Improving the Secure Socket Layer by Modifying the RSA Algorithm

Parshotam[1] and Rupinder Cheema[2] and Aayush Gulati[3]

[1]Department of Computer Engineering, Lovely Professional University (LPU), Punjab, India
Parshotampal90@yahoo.com
[2]Department of Computer Science and Engineering, PEC University of Technology Chandigarh, India.
cheemarupinder@gmail.com
[3]Department of Computer Engineering, Lovely Professional University (LPU), Punjab, India
aayush_kap23@yahoo.com

Abstract

Secure Socket Layer (SSL) is a cryptographic protocol which has been used broadly for making secure connection to a web server. SSL relies upon the use of dependent cryptographic functions to perform a secure connection. The first function is the authentication function which facilitates the client to identify the server and vice versa [1]. There have been used, several other functions such as encryption and integrity for the imbuement of security. The most common cryptographic algorithm used for ensuring security is RSA. It still has got several security breaches that need to be dealt with. An improvement over this has been implemented in this paper. In this paper, a modification of RSA has been proposed that switches from the domain of integers to the domain of bit stuffing to be applied to the first function of SSL that would give more secure communication. The introduction of bit stuffing will complicate the access to the message even after getting the access to the private key. So, it will enhance the security which is the inevitable requirement for the design of cryptographic protocols for secure communication.

Keywords

Secure Socket Layer, RSA, Bit Stuffing, TCP, HTTP

1. INTRODUCTION

SSL secures the communication by providing message Encryption, Integrity, and Authentication. The SSL standard allows the concerned components to negotiate the encryption, authentication, and integrity mechanisms to use. [2]

Secure Socket Layer (SSL) is a protocol used to make secure communication between a client and a server. [13] Both the Netscape and Internet Explorer support versions of SSL and the Internet Engineering Task Force (IETF) has approved SSL as a standard. SSL is located between the TCP and HTTP protocols [3]. In this case, HTTP is modified to be HTTP (Secure) abbreviated by HTTPS, which is the standard encrypted communication mechanism on the World Wide Web.

DOI : 10.5121/ijcsea.2012.2307

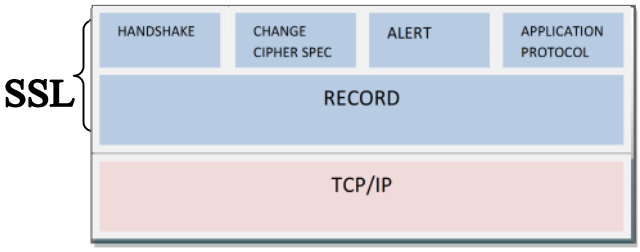


Figure 1. Secure Socket Layer

The SSL protocol is designed using three interdependent cryptographic functions [4]. Authentication is the first function found in SSL. Its goal is to perform identification and authentication of the parties involved in the communication. Authentication is achieved using public key encryption and a digital certificate issued by the trusted Certificate Authority [5]. There are many public key cryptographic algorithms that could be used to achieve authentication such as RSA, Diffie Hellman. RSA is the most common cryptographic algorithm used to achieve authentication [3]. RSA keys are classified into three categories as follows:

a) Short Key whose range is less than the 900 bits.
b) Medium key whose range is between the 900 and 1250 bits.
c) Long Key whose range is greater than 1250 bits.

