Advanced remote user authentication protocol for multi-server architecture based on ECC

Sheetal Kalra a,.*, Sandeep Sood b

aDept. of Computer Science & Engg., GNDU, Regional Campus, Jalandhar, India

bDept. of Computer Science & Engg., GNDU, Regional Campus, Gurdaspur, India

Keywords:
Authentication
Elliptic Curve Cryptography
Multi-server architecture
Smart card

Abstract

We have reached an era where desired web services are available over the networks by click of a button. In such a scenario, remote user authentication plays the most important role in identifying the legitimate users of a web service on the Internet. Researchers have proposed a number of password based authentication schemes which rely on single server for authentication. But, with tremendous advancements in technology, it is possible to engage multiple servers in authenticating their clients in order to achieve better security. In this paper, we propose an efficient password based authentication protocol for multi-server architecture. The protocol provides mutual authentication using smart card and is based on Elliptic Curve Cryptography, therefore offers best security at a low cost. In 2011, Sood et al. proposed a multi-server architecture protocol using smart cards. In this paper, we improve Sood et al. scheme by increasing its security and reducing the computation cost. The protocol is based on the concept of dynamic identity that uses a nonce based system and has no time synchronization problem.

1. Introduction

Remote user authentication is the process of identifying a legitimate user of a particular web service on the Internet. Due to their low cost, efficiency and portability, smart cards are widely used in e-commerce applications for remote user authentication. Smart card is a tamper proof plastic card almost the size of an ATM card with built in microprocessor and memory. The user of the smart card inserts his card in a card reader machine and enters his credentials such as identity and password. Based on this information, the authentication server and the smart card perform cryptographic operations to authenticate the user of the web service. A number of password based authentication schemes have been developed where only single server is involved in the authentication process. The authentication information stored on a single server becomes highly susceptible to various attacks such as leak of verifier, server spoofing and stolen verifier attack. Nowadays, ubiquitous computing has become very popular where multiple servers are involved in authenticating their users. Therefore, multi-server authentication schemes are required to cater to the needs of modern computing services. Over last few years, researchers have developed password based authentication schemes based on multi-server architecture (Ford and Kaliski, June 2000; Jablon, 2001; Lee and Chang, 2000; Lin et al., 2003; Brainard et al., 2003; Mackenzie et al., 2006; Juang, 2004; Chang & Lee, 2004; Yang et al., 2005; Hu et al., 2007; Tsai, 2008; Liao and Wang, 2009; Hsiang and Shih, 2009; Sood et al., 2011). Most of the proposed schemes are susceptible to one or more security attacks and involve high computation and communication cost. In this research paper, we propose an authentication scheme.

* Corresponding author.

E-mail addresses: kalra_8182@gmail.com, sheetal.kalra@gmail.com (S. Kalra), san1198@gmail.com (S. Sood).

2014 Elsevier Ltd. All rights reserved.

http://dx.doi.org/10.1016/j.jisa.2013.07.005
which is based on two-server-architecture paradigm. The authentication parameters of the user are distributed among two servers namely the control server and service provider server. The back-end control server is less exposed to the clients and therefore is more secure from various security attacks. The user directly communicates only with the service provider server which in turn communicates with the control server to authenticate the user of the web service.

Passwords are the most common and convenient way to authenticate the remote user of a web application. But users tend to use easy and common passwords for a number of web applications. Such passwords become a sensitive target for the hackers which lead to compromise of the password based authentication scheme. Another point of susceptibility is the static identity of the user where the user may change his password but he cannot change his identity. Such schemes leak out the partial information about the user which again leads to compromise of the scheme. On the other hand, in case of dynamic identity schemes the identity of the user changes with every login and even if the attacker launches a replay attack by recording various communication messages, he fails to login as a legitimate user. In this paper, we propose a dynamic identity based multi-server architecture protocol which is secure against various security attacks.

Cost and efficiency are another two factors on which the strength of any authentication scheme depends. Authentication schemes based on public key cryptography are very difficult to comprise because of the inherent strength of public key systems, but these schemes are very expensive as the use of public key cryptography involves calculation of exponential operations which needs a lot of processing time. Smart cards are small electronic cards with limited resources and therefore public key cryptography does not make an ideal choice for them. Symmetric cryptographic parameters are inexpensive in terms of computation cost but they are simpler to forge as compared to public key cryptographic parameters. In comparison to other public key systems (PKS), Elliptic Curve Cryptosystem provides maximum security per bit for a given key size (Borst et al., 2001). Smaller key size implies faster computation even with limited resources. Thus, ECC can very well be implemented in smart cards without increasing its computational resources and consequently its size and cost. Therefore, the most efficient way to implement security with smart cards is to use a password based authentication scheme based on ECC.

