629–31.

Ford W, Kaliski BS. Server-assisted generation of a strong secret from a password. In: Proceedings of IEEE 9th international workshop enabling technologies, June 2000, p. 176–80.

Hsiang HC, Shih WK. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards & Interface 2009;31(6):1118–23.

Hu L, Niu X, Yang Y. An efficient multi-server password authenticated key agreement scheme using smart cards. In: Proceedings of the international conference on multimedia and ubiquitous engineering (MUE'07), April 2007, p. 903–07.

Jablon DP. Password authentication using multiple servers. In: Proceedings of the RSA security conference, April 2001, p. 344–60.

Juang WS. Efficient multi-server password authenticated key agreement using smart cards. IEEE Transactions on Consumer Electronics 2004;50(1):251–5.

Kocher P, Jaffe J, Jun B. Differential power analysis. In: Proceedings of the CRYPTO 99, Springer-Verlag, August 1999, p. 388–97.

Lee WB, Chang CC. user identification and key distribution maintaining anonymity for distributed computer network. Computer System Science 2000;15(4): 211–4.

Li LH, Lin IC, Hwang MS. A remote password authentication scheme for multi-server architecture Using neural networks. IEEE Transactions on Neural Network 2001;12(6):1498–504.

Liao IE, Lee CC, Hwang MS. Security enhancement for a dynamic id-based remote user authentication scheme. In: Proceeding of the conference on next generation web services practice, July 2005, p. 437–40.

Liao YP, Wang SS. A secure dynamic id-based remote user authentication scheme for multi-server environment. Computer Standards & Interface 2009;31(1):24–9.

Lin IC, Hwang MS, Li LH. A new remote user authentication scheme for multi-server architecture. Future Generation Computer System 2003;19(1):13–22.

Liou YP, Lin J, Wang SS. A new dynamic id-based remote user authentication scheme using smart cards. In: Proceedings of 16th information security conference, Taiwan, July 2006, p. 198–205.

Mackenzie P, Shrimpton T, Jakobsson M. Threshold password-authenticated key exchange. Journal of Cryptology 2006;19(1):27–66.

Messerges TS, Dabbish EA, Sloan RH. Examining smart-card security under the threat of power analysis attacks. IEEE Transactions on Computers 2002;51(5): 541–52.

Raimondo MD, Gennaro R. Provably secure threshold password-authenticated key exchange. In: Proceedings of the advances in cryptology (Eurocrypt '03), May 2003, p. 507–23.

Shih HC. Cryptanalysis on two password authentication schemes, Laboratory of cryptography and information security, National Central University, Taiwan, July 2008.

Tsai JL. Efficient multi-server authentication scheme based on one-way hash function without verification table. Computers & Security 2008;27(3–4): 115–21.

Tsaur WJ, Wu CC, Lee WB. A smart card-based remote scheme for password authentication in multi-server internet services. Computer Standard & Interfaces 2004;27(1):39–51.

Yang Y, Deng RH, Bao F. A practical password-based two-server authentication and key exchange system. IEEE Transactions on Dependable and Secure Computing 2006;3(2):105–14.

Yang YJ, Bao F, Deng RH. A new architecture for authentication and key exchange using password for federated enterprises. In: Proceedings of 12th international federation for information processing information security conference (SEC '05), March 2005, p. 95–112.

Yoon EJ, Yoo KY, Improving the dynamic id-based remote mutual authentication scheme. In: Proceedings of the OTM workshops, LNCS 4277, July 2006, p. 499–507.