- Encryption allows only the intentional recipient to read the message. SSL can use different encryption algorithms to encrypt the messages. During the SSL handshake that occurs at the start of each SSL session, the client and the server negotiate which algorithm to use. Examples of encryption algorithms which are supported by the SSL include AES, RC4, and 3DES.[12]
- Integrity ensures that a message sent by a client is received unharmed by the server. To ensure message integrity, the client hashes the message into a digest using a hash function and sends this message digest to the server. The server also hashes the message into a digest and compares the digests. The server can tell that if the digests do not match, then someone had altered the message. An example of a hash function supported by SSL is SHA1.[12]
- Authentication on the other hand enables the server and client to check that the other party is who it claims to be. When a client initiates an SSL session, the server typically sends its certificate to the client. Certificates are digital identities that are issued by trusted certificate authorities, such as VeriSign.[5]

At present the use of 512 bits keys are considered as unsecured keys after the successful attack and implementing the General Number Field Sieve (GNFS) algorithm. The GNFS algorithm is used to factorize n, where n is the multiplication of two large prime numbers p and q. It is computed by distributing the result over large number of computers. [6]

Its goal is to keep the communications confidential. This is performed using symmetric encryption scheme. Symmetric encryption means that both parties use the same key. Generally, symmetric encryption is used to exchange the messages between the client and the server and not asymmetric encryption (Public Key) because symmetric scheme is 1000 times faster than

asymmetric scheme. Asymmetric scheme is used only to exchange the symmetric keys between both parties [3].

For this reason, using secure public key is a necessity in order to secure our symmetric keys that are later used for encrypting the messages. The RC4 is the symmetric cipher used to encrypt the exchanged messages. Integrity is the last function used in SSL. Its job is to ensure the integrity of the data against interfering [12].

This is performed using message digests. The same RSA algorithms is used to compute a complex function based on two things first the message that was sent and second the secret keys known to both parties. The receiver computes the same function on the data that arrives. If the values computed at both parties matches then message integrity was checked indicating positive result. [12]

If an attacker knows the RSA key then the attacker can destroy the trust between the client and server and also can offer invalid information then can play the role of the server and in this case he can receive either personal or financial information from the user. This is really a very big problem.

Also, besides authentication, the use of RSA in SSL ensures the confidentiality of the data [7]. So, in this case the exchanged messages will be known. In order to prevent such types of attacks, it is usually recommended to use a larger size of RSA keys [8]. Currently, the key size that is considered to be secure is of length 2048 bits.

This paper is organized as follows: Section 2 describes a background of SSL. Section 3 describes the RSA used in the classical SSL, which depends on integer arithmetic. Section 4 describes the Modified RSA, which depends on BIT STUFFING that will be used in the modified SSL. Finally section V concludes the work.

2. BACKGROUND

SSL was designed by Netscape to perform a secure communication between a client and a server [9]. SSL uses three interdependent cryptographic algorithms. The first function is authentication. It is used to allow the client to identify the server and optionally allow the server to identify the client. SSL uses digital certificates to authenticate servers [7]. The most common cryptographic algorithm used in this phase is the RSA algorithm.

The second function is confidentiality that is used to keep the communication confidential. It uses symmetric cryptography to exchange messages confidentially. Integrity is the last function used to ensure the integrity of the data against snooping. This is performed using message digests. Checksum of the message is used for message digest. [12]

SSL uses three protocols to implement the above three algorithms Handshake protocol, Record protocol, and the Alert protocol. Handshake protocol is used to let a client authenticates a server. The Handshake protocol is shown follows in communication occurrence [10].

After the handshake is complete the Record protocol takes place. The exchange of the encrypted messages is handled in the Record protocol. If one of both parties finds an error, it sends an alert containing the error. This is handled in the Alert protocol. Change-Cipher-Spec message used in step 8 in communication occurrence is used to specify the following [11]:

- The hash algorithm used for MAC calculation such as MD5 or SHA-1.
- It also defines cryptographic attributes such as the hash size.

Communication [12] occurrence as follows:

1) The client sends the "Hello-Server" message.
2) Then server acknowledges with the "Hello-Client" message.
3) Server also sends its certificate.
4) And Optional: Server requests Client's certificate.
5) Also Optional: Client sends its certificate.
6) Now the client sends "Client - key - Exchange" message.
7) And the client will sends Certificate Verify message.
8) Finally both the Client and the server "Change - Cipher - Spec" message.
9) Now both the client and the server send "Finished" messages.