In this paper, we propose a protocol by improving the protocol proposed by Sood et al. (2011) which is based on multi-server architecture. We propose a remote user password authenticated protocol based on ECC for multi-server architecture. We have reduced the communication and computation cost considerably and also increased the security of the protocol. The major benefits of our protocol are 1) Asymmetric cryptographic primitives provide maximum security against brute force attacks as compared to symmetric primitives. ECC is based on the hard problem of ECDLP (Elliptic Curve Discrete Logarithm Problem) and there is no polynomial time algorithm available to solve it. Therefore, the use of ECC (public key cryptography) greatly enhances the security of any authentication scheme. Hence, our protocol based on Elliptic Curve Cryptography provides mutual authentication and session key agreement at a low computation cost. 2) The communication cost of our scheme is considerably low as compared to the schemes based on symmetric primitives and other schemes based on the public key cryptography. With a low communication cost and increased efficiency our protocol can very well be implemented for smart cards which are constrained devices with limited computational resources. 3) This protocol is a nonce based protocol and the identity of the user changes dynamically every time when the user is authenticated by the server. The user's identity is always hidden to the attacker in an insecure communication channel. This provides an extra level of security and prevents many well known attacks possible otherwise. 4) It is based on two-server-architecture and therefore offers much higher security than protocols based on single server architecture. 5) Our protocol provides multi-factor authentication where the legitimacy of a user is verified on multiple factors. The protocol authenticates the user on the basis of password, smart card and two-server-architecture. Therefore, the protocol is extremely robust and cannot be forged. 6) It achieves mutual authentication and session key agreement. 7) It is a dynamic identity based authentication scheme where the concept of dynamic identity is implemented using a nonce based system so there is no time synchronization problem. 8) The protocol is secure against all the well known attacks. 9) The password can be chosen freely by the user. 10) The password change phase of the protocol is much simpler and efficient in comparison to all the other protocols.

The paper is organized as follows. In Section 2 of the paper we discuss the related work in the area of authentication protocols. In Section 3 of the paper we review the protocol proposed by Sood et al. (2011). In Section 4 of the paper we propose our password based authentication scheme on multi-server architecture. In Section 5 we have done the security analysis of the protocol. Cost and functionality analysis is done in section 6 of the paper and Section 7 concludes the paper.

2. Related work

Over the last decade, a number of smart card based schemes have been developed to authenticate the users of web services over the Internet. Password based authentication schemes using smart card have gained a lot of popularity due to their convenience and ease of use. Most of the password based schemes developed by the researchers rely on single server for authentication. In 2004, a dynamic identity based remote user authentication was proposed by Das et al. (2004). User’s identity changes dynamically during each new session of the authentication process thereby protecting user’s anonymity. Das et al. (2004) claimed that their protocol is secure against replay attack, stolen verifier attack, guessing attack, forgery attack, identity theft and insider attack. Later, many weaknesses of their scheme were highlighted by researchers. Chien and Chen (2005) found that Das et al. protocol does not preserve user’s anonymity and proposed another protocol that preserves user’s anonymity. Liao et al. (2005) also improved the security of Das et al. scheme to achieve mutual authentication. Yoon and Yoo (2006) found that Liao et al. scheme
was susceptible to reflection attack. They proposed an improved scheme to remove the security flaws in Liao et al.‘s scheme. Sood (2011) proposed a password based authentication scheme using smart cards. All these schemes rely on single server architecture for their authentication process. A few schemes have been developed for authenticating users using smart cards which are based on Elliptic Curve Cryptography. Juang et al. (2008) proposed an efficient smart card based authentication scheme using smart cards which has low computation cost as compared to schemes based on other public key cryptographic techniques such as RSA. Yang and Chang (2009) proposed a mutual authentication with key agreement scheme for mobile devices based on ECC. Wang et al. (2011) proposed a robust authentication and key agreement scheme based on ECC and highlighted the strength of ECDLP (Elliptic Curve Discrete Logarithm Problem).

The first multi-server password based authentication scheme was proposed by Ford and Kaliaski (2000). The protocol splits the password information among multiple servers and therefore a malicious user cannot compromise the password by launching various attacks. The protocol uses public key systems to achieve authentication and therefore is computationally expensive. Jablon (2001) improved this protocol without the use of public keys. Lee and Chang (2000) also proposed a multi-server authentication protocol based on the problem of factorization and hash functions. Lin et al. (2003) proposed a multi-server authentication scheme that uses the ElGamal public key system. Though the scheme was very efficient as the server does not store verification information but it is computationally expensive due to the use of public key cryptosystems. Brainard et al. (2003) proposed a password based two-server authentication protocol in which only one server was exposed to the user. But this protocol again was based on public key cryptography and uses Secure Socket Layer (SSL) therefore becomes computationally intensive. In 2004, Tsauer et al. proposed a RSA cryptosystem based protocol for multi-server architecture without using a password verification table. Mackenzie et al. (2006) proposed a password based authenticated key exchange protocol that uses fixed number of servers with known public keys in which certain threshold number of servers participate in the authentication process of the server. The use of public key cryptography involves the calculation of exponential operations and hence is computationally intensive.

