SUPPORTIVE SECURITY FOR DATA BEING TRANSMITTED OVER SSL/TLS

[1]ROBIN TOMMY, [2]HIMA JOSE, [3]AKHIL RAMESH

[1,2,3]ILP Innovation, Tata Consultancy Service, India

Abstract— The aim is to develop a mechanism by which we can provide additional security to the data being transmitted over secure socket layer (SSL) or transport layer security (TLS). SSL or TLS establishes a secured, bidirectional tunnel for arbitrary binary data between two hosts. The data which is being transmitted over network via SSL is encrypted before sending and is decrypting after receiving. So the idea for implementing this is to use some specific encryption standard for performing the encryption and decryption from the client and server. The public key cryptosystem is used to avoid the complexity of key exchange, and the algorithm used is RSA (Rivest, Shamir, Addleman).

Keywords— Information Security, Attack, Secure Socket Layer(SSL), Transport Security Layer(TLS), Encryption, Decryption.

I. INTRODUCTION

SSL is a cryptographic protocol that is widely used in secure applications based on web browser, such as e-payment. The protocol is based on the principle of PKI and uses digital certificates to realize secure communication. However, the use of certificate in SSL does not obey the strict hierarchy CAs, which causes some secure defects in SSL. Besides key length of SSL session is limited to 40 bits and the length is vulnerable to exhaustive attacks [1]. If an attacker is able to breach the security of SSL then the CIA triads will get compromised. It is better to take a precaution to avoid the further conflicts.

II. AIM

The aim is to provide more security to the data which is being transmitted over a network via SSL/TLS. Even the SSL is more secure the time where attackers can break the SSL is not so far. If something like breaking the SSL/TLS happened then all the sensitive data may get looted or get modified. The major concern of using the SSL is to preserve the CIA triads-Confidentiality, Integrity and Availability [3]. To add one more level of protection to the data what needs to be done is secure the data in some manner before sending it to network via SSL.

III. IDEA

The idea for achieving the aim is nothing but give one more level of security to the data before it is given to SSL for transmitting. For this the procedure is encrypting the data from client to the server and decrypting it from the server side. For this a good encryption algorithm is required and needs to implement it on both client and server side. Since the key exchanging will become a problem it is good to use a public key cryptosystem for the encryption and decryption also the public key cryptosystem is providing more security than private key cryptosystem [2]. Among the public key crypto system, RSA is the most secured algorithm to implement, which provides a pair of keys – one is public key and other is private key. As the name represents public key is for public and private key is for private use. The data encrypted using one key can only be decrypted using the other key.

IV. IMPLEMENTATION

The idea behind adding more security to the data can be implemented using RSA algorithm in both client side and server side. Since the encrypted data can only be decrypted using the other key, the data from client side needs to be encrypted by using the private key only. Because the data encrypted using private key can only be decrypted using the public key. Since the private key of the client needs to be confidential to client, so the data can only be encrypted using the private key so that the server can decrypt it using the public key of the client. But the public key is known to the public and the attacker who can capture the data while transmission can decrypt it using the server public key. So to solve this problem what needs to be done is before sending the client encrypted data the server will generate a pair of RSA keys. And share the public key of the server to the client. Then the client is again encrypting the client encrypted data with the server public key. After this second encryption the data will be transmitted through the network via SSL to the server.

The RSA public key cryptosystem is used to implement this mechanism. The working of this algorithm is as follows.
It has .mainly of three steps

A. Key generation

Key generation is the first step of the RSA algorithm.

a. Choose two distinct prime numbers p and q.
b. Find n such that n = p*q.

n will be used as the modulus for both the public and private keys

c. Find the totient of n, $\phi(n) = (p-1)*(q-1)$
d. Choose a public exponent e such that $1 < e < \phi(n)$, such that e and $\phi(n)$ are relatively prime.
e. Determine private exponent d such that $de \equiv 1 \pmod{\phi(n)}$.