3. RELATED WORK

3.1 CLASSICAL RSA USED IN SSL

The authentication in SSL did using RSA. The standard value used for RSA Key was 512 bits. Then, a modified version of SSL was published using 1024 bits which is measured to be more secure but now it is currently recommended to use 2048 bits key for better secure communication [10]. The RSA used in SSL depends on the integer arithmetic. In order to generate a key with size 512 bits we need two distinct primes each with 256 bits size. 512 bits is equivalent to 155 decimal digits.

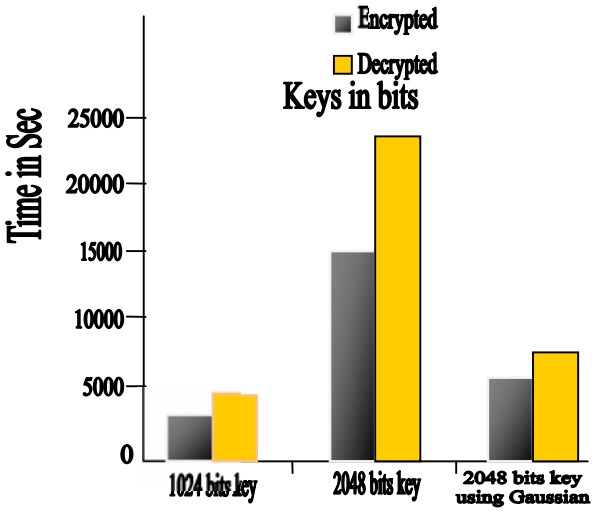
The standard RSA Algorithm [12] used for authentication is as follows:

1) Firstly find two large primes p and q and compute their product n =p x q.
2) Secondly find an integer d that is co-prime to $\varphi(n) = (p-1)(q-1)$.
3) Compute e from $e\, d \equiv 1 \bmod (p-1)(q-1)$.
4) Then broadcast the public key, which is the pair of numbers (e, n).
5) And represent the message to be transmitted, that is m, say as a sequence of integers {m} each in the range 1 to n.
6) Now encrypt each message, m, using the public key by applying the rule $C = m^e \pmod n$.
7) The receiver will decrypts the message using the rule $m = C^d \pmod n$.

3.2 EXPERIMENTAL RESULTS OF CLASSICAL RSA

In this section, the authors have compare and evaluate the classical and the modified authentication functions of SSL [12] by showing the run time results of three different examples as follows:

1) The 1024 bits key generated using two prime numbers each with 512 bits.
2) The 2048 bits key generated using two prime numbers each with the 1024 bits.
3) And 2048 bits key generated using two prime numbers each with 512 bits (In this they have used Gaussian Integer).



igure 2. Time for Encryption and Decryption

They have tested examples on messages and the corresponding results are shown in the above Figure 2 respectively.

They have also tested the three different key sizes on key exchange. Key exchange is usually used to exchange symmetric keys between parties and generally it uses 128 bits key.

From Figure 2, it can be conclude that the time needed to encrypt or decrypt a message using Gaussian integer with key size 2048 bits is double the time needed to encrypt or decrypt any message in the domain of integers with key size 1024 bits.

And it is concluded that while encryption and decryption using 2048 in the domain of integer is 6 times greater than the one uses 1024 bits. [12]

4. PROPOSED ALGORITHM AND RESULT

In this section, we will briefly present the modified version of RSA in the BIT STUFFING RSA. The idea behind this paper is to modify the RSA key from 512 bits to 512 bits by applying BIT STUFFING instead of ordinary integers using the same prime numbers used by the 512 bits. In this way we are making SSL more secure by using 512 bits and 512 bits for prime numbers. Confidentiality is the second function used in SSL [8].