Juang (2004) proposed a multi-server authentication protocol using smart cards which was based on symmetric encryption algorithm. Chang and Lee (2000) improved Juang’s protocol for multi-server architecture in which no verification table is stored in the server. Yang et al. (2005) proposed a two-server based password authentication scheme. In their scheme, the backend control server is controlled by an enterprise headquarter and a frontend external server is operated by each affiliating organization. Hu et al. (2007) proposed an authentication protocol for multi-server architecture in which users can access multiple servers using smart cards. The server and the user communicate with each other using a session key. Tsai (2008) proposed a multi-server authentication protocol based on a nonce system and one-way hash function. Liao and Wang (2009) proposed a multi-server architecture authentication scheme using one-way hash functions. Hsiang and Shih (2009) found that Liao and Wang’s protocol was susceptible to server spoofing, insider attack, masquerade attack and registration server spoofing attack. Sood et al. (2011) pointed out that even Hsiang and Shih (2009) protocol was susceptible to replay attack, impersonation attack, stolen smart card attack and incorrect password change phase. In this paper, we propose an authentication protocol for multi-server environment based on ECC using smart cards. It is secure against all the security attacks and is computationally efficient.

3. Review of Sood’s protocol

Sood et al. (2011) proposed a dynamic identity based authentication protocol for smart cards based on multi-server architecture. The protocol is entirely based on hash function and XOR operation. It consists of four phases: registration phase, login phase, authentication and session key agreement phase and password change phase. The protocol consists of four phases.

1. Registration phase

The user selects a random number b and 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. The CS computes $F_i = A_i \oplus Y_i, G_i = B_i \oplus H(Y_i) \oplus H(X)$ and $C_i = A_i \oplus H(Y_i) \oplus X$. CS then stores $Y_i \oplus X$ in its database and issues a smart card containing security parameters $(F_i, G_i, H(0))$ to the user. User then computes security parameters $D_i = b \oplus H(ID_i(b), E_i = H(ID_i(P_i)) \oplus X$ and enters these parameters in smart card. All the service provider servers register themselves with CS on a unique secret key $SK_k$ with each service provider $S_k$. The server $S_k$ remembers the secret key $SK_k$ and CS stores secret key $SK_k$ as $SK_k \oplus H(X|SID_k)$ in its database.

2. Login phase

The user inserts his smart card into card reader machine and submits his identity $ID_i$, password $P_i$, and server identity $SID_k$. The smart card computes $E' = H(ID_i|P_i) \oplus P_i$ and compares with the stored value of $E_i$. Then 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 $M_i = H(H(X) \oplus Y_i|SID_k|N_1)$ and sends the login request message as $(SID_k, Z_i, CID_i, M_i)$ to $S_k$.

3. Authentication and session key agreement phase

After receiving the login request from the user $UID$, $S_k$ generates random nonce $N_2$ and 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 CS. CS extracts $SK_k$ from $SK_k \oplus H(X|SID_k)$ and 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 compares it to the stored value of $C_i$. Then CS extracts $Y_i$ from $Y_i \oplus X$ and computes $M'_i = H(H(X)|Y_i|SID_k|N_1)$ and compares with the received value of $M_i$. If they are not equal then CS rejects the login request otherwise CS generates random
nonce value $N_3$ and computes $K_i = N_i \oplus N_3 \oplus H(SK_k[N_3])$, $X_i = H(ID_i[Y_i][N_i]) \oplus H(N_i \oplus N_2 \oplus N_j), V_i = H[H(N_1 \oplus N_3 \oplus N_j)|H(ID_i[Y_i][N_i])], T_i = N_2 \oplus N_3 \oplus H(Y_i[ID_i|H(X)|N_i])$ and sends the message $(K_i, X_i, V_i, T_i)$ back to $S_k$. The server then computes $V_i' = H[H(N_1 \oplus N_3 \oplus N_j) | H(ID_i[Y_i][N_i])]$ and compares it with $V_i$. Then the server $S_k$ sends $(V_i, T_i)$ to smart card of the user $U_i$. Then smart card computes $N_2 \oplus N_3 - T_i \oplus H(Y_i | ID_i | H(X) | N_i), V_i' = H[H(N_1 \oplus N_2 \oplus N_3) | H(ID_i | Y_i | N_i)]$ and compares it with the received value of $V_i$. Finally, the user $U_i$‘s smart card, the server $S_k$ and the control server $CS$ agree on the common session key $SK = H(ID_i[Y_i][N_i]) (N_1 \oplus N_2 \oplus N_3)$. All the subsequent messages between the user $U_i$, the server $S_k$ and $CS$ are XORed with the session key.

4. Password change phase

The user $U_i$ inserts his smart card into a card reader and enters his identity $ID_i$ and password $P_i$. The smart card computes $E_i = H(ID_i | P_i) \oplus P_i$ and compares it with $E_i$. Once the authenticity of card holder is verified, smart card computes the values of $b_i, H(Y_i), H(X)$ and then prompts the card holder to resubmit a new password $P_{new}$. Then the value of $D_i, E_i$ and $G_i$ stored in the smart card are updated with $D_i^{new} = b \oplus H(ID_i | P_{new}), E_i^{new} = H(ID_i | P_{new}) \oplus P_{new}$ and $G_i^{new} = H(b \oplus P_{new}) \oplus H(Y_i) \oplus H(X)$ and the password gets changed.

