Remote User Authentication Protocol for Multi-Server Architecture Based on ECC

Vishal Chaugule¹ Ravindrakumar Deshmukh² Shreyas Deshpande³ Prof. Bhaté D.V⁴ Prof. Sonawane K. P⁵

¹,²,³,⁴,⁵Department of Computer Engineering
Dnyanganga College of Engineering and Research, Pune, India

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 multiserver architecture. The protocol provides mutual authentication using user system 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 user systems. 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.

Key words: ECC, CS, Elliptic Curve Cryptography

I. 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, user systems are widely used in e-commerce applications for remote user authentication process. The user of the user system firstly enters his credentials such as identity and password. Based on this information, the authentication server and the user system 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, 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 multiple server i.e. (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 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. The system is proposed as follows in the following ways:

II. RELATED WORK

The first multi-server password based authentication scheme was proposed by Ford and Kaliski (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.

III. PROPOSED METHODOLOGY

In this system, we proposed following phases:

1) Registration Phase
2) Precomputation Phase
3) Login Phase
4) Authentication Phase
5) Password Change Phase

A. Registration Phase:
User system submits his identity Idi , password Pi and server identity SIDk then CS validates,
\[ F_i = H(X | Idi | Yi) \times G \]
Where, \( F_i = \) Security Parameter
\( X = \) private key of server based on ECC
\( Idi = \) User identity
\( Yi = \) server secret no. for user Ui
\( H() = \) hash function
\( G = \) generator point
After generating security parameter US generates,
\[ E_i = H(\text{Id}_i | P_i) \oplus P_i \oplus N_i \]
Where, \( E_i = \text{Security parameter} \)
\( P_i = \text{User password} \)
\( N_i = \text{random no. generated by US} \)
Then US stores \((F_i, E_i)\)
CS computes,
\[ M_k = S_k \oplus H(X | S_{idk}) \times G \]
Where, \( M_k = \text{Security parameter} \)
\( S_k = \text{Kth service provider server} \)
\( S_{idk} = \text{Unique identification of Kth SPS} \)
CS stores, \((M_k, S_{idk})\)

### B. Precomputation Phase:
The user system selects a random number \(N_1\) and computes ECC point \(P_1 = N_1 \cdot G\). Then it stores \(P_1\) in its memory as a communication parameter which is used in the process of authentication.

### C. Login Phase:
The user \(U_i\) in order to login with the service provider server \(S_k\) inserts his smart card into a card reader machine and submits his ID, password \(P_i\) and the identity \(S_{idk}\) of service provider \(S_k\). The authenticity of the user is verified by the smart card which then sends the verification and login information to the destination server \(S_k\).

US computes,
\[ E_i^* = H(\text{Id}_i* | P_i*) \oplus P_i* \oplus N_i \]
Then checks \(E_i^* = E_i\)
US calculates,
\[ P_{11} = N_1 \times F_i \]
Where, \(P_{11} = \text{ECC} \)
\( N_1 = \text{Nonce generated by US} \)
Sends \([P_{11}, P_{11}]\) to \(CS\)

### D. Authentication and Session Key Agreement Phase:
The verification information of user and server \(S_k\) is passed to the control server \(CS\) by the service provider server \(S_k\). The server \(S_k\) and the control server \(CS\) mutually authenticate each other and the user \(U_i\). Once authenticated, the user \(U_i\), service provider server \(S_k\) and control server \(CS\) agree on a common session key for further communication.

SP calculates,
\[ P_2 = N_2 \times G \]
\[ P_{22} = N_2 \times M_k \]
Where, \(P_2 \& P_{22} = \text{ECC point}\)
\(N_2 = \text{Nonce generated by SPS}\)
\(M_k = \text{security parameter computed by CS}\)
CS extracts \(Y_i\) from \(Y_i \oplus X\).
Calculates point,
\[ P_{11}' = P_1 \times H(X | Id_i \mid Y_i) \]

### E. 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 \(p > 2160\) and \(a\) and \(b\) satisfy the equation \((4a^3 + 27b^2) \mod p = 0\). Then the server selects an elliptic curve equation \(E_p\) over the finite field \(p: y^2 \equiv (x^3 + ax + b) \mod p\). The server selects a generator point \(G\) of order \(n\), where \(n\) is a large divisor such that \(n \equiv G \not\equiv 0\). The server also selects \(X\) as its private key and publishes \((E_p, G, n, p)\).

US computes,
\[ E_i^* = H(\text{Id}_i* | P_i*) \oplus P_i* \oplus N_i \]

### Table 1:

<table>
<thead>
<tr>
<th>User i</th>
<th>User i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sk</td>
<td>Kth service provider server</td>
</tr>
<tr>
<td>CS</td>
<td>Control server</td>
</tr>
<tr>
<td>Idi</td>
<td>Unique Identification of User Ui</td>
</tr>
<tr>
<td>SIDk</td>
<td>Unique Identity of kth service provider server</td>
</tr>
<tr>
<td>Pi</td>
<td>Password of User Ui</td>
</tr>
<tr>
<td>X</td>
<td>Private key of Server based on ECC</td>
</tr>
<tr>
<td>Yi</td>
<td>Server’s secret number for user Ui</td>
</tr>
<tr>
<td>N_i</td>
<td>Random number generated by smart card for user Ui</td>
</tr>
<tr>
<td>N_1</td>
<td>Nonce generated by smart card</td>
</tr>
<tr>
<td>N_2</td>
<td>Nonce generated by server provider server</td>
</tr>
<tr>
<td>N_3</td>
<td>Nonce generated by control server</td>
</tr>
<tr>
<td>E_p</td>
<td>Elliptic Curve equation over (Z_p)</td>
</tr>
<tr>
<td>P</td>
<td>Large prime number</td>
</tr>
<tr>
<td>Z_p</td>
<td>Algebraic Group over (p)</td>
</tr>
</tbody>
</table>

### IV. H/W, S/W REQUIREMENTS
Following are the hardware and software requirements for our system,
1) RAM 256 MB
2) 1 GB HDD
3) Apache Tomcat Server,
4) Internet Browser.
5) Ellipse Indigo
6) SQL
7) Java

### V. COST 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 1 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 in not synchronized. While calculating the cost of the protocol, the identity IDi, password Pi, X, Yi and nonce values (N1, N2, N3) 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 e 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. 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 (352) bits Let (E2) be the cost of communication parameters involved in the authentication process.

VI. Conclusion
Password based authentication schemes make an ideal choice for e-commerce applications over cooperate networks as they provide multifactor authentication between the user and server. With a low computational and communication cost it prevents all well-known attacks by the malicious users of the network. 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 crypto system 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