AUTHENTICATION SCHEME FOR CLIENT AND SERVER USING ELLIPTIC CURVE CRYPTOGRAPHY

M.Lavanya², G.Pra vein kumar¹, N.Lena Murugan¹, M.Vigneshwaran¹S.Saravanan²
¹B.Tech Information Communication Technology, School of Computing.
²Assistant Professor, School of Computing, SASTRA University, Thanjavur, Tamilnadu, India.
Email: m_lavanyass@ict.sastra.edu

Received on 20-08-2016 Accepted on 25-09-2016

Abstract

There is a phenomenal growth in data transfer and sharing. In a client server environment transmitting the data over the communication medium tends to be a challenging task. Existing studies concentrates more on increasing the security of data in storing and retrieving the data. They focus more on improving the cryptographic technique and mutual authentication schemes between the server and client. The importance given to the client’s processing powers and resource availability is comparatively less. In this paper we enhanced a mutual authentication system using Elliptic curve cryptography by employing a hash function before transmitting. The public key cryptography provides good level of security but always gives a variable length of output. Transmission of variable length data is susceptible to computational errors. We provide a fixed length of data for transmission using MD-5 hashing algorithm as it provides optimum amount of security and requires optimal resources.

Keywords: Elliptic Curve Cryptography, Server and Client, Session key, Hash Function.

I. Introduction

Increase in application processing Efficiency with reduced costs and gaining maximum benefit from all resources working together make the approach of client server environment a reliable one. The benefits are achieved by dividing the tasks and processes between the client machine and server machine. Each task works independently. This also ensures the cooperation and compatibility with other counterpart. If the connection from client to server or connection from server to client is not secured or left open, it provides an opening for intruders to use the credentials in a negative way. This is where cryptography comes into play. We select a elliptic curve or a plane curve over a finite field that has points which will satisfy the equation

$$y^2 = x^3 + ax + b,$$
Elliptic Curve Cryptography (ECC) is a cryptographic technique which provides relatively smaller length keys and higher order of security on comparing with other algorithms. It is a public curve cryptography approach based on the elliptic curves’ algebraic structure over finite fields. Mutual authentication or two-way authentication, is a technology or process in which in a communications link, both the entities authenticate each other before transmission. In a network environment, the server authenticates the client and vice-versa. A cryptographic hash function is considered impossible to invert practically that is, to get back the input data from its hash value alone. These one-way hash functions have been the workhorses of modern cryptography. Hash functions are used in the process of generation of the session key which is agreed by both client and server after the process of mutual authentication between them.

II. Existing System

Various concepts were proposed in this domain. In paper [1] Hyper Text Transfer Protocol [HTTP] is used to establish a secured ECC connection between the embedded systems and the cloud environment. The connection is verified using AVISPA tool which confirms the algorithm in the presence of an intruder. Advantages of using ECC algorithm [2], ECDLP (Elliptic curve discrete logarithm problem) and it’s pollard's rho algorithm were discussed. In paper [3] it focuses on the server authentication based SIP [Session Initiation Protocol], which in turn relies on HTTP for security purposes. E-payments [4] shows how mobile can be unsafe and how to share the key between the sender and the receiver alone. In paper [5] it proposes three algorithms to make a secure connection between the client and the server. Algorithm was provided for both mutual authentication and secure key distribution. ECC algorithm in public key cryptography [6] is worked on Elliptical equations. ECC algorithm can be implemented through RSA and Diffie-Hellman algorithm. ECC algorithm provides 164 bit key where as other algorithms provides 1024 bit key size which is heavy.

MD-5 algorithm is used here instead of MD-4 algorithm [7]. MD-5 algorithm uses modular integer subtraction technique to find out the collisions occurring in a time span of 15 minutes to one hour. An HPC application [8] shows the need of high computing techniques which is used in cloud services like Amazon EC2platform. As an outcome of this, two problems were occurred namely Virtualization overhead and poor bandwidth. HPC applications were discussed in paper [11]. In paper [12], Pipelining concept in the cloud platform was discussed. SN factory pipeline to the Amazon Web Services environment is also discussed in this paper. Pin numbers of ATM cards can be easily leads to various chaoses were discussed in paper [13]. Virtual password was proposed with little amount of calculations. It
can prevent phishing, key logger, and shoulder-surfing attacks. An online transaction [14] shows encrypted cookies to encrypt the password along with the key. ECS [Encrypted Cookie Scheme] has the advantage of not being noted by the SSL indicator in the transaction process. A user keeps the same password [16] for all of his accounts because he cannot remember all complex passwords which lead to poor protection against phishing methods. To overcome this they have introduced the generation of dynamic passwords at run time. In paper [17], an inverse cookie based virtual password authentication protocol is being proposed to avoid meet in the middle attack and online dictionary attacks. Authenticated key exchange algorithms [18] to ensure the security on the internet against the brute force attack techniques. But on the other hand the efficiency of the algorithm is reduced. To overcome this one time ID created using Diffie-Hellman algorithm using SVO logic. In papers [19 – 21] they have used the concept of generating remote passwords using smart cards. So the user has no hassle of remembering all the user names and passwords for the accounts he use. This method is implemented with the help of ECC algorithms. A fast and Secure Authenticated Key Agreement (EC-SAKA) [22] protocol based on Elliptic Curve Cryptography is used in this paper. It provides secure mutual authentication, key establishment and key confirmation over a unreliable network. In paper [23] ID-based remote mutual authentication with key agreement scheme on ECC was discussed. It does not require public key thereby reducing the computations of the user. Various web link [9,10,15] also referred for this work.