3.1. Weakness of Sood’s protocol

The protocol proposed by Sood et al. (2011) is a dynamic identity based multi-service architecture scheme. The protocol has the following weaknesses:

- Insufficient information for session key agreement. In the registration phase of Sood et al.’s (2011) protocol, the user $U_i$ submits the parameters $A_i$ and $B_i$ and not the real identity $ID_i$, to the CS for registration and therefore CS does not store the user’s real identity in its database. Instead, the pseudo identification $CID_i = A_i \oplus H(Y_i) \oplus H(X) \oplus N_i$ is generated by user $U_i$‘s smart card for its authentication to the service providing server $S_k$ and the control server $CS$. Nowhere in the protocol the real identity information about the user is passed to th control server $CS$. Therefore, there is no way for the server $S_k$ and the control server $CS$ to get the real identity of the user $U_i$ throughout any phase of Sood et al.’s protocol. So the service provider $S_k$ and the control server $CS$ cannot compute and verify any verification information using the real identity of the user $U_i$ in authentication and session key agreement phase. Moreover, in step 4 of the authentication and session key agreement phase of Sood et al.’s (2011) protocol, the control server $CS$ computes the mutual authentication information $X_i = H(ID_i[Y_i][N_i]) \oplus H(N_1 \oplus N_2 \oplus N_j), V_i = H[H(N_1 \oplus N_3 \oplus N_j)|H(ID_i[Y_i][N_i])], T_i = N_2 \oplus N_3 \oplus H(Y_i | ID_i | H(X) | N_j)$. All these parameters use the identity $ID_i$ of user $U_i$ which is a contradiction. Therefore we found out that in Sood et al.’s (2011) protocol the authentication and session key agreement phase is incorrect.

- Use of symmetric parameters: In Sood et al.’s (2011) protocol all the security parameters $(Z_i = H^2(X) \oplus N_i, CID_i = A_i \oplus H(Y_i) \oplus H(X) \oplus N_i$ and $M_i = H(H(X)|Y_i|SID_k[N_i], R_i = N_2 \oplus SK_k)$ are computed using XOR operation and hash function. Hash functions are constructed using symmetric block ciphers. Symmetric cryptographic techniques involve low computation cost but are easier to forge in comparison to asymmetric cryptography. Asymmetric cryptography provides better security as it based on complex mathematical operations which are expensive in terms of communication cost. The authentication protocols based on the traditional asymmetric techniques of RSA and ELGMA are quite secure but do not make an ideal choice for the constrained environment of a smart card. A typical smart card available in the market today has a RAM that ranges between 256 bytes and 1 KB, EEPROM between 1 KB and 16 KB, ROM between 6 KB and 24 KB. It has an inbuilt microprocessor typically clocked at a speed of mere 5 MHz. Any further increase in the processing capacity will increase the size as well as the cost of a smart card. Thus, the authentication protocols designed for smart cards must provide high efficiency at minimum computation cost. ECC which is a form of asymmetric cryptography is most suitable for implementing security protocols for constrained devices like smart cards, cell phones etc. In comparison to other asymmetric cryptography, Elliptic Curve Cryptosystems provide maximum security per bit for a given key size. For the same level of security, the length of key required in ECC is much smaller than in other PKS. For e.g. a key size of 166 bits in ECC and 1024 bits in RSA provides equivalent security. Smaller key size implies faster computation even with limited resources. Table 2 shows the comparison of key sizes among various cryptographic techniques. Thus, ECC can very well be implemented in smart cards without increasing its computational resources and consequently its size and cost. In our protocol, the security parameters are computed using ECC which is highly secure and has a low computational cost. The protocol is robust and computationally inexpensive.

- High Communication Cost: Communication cost is the most important factor which determines the efficiency of a security protocol. The communication cost of Sood et al.’s (2011) protocol is higher compared to others as it involves exchanges of more messages.

<table>
<thead>
<tr>
<th>Table 1 – Notations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_i$</td>
</tr>
<tr>
<td>$S_k$</td>
</tr>
<tr>
<td>$CS$</td>
</tr>
<tr>
<td>$ID_i$</td>
</tr>
<tr>
<td>$SID_k$</td>
</tr>
<tr>
<td>$P_i$</td>
</tr>
<tr>
<td>$X$</td>
</tr>
<tr>
<td>$Y_i$</td>
</tr>
<tr>
<td>$N_i$</td>
</tr>
<tr>
<td>$N_1$</td>
</tr>
<tr>
<td>$N_2$</td>
</tr>
<tr>
<td>$N_3$</td>
</tr>
<tr>
<td>$E_p$</td>
</tr>
<tr>
<td>$P$</td>
</tr>
<tr>
<td>$Z_p$</td>
</tr>
<tr>
<td>$H(.)$</td>
</tr>
<tr>
<td>$\oplus$</td>
</tr>
<tr>
<td>$\mid$</td>
</tr>
</tbody>
</table>
et al.’s (2011) protocol as depicted in Table 3 (E1, E3, E4, E5) is quite high. The protocol proposed in this paper is based on ECC and therefore the communication cost of our scheme is considerably less as compared to the schemes based on symmetric primitives and other schemes based on the public key cryptography. Tables 3 and 4 depict the cost and functionality comparison respectively among the related schemes and proposed protocol. Mathematically, while calculating an ECC point the basic operations involved are addition and multiplication over finite field numbers. The calculation of ECC points does not involve any exponential operations. With a low communication cost and increased efficiency our protocol can very well be implemented for smart cards which are constrained devices with limited computational resources.