B. Encryption

Consider m as a plain text to be encrypted, then the encryption is as follows:

$$\text{Cipher text, } c = m^e \pmod{n}$$

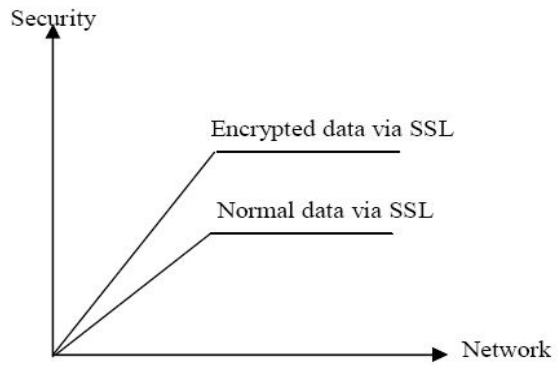
C. Decryption

The cipher text c can be decrypted as follows:

$$\text{Plain text, } m = c^e \pmod{n}$$

The level of security that SSL provides is high even though there is chance of breaking the SSL. Last year we have seen a malwares which itself sign the digital certificate of a well-known certification authority (CA) and accessing data without any interruptions. So we cannot assure the security of SSL even though it is unbroken till date. If some attack on existing SSL has succeeded in future the entire transactions which uses SSL will get compromised and it will affect the whole digital world. But what if the data which an attacker is getting after braking the SSL is encrypted with RSA. Usually it will take many years to decrypt the data. Hence we are providing an additional security over SSL.

The client side key generation is different from the server side key generation. In client side the key is generated on the basis of the default values given. It is possible to generate different size keys for our purpose. Since the n bit key can encrypt a data of maximum size (n/8)-11. Hence the 2048 bit key can encrypt a data of maximum size 245 bytes. So the data which is encrypted using the client private key may exceed the limit of 245 bytes, to solve this problem the data can be divided into blocks of 245 bytes and can encrypt.



As described earlier the RSA encryption is again performed in the encrypted data using the server public key. The procedure for encryption is same as explained before, a set of public and private key will be generated from server side and shares the public key with the client for each request. Once the key is generated the communication should be done within a particular time period. After this time period the server won't accept the public key shared to the client, this is a security mechanism for avoiding key duplication. If the key expired the client needs to create another connection to the server.

V. ADVANTAGES

This mechanism will be able to preserve confidentiality, integrity and the authentication and it is as follows. The data which is receiving in the server side is doubly encrypted, first the data needs to be decrypted using the private key of the server, since it possible to decrypt the data only with the private key of the server, it is assured that the confidentiality and integrity are preserved. And during the next decryption the data can only be decrypted using the public key of the specified client, which ensures that the data has come from the client to which the connection has been made. Hence the authentication is also preserved.

SSL provides secure communication against eavesdropping, and enables authentication of end hosts. Nowadays, SSL plays an essential role in online-banking, e-commerce, and other Internet services to protect passwords, credit card numbers, social security numbers and other private information [4]. The SSL is mainly deployed on security-critical domains and the attack is also targeted on such domains. So there is a possibility that the SSL can be breached. Here comes the importance of one more level of protection to the data.

VI. DISADVANTAGES

The major problem that this mechanism will face is the high computation overhead in both the client and server. For each connection the server needs to perform one additional key exchange other than SSL key exchange that involves expensive public key cryptography and the public-key decryption quickly becomes the bottleneck when large number of connections have to be established at server side [5]. For instance even the state of the art CPU core can handle 1000 HTTPS request per second, but the same core can handle around 500 request which is of the new type.

VII. FUTURE WORKS

The complexity of the algorithm is very high. It is a good option to implement an algorithm or mechanism which has less computation power or complexity than this. Also it is very difficult to modify each and every server and client application to implement this mechanism, for this it is better to develop a library or

something which can be easily added to the server and client instead of coding again.

REFERANCES

[1] Zhao Huwaei, “A scheme to improve security of SSL”, 2009.

[2] A comparison of public key and private key cryptosystem. http://www.dcs.ed.ac.uk/home/adamd/ essays/crypto.html

[3] O. Harrison and J. Waldron. Efficient Acceleration of Asymmetric cryptography, In Africacrypt, 2009.

[4] Keon Jang, Sangjin Han, Seungyeop Han, Sue Moon, KyoungSoo Park, ”Accelerating SSL with GPU”, ACM 978-1-4503-0201-2/10/08

[5] C. Coarfa, P. Druschel, and D. S. Wallach. Performance Analysis of TLS Web Servers. In NDSS, 2002.

★★★