III. Proposed System

In the proposed model, mutual authentication between client and the server is provided by Elliptic Curve Cryptography (ECC). The client first registers to the server by providing its ID to the server. The server in turn generates a password for the client and also selects one point from a set of points that satisfies the elliptic curve equation of the curve which is processed by the server. The server provides the password and the point generated to the server for the purpose of authentication. Once the client is authenticated by the server and the server is authenticated, they both agree over the generation of a mutual session key which is generated
by the concatenation of the Random number generated by the server along with its private key and also the device ID of the client. Thus in the case server’s private key is also preserved.

The Phases involved in the process are

1. Registration Phase
2. Pre-Computation and Login Phase
3. Authentication Phase

The steps involved in each phase are summarized as follows.

A. **REGISTRATION PHASE** as shown in figure 1 and 2.

Step1: Client sends Device id to server.

Step2: Server generates a password and a generator point that satisfies the elliptic curve equation of the server.

Step3: The generated password is stored in a database.

**Pseudo Code**

Base point selection algorithm of ECC

Constants a and b Prime number p as input and effective base point G on curve with order n.

Step 1: Select x randomly between 0 and p.

Step 2: Calculate \( a = (x^3 + ax + b) \pmod{p} \)

Step 3: If \( a \) belongs to the quadratic residue of mod p, then \( y \) value is obtained as \( G(x, y) \) and move to step 4, else go to step 1.

Step 4: Calculate \( hG \) according to the value of \( G \) and verify whether the point \( G \) meets \( y^2 = x^3 + ax + b \) and also make sure that \( G \) is not an infinite point. If everything is satisfied go to step 5, else go to step 1.

Step 5: Return \( G(x, y) \)

---

**Figure 1. Line Diagram of Registration Phase.**

Step 4: Calculate \( hG \) according to the value of \( G \) and verify whether the point \( G \) meets \( y^2 = x^3 + ax + b \) and also make sure that \( G \) is not an infinite point. If everything is satisfied go to step 5, else go to step 1.

Step 5: Return \( G(x, y) \)
Figure 2. Function of registration module which generates the password and Generator point that will satisfy the server’s elliptic curve equation.

B. PRE-COMPUTATION AND LOGIN PHASE as shown in figure 3 and 4

Step1: Client submits user id and corresponding password along with an ECC point obtained using the generator point given by the server.

Step2: Server checks for the validity of the password and also checks for the satisfaction of its elliptic curve equation on that point submitted by the client.

Step3: If both the conditions are satisfied, the server will send a point that would satisfy the elliptic curve equation of the server.

Figure 3. Line Diagram of Pre-Computation and Login Phase.

Figure 4. Function of Login module which checks the password and the point.
c. AUTHENTICATION PHASE as shown in figure 5

Step 1: Client checks whether the point given by the server satisfies its elliptic curve equation or not

Step 2: If the point satisfies Session key will be generated.

![Figure 5: Line Diagram of Authentication Phase.](image)

IV. Experimental Analysis

In the system used once the client authenticates the server, server concatenates the Client’s device id, Random number and private key of the server. The result of concatenation will be of variable length and is prone to computational errors. So it is given as an input to a hash function which will provide a session key of fixed length that will reduce the computational errors. The server can then store the session key for each client and use it for verifying the client before each and every instant of communication.

![Figure 6: Database that stores user id and password](image)

![Figure 7: Function of authentication module in which client checks for the validation of the point and the session key is generated.](image)
Validation testing is done by providing invalid user id, password and incorrect points as input to the Login module as shown in figure 6 and 7. Response for a wrong input is shown in figure 8.

![Figure 8: Response for a wrong input.](image)

V. Conclusion

Generation of points satisfying the equations of elliptic curves generated by the client and the server in a mutual way adds to the security of normal username and password systems. In addition introduction of hashing techniques to transmit the session with the processing power of client as a constraint we provide a second level of security without any extra effort required by the client. This method can be used to authenticate a client and a server in a mutual way. The proposed system can be further enhanced by using the session key generated at the end of the session on each and every instance of communication between the client and the server. The server can have the database of the possible hashes and can compare the session key provided by the client before staring a session of communication.

References


5. Charles Cavaiani and Jim Alves, 2013, Mutual authenticating protocol with key distribution in client/server environment, Computing – special issue on embedded systems.