4. Proposed protocol

In this section, we propose an ECC based protocol for multi-server architecture using smart cards. The protocol is secure against all well known security attacks. The user Ui logs in to the server by inserting his smart card in the card reader machine by submitting his IDi and password Pi. The notations used in the protocol are listed in Table 1. The protocol consists of 5 phases.

1. Registration phase

In order to register with the control server CS, the user Ui submits his identity IDi via a secure communication channel to the control server CS.

\[ \text{Step 1.} \quad U \rightarrow \text{CS:IDi} \]

The control server CS validates the identity of the client and computes the security parameter \( F_i = H(\text{ID}_i, Y_i) \times G \), by employing the master key X of the server. Then the control server updates the database and stores IDi and Yi corresponding to the identity of the user Ui. Then the control server issues a smart card to the user which stores the value of \( F_i \).

\[ \text{Step 2.} \quad \text{CS} \rightarrow U_i; \text{Smart Card (F}_i, H(\cdot)) \]

The user Ui on receiving the smart card activates his card by entering his password Pi. Smart card calculates the security parameter.

2. Precomputation phase

Before the system begins the smart card calculates an ECC point and stores it in its memory for further communication.

3. Login phase

The user Ui in order to login with the service provider server Sk inserts his smart card into a card reader machine and submits his IDi, password Pi and the identity SIDk of service provider Sk. The authenticity of the user is verified by the smart card which then sends the verification and login information to the destination server Sk.

4. Authentication and session key agreement phase

The verification information of user and server Sk is passed to the control server CS by the service provider server Sk. The server Sk and the control server CS mutually authenticate each other and the user Ui. Once authenticated, the user Ui, service provider server Sk and control server CS agree on a common session key for further communication.

5. Password change phase

The user can freely change his password without the interference of the control server CS.

Before the system begins, the control server CS selects a large prime number \( p \) and two integer elements \( a \) and \( b \) where \( p \) is of high order such that \( p > 2^{380} \) and \( a \) and \( b \) satisfy the equation \( 4a^3 + 27b^2 \mod p \neq 0 \). Then the server selects an elliptic curve equation \( E_p \) over the finite field \( p \): \( y^2 = x^3 + ax + b \mod p \). The server selects a generator point \( G \) of order \( n \), where \( n \) is a large divisor such that \( n \times G = O \). The server also selects \( X \) as its private key and publishes \( (E_p, G, n, p) \).

4.1. Registration phase

In order to register with the control server CS, the user Ui sends his identity IDi through a secure communication channel to the control server CS.

\[ \text{Step 1.} \quad U \rightarrow \text{CS:IDi} \]

The control server CS validates the identity of the client and computes the security parameter \( F_i = H(\text{ID}_i, Y_i) \times G \), by employing the master key X of the server. Then the control server updates the database and stores IDi and Yi corresponding to the identity of the user Ui. Then the control server issues a smart card to the user which stores the value of \( F_i \).

\[ \text{Step 2.} \quad \text{CS} \rightarrow U_i; \text{Smart Card (F}_i, H(\cdot)) \]

The user Ui on receiving the smart card activates his card by entering his password Pi. Smart card calculates the security parameter.
parameter $E_i = H(ID_i,P_i) \oplus P_i \oplus N_i$ where $N_i$ is a random number generated by the smart card for the user $U_i$. The smart card now stores $(F_i, E_i, H())$.

**Step 3.** $U_i \rightarrow$ Smart Card:$P_i$ and Smart Card stores $(F_i, E_i, H())$

All the service provider servers $S_k$ register themselves with the control server $CS$ and agree on a unique secret key for communication $SK_k$ corresponding to server $CS$ and agree on a unique secret key for communication $SK_k$ corresponding to service provider server $S_k$ and submits his identity $ID_i^*$, password $P_i^*$ and SID$_k$ which is the identity of the service provider server $S_k$ with which the user wants to communicate. The smart card now stores $(F_i, E_i, H())$.

**Step 4.** $S_k \rightarrow CS$: SID$_k$, SK$_k$

The control server computes the security parameter $M_k = SK_k \oplus H(X \mid SID_k) \times G$ and stores it on the service provider’s database. The control server stores $SK_k \oplus H(X \mid SID_k)$ corresponding to SID$_k$ in its database.

**Step 5.** $CS \rightarrow S_k$: $M_k$ and Control server stores $(SK_k \oplus H(X \mid SID_k), SID_k)$

Different phases of the protocol and their communication parameters are shown in Fig. 1.

### 4.2 Precomputation phase

The smart card chooses a random number $N_i$ and computes ECC point $P_1 = N_i \times G$. Then it stores $P_1$ in its memory as a communication parameter which is used in the process of authentication.

### 4.3 Login phase

The user $U_i$ inserts his smart card into a card reader to login to a service provider server $S_k$ and submits his identity $ID_i^*$, password $P_i^*$ and SID$_k$ which is the identity of the service provider server $S_k$ with which the user wants to communicate. The smart card computes $E_i^* = H(ID_i^*, P_i^*) \oplus P_i^* \oplus N$ and compares it with the stored value of $E_i$ in its memory. This verifies the legitimacy of the user $U_i$.

**Step 1.** Smart Card checks $E_i^* = E_i$

After verification, the smart card calculates the ECC point $P_{11} = N_1 \times P_1 = N_1 \times (H(X \mid ID_i) \times G)$ sends the ECC points $[P_1, P_{11}]$ along with the identity $ID_i$ of the user and SID$_k$; identity of the service provider $S_k$ as login request to $k$th service provider $S_k$.

**Step 2.** Smart Card $\rightarrow S_k$: $P_1$, $P_{11}$, $ID_i$, SID$_k$

### 4.4 Authentication and session key agreement phase

After receiving the login request from the user $U_i$, the service provider server $S_k$ selects a random nonce $N_i$ and calculates an ECC point $P_2 = N_2 \times G$ and $P_{22} = N_2 \times M_k$ and sends the login request to the control server $CS$ $[P_1, P_{11}, P_{22}, ID_i, SID_k]$ along with the identity of the user $ID_i$ and service provider server $SID_i$.

**Step 1.** $S_k \rightarrow CS$: $P_1$, $P_{11}$, $P_2$, $P_{22}$, $ID_i$, SID$_k$

CS extracts $Y_i$ from $Y_i \oplus X$ corresponding to $ID_i$ in its database. Then the CS employs its private master key $X$ and computes $H(X \mid ID_i) \times Y_i$ and calculates point $P_{11} = P_1 \times H(X \mid ID_i) \times Y_i = (N_1 \times H(X \mid ID_i) \times G)$ and compares the value of $P_{11}$ and $P_{11}$ to verify the authenticity of the smart card. The equivalency confirms the legitimacy of the smart card otherwise the session is terminated.

Then the control server computes $P_{22} = P_2 \times SK_k \oplus H(X \mid SID_k)$ and compares the value of $P_{22}$ to the received value of $P_{22}$ to verify the authenticity of the service provider server $S_k$. The equivalency confirms the legitimacy of the service provider server $S_k$ otherwise the session is terminated.

**Step 2.** Control Server checks $P_{11}^* = P_{11}$ and $P_{22}^* = P_{22}$

If they are not equal, the control server rejects the login request and terminates the session. Otherwise, the CS extracts $SK_k$ from $SK_k \oplus H(X \mid SID_k)$ corresponding to SID$_k$ in its database. Then the control server CS selects a random number $N_3$ and calculates two ECC points $P_3 = N_3 \times G$ and $P_{23} = N_3 \times P_3$ and sends the $(P_3, D_1 = SK_k \times P_3)$ to the service provider server $S_k$. The service provider server $S_k$ computes...
the value of $D'_0 = SK_k \times N_2 \times P_3$ and compares the value of $D'_0$ with the received value of $D_i$ to verify the legitimacy of CS. The equivalency confirms the legitimacy of the control server CS.

Step 3. Server $S_k$ checks $D'_0 = D_i$.

If they are not equal, the session is terminated.

Then the server $S_k$ computes $T'_i = N_2 \times P_1$ and sends $(P_2, T'_i)$ to the smart card of user $U_i$. Upon receiving the security parameters the smart card computes $T'_0 = N_1 \times P_1$ and compares the value of $T'_0$ to the received value of $T_i$. This authenticates the service provider server $S_k$.

Step 4. Smart card checks $T'_0 = T_i$.

The equivalency of the security parameters authenticates the legitimacy of the control server CS, the service provider server $S_k$ and the smart card of the legitimate user $U_i$. Finally, the control server CS, the user $U_i$'s smart card and the service provider server $S_k$ agree on a common session key $SK = H(H(ID_i) || N_1) \oplus (N_1 \oplus N_2 \oplus N_3)$. The subsequent messages between CS, server $S_k$ and user $U_i$ are XOR'd with the session key. Using this session key the user $U_i$, server $S_k$ and CS may retrieve the original message as all of them know the common session key.

4.5. Password change phase

In the proposed protocol, the user can change his password without the intervention of the control server CS. The legitimate user $U_i$ inserts his smart card into the card reader machine and submits his identity $ID_i$ and password $P_i$. The smart card computes $E_i = H(ID_i || P_i) \oplus P_i \oplus N'$ and compares it with the stored value of $E_i$ in its memory. This verifies the legitimacy of the user $U_i$. Once the authenticity of card holder is verified, the smart card asks the user to enter his new password $P_{new}$ and updates the value of $E_i$ stored in the smart card to $E_{new} = H(ID_i || P_{new}) \oplus P_{new} \oplus N_{new}$. So the password gets changed by using a simple password change phase.

5. Security analysis

Our protocol achieves mutual authentication between the user and both the servers (control server and service provider server) and is immune to all the security attacks such as man-in-the-middle attack, replay attack, malicious server and offline dictionary attack. Kocher et al. (1999) and Messerges et al. (2002) pointed out that the information stored on smart cards can be extracted using some reverse engineering techniques such as power consumption. This means that once the smart card is stolen, the information stored on it can be used by the malicious user for his personal benefit. The authentication protocols proposed by the researchers are susceptible to one or more security attacks, once the smart card is stolen by a malicious user. But, in our proposed protocol, the password verification information is split between the control server and service provider server and therefore the protocol is robust and prevents all the security attacks. In this section, we have done a detailed security analysis of our protocol against the various security threats.

1. Impersonation attack

In this type of attack, the malicious user forges the security parameters from the authentication protocol and tries to
impersonate as a legitimate user. In our protocol, the malicious user has to guess the parameters $H(X)$, $Y_i$ and the nonce $N_i$ which is used in the calculation of ECC points $P_i$ and $P_{i1}$ for generating a valid login request. The Elliptic curve discrete logarithm problem is impossible to solve in real polynomial time, thus the ECC points cannot be forged. Moreover, it is not possible to guess all the security parameters viz. $H(X)$, $Y_i$ and $N_i$ from the communication parameters $[P_{i1}, P_{22}, P_{23}, D_i, T_i]$ sent over the communication channel. Therefore, the malicious user will fail to launch an impersonation attack on our proposed protocol.

2. Malicious server attack

In this type of attack, malicious servers $S_k$ may act as a legitimate server $S_k$ by monitoring and gathering the information related to the user $U_i$ during the login process. The malicious server $S_k$ can gather information $P_{i1}$, $P_{22}$ and $ID_i$ of the legitimate user $U_i$. But the attacker cannot compute the value of $H(X)$, $Y_i$ as ECC point $P_2$ cannot be forged. Therefore, the proposed protocol is secure against malicious server attack.

3. Replay attack

In this type of attack, the malicious user gathers the communication messages exchanged between the user $U_i$ and server $S_k$ and then tries to replay the same messages acting as a legitimate user. But replaying the same messages is useless in our proposed protocol as the smart card, service provider server $S_k$ and control server $CS$ select random nonce values $(N_1, N_2, N_3)$ in each new session. Session key of the protocol is based on the random nonce values of $(N_1, N_2, N_3)$ and hence cannot be computed by the malicious user. Therefore, the proposed protocol is secure against replay attack.

4. Stolen smart card attack

Using reverse engineering techniques, a malicious user may try to gather information stored on the smart card if he gets the smart card of a legitimate user $U_i$. In our protocol the smart card contains the parameters $F_i$ and $E_i$. Even after gathering this information the malicious user has to guess the value of $H(X)$, $Y_i$, $P_i$, and random number $N$ simultaneously, which is not possible in real polynomial time. Therefore the protocol is secure against stolen smart card attack.

5. Leak of verifier attack

In this type of attack the malicious user may be able to forge the information stored on the control server $CS$ or any of the service provider servers $S_k$ and may use this information in order to impersonate as a legitimate user. In our protocol the service provider server stores $SK_k$ and does not store any other information in its database. The control server stores $Y_1 @ X$ corresponding to $ID_0$ in its database, $SK_k @ H(X | SID_k)$ corresponding to service provider server identity $SID_k$ in its service provider server’s database. Now, it is impossible for the malicious user to compute the value of $X$ and $Y_1$ using this information. Therefore our protocol is secure against this type of attack.

6. Online dictionary attack

This is most popular type of attack launched by a malicious user once he is able to get a valid smart card of a legitimate user $U_i$. He may try to guess the password $P_i$ and $ID_i$ randomly, but the smart card gets locked after certain number of invalid attempts (maximum of three attempts). It is impossible for the malicious user to guess the values of $P_i$ and $ID_i$ simultaneously in real polynomial time. Hence our protocol is secure against online dictionary attack.

7. Offline dictionary attack

In this type of attack, the malicious user listens to the communication channel and records the communication messages and then tries to guess the legitimate user’s password $P_i$. In our protocol, the malicious user tries to get the password security information $E_i = H(ID_i | P_i) @ P_i @ N_i$, $F_i = H(X | ID_i | Y_i) @ G$, and nonce value $N_i$ which is used in the calculation of ECC point $P_1$ and $P_{i1}$ and then tries to guess the password $P_i$ by offline guessing. In our protocol, it is impossible to forge the ECC points $P_1$ and $P_{i1}$. In addition to this it is not possible to guess more than one parameter from the security parameters in real polynomial time. Hence the protocol is secure against offline dictionary attack.

8. Mutual authentication

Three entities that are participating in our protocol; smart card of the legitimate user $U_i$, service provider server $S_k$ and control server $CS$ mutually authenticate each other before agreeing upon a common session key. The common session key $SK = H(H(ID_i | Y_i | N_i) | (N_1 @ N_2 @ N_3))$ derived by contribution of nonce value $(N_1, N_2, N_3)$ from all the three entities. The control server $CS$ authenticates the user $U_i$ and the service provider server $S_k$ by calculating the ECC point $P_{i1}$ and $P_{22}$ and comparing it with the received value of $P_{i1}$ and $P_{22}$ respectively. The service provider server $S_k$ validates the control server by security parameter $D_i$ and smart card validates the service provider server by security parameter $T_i$. Hence the proposed protocol achieves mutual authentication.

9. Denial of service attack

In this type of attack, the malicious user prevents the legitimate user from login by updating the password verification information stored on the smart card to some arbitrary value. But in our protocol even if the malicious user steals the smart card of the legitimate user, he has to undergo the password change phase before making any update to the verification information. He has to guess the $ID'_i$ and $P'_i$ simultaneously as the smart card would compute $E'_i = H(ID'_i | P'_i) @ P'_i @ N'_i$ and match with the stored value of $E_i$ on the smart card before undergoing the password change phase. But in real polynomial time it is not possible to guess $ID'_i$, $P_i$, $N_i$ simultaneously. Hence the proposed protocol is secure against denial of service attack.
10. Man-in-the-middle attack

In this of the malicious user listens to the communication channel and intercepts the messages between the server and the user Ui and relay these messages back. The malicious user may act as a server to the user or it may act as a legitimate user to the server. In our protocol the malicious user may intercept the communication messages but he will never be able to compute the session key SK = H(H(IDiYiNi) | (Ni ⊕ Ni ⊕ Ni)) as it is based on Yi and random nonce values (Ni, Ni, Ni) which is chosen fresh for each new session. Hence the protocol is secure against man-in-the-middle attack.

11. Parallel session attack

In this type of attack, the malicious user initiates a parallel session with the legitimate user Ui by listing to communication messages and resending the same messages to the server within the valid time frame window. In our protocol he may masquerade as a legitimate user by resending the login request [P1, P2, ID, SID] but cannot compute the session key SK = H(H(ID)| Yi) Ni as it is based on Yi and random nonce values (Ni, Ni, Ni) which is chosen fresh for each new session. Hence the protocol is secure against parallel session attack.

6. Cost and functionality analysis

An efficient authentication protocol must consider computation and communication cost while authenticating a remote user. Elliptic Curve Cryptography is a public key cryptography that provides maximum strength per bit in terms of security. For the same level of security, the length of cryptographic keys in ECC is comparatively much smaller than any other public key systems. Table 2 shows the comparison of key sizes among various cryptographic techniques. Moreover, our protocol uses only XOR operations and one-way hash functions, both of which are very inexpensive operations in cryptography. Our protocol is very secure and efficient as it is based on random nonce values and has no time synchronization problem as the protocol does not use timestamps. The remote user authentication protocols based on timestamps are subjected to time synchronization problems if the server’s and client’s clock is not synchronized.

While calculating the cost of the protocol, the identity IDi, password Pi, X, Yi and nonce values (Ni, Ni, Ni) all are assumed to be 128 bits long. Also, the output of the one way hash function is 128 bits and elliptic curve cryptosystem is ECC – 224 bits. Let TH, TE, TECM, Ts be the time for one hashing operation, one exponential operation and time for one multiplication of a number over elliptic curve, time to carry out symmetric encryption/decryption respectively. The comparison of the time complexity associated with these operations can be expressed as Ts > TE > TECM > TH. The time taken to perform an exponential operation is much more (approx. 8 times) than the time taken to perform one elliptic point multiplication (Juang et al., 2008). Let (E1) be the memory needed in smart card to store the security parameters. In the proposed protocol, the parameters stored in the smart card are Ei and Fi is (128 + 224 = 352) bits Let (E2) be the cost of communication parameters involved in the authentication process. In the proposed protocol the communication parameters are [P1, P11, P2, P22, P3, Dk, Ti, Mk, IDi, SID] = [8 × 224 + 2 × 128 (bits)]. Let (E3) be cost of registration which includes the total time of all operations executed by the user Ui in the registration phase. In the proposed protocol the cost of registration is (2TH + 1TECM). Let (E4) cost of be the time spent that user for computation during the authentication process. In the proposed protocol this time is (1TH + 3TECM). Let (E5) be the computation cost of the service provider server and control server during the process of authentication. In the proposed protocol the total time spent by the both the servers is (2TH + 9TECM). Tables 2–4 show the comparison of key sizes among different cryptographic techniques, cost comparisons among related smart card based multi-server authentication protocols, functionality comparison among smart card based multi-server authentication protocols respectively.

7. Conclusion

Modern network technologies have geared up the use of e-commerce applications tremendously. A lot of sensitive information travels over the networks nowadays. In such a scenario, user authentication becomes vital for cooperate networks to flourish. Password based authentication schemes using a smart card make an ideal choice for e-commerce applications over cooperate networks as they provide multifactor authentication between the user and server. Researchers have proposed different multi-server authentication protocols to remove the main point of susceptibility of a single authentication server. We have proposed an efficient multi server authentication protocol using smart cards based on Elliptic Curve Cryptography (ECC). The use of ECC provides all the benefit of using an asymmetric cryptosystem even for a constrained environment of a typical smart card. With a low computational and communication cost it prevents all well known attacks by the malicious users of the network.

REFERENCES