The following is the proposed diagram for this modifies communication which we designed. From this diagram it is clear that the communication which will occur will be secure because of the keys are only known to the sender and receiver as follows:

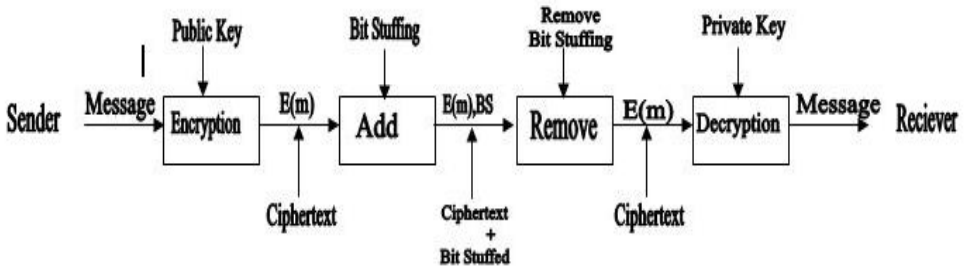


Figure 3. Encryption and Decryption

In data transmission and telecommunication, bit stuffing is the insertion of non-information bits into data. Stuffed bits should not be confused with overhead bits [8].

Bit stuffing is used for various purposes, such as for bringing bit streams that do not necessarily have the same or rationally related bit rates up to a common rate, or to fill buffers or frames. The location of the stuffing bits is communicated to the receiving end of the data link, where these extra bits are removed to return the bit streams to their original bit rates or form. Bit stuffing may be used to synchronize several channels before multiplexing or to rate-match two single channels to each other but I will use it as the bits which are not important for messaging but will be useful for security. Like these extra bits will confused to the attacker and then attacker will not determine the extra bits because only the authorized person will know which extra bits there are.

4.1 Modified RSA

The RSA algorithm has been modified from the domain of integers to the domain BIT STUFFING [8]. It was proposed that this modification is reliable and more secure than the classical RSA. The modified algorithm is as follows:

1) Find two large primes p and q and compute their product in =p x q with bit stuffing mechanism.
2) Find an integer d that is co-prime to $\phi(n) = (p - 1). (q - 1)$.
3) Compute e from e*d =1.mod (p - 1). (q -1).
4) Broadcast the public key, that is, the pair of numbers (e, n).
5) Represent the message to be transmitted, m, say as a sequence of integers {M} each in the range 1 to n.
6) Encrypt each message, m, using the public key by applying the rule $C= M^e \pmod n$.
7) Add the random number into C using
   C`=C+BS
8) Now remove random number at receiver side as
   C=C`-BS
9) The receiver decrypts the message using the rule $m =C^d \pmod n$.

4.1.1 Case Study

In the case study we have took an one general example and performed the operation by using the algorithm

1) Firstly we have to select the primes numbers as p=11, q=3.
2) Compute n = p*q = 11*3 = 33 phi = (p-1)(q-1) = 10*2 = 20
3) Now choose the value of e=3, Check gcd(e, p-1) = gcd(3, 10) = 1 because 3 and 10 have no common factors except 1 and check gcd(e, q-1) = gcd(3, 2) = 1, therefore gcd(e, phi) = gcd(e, (p-1)(q-1)) = gcd(3, 20) = 1
4) Now we have to compute d such that ed ≡ 1 (mod phi), compute d = e-1 mod phi = 3-1 mod 20 and find a value for d such that phi divides (e*d-1) and find d such that 20 divides 3d-1. Simple testing (d = 1, 2,) gives d = 7. Now check ed-1 = 3*7 - 1 = 20, which is divisible by phi.
5) Now the Public key = (n, e) = (33, 3) and Private Key = (n, d) = (33, 7).
6) Now say we can encrypt the message m = 7 as follows in next step.
7) $C = m^e \bmod n = 7^3 \bmod 33 = 343 \bmod 33 = 13$.
   The cipher text C = 13.

Now the generated bit stuffed number is let's saying 27, we will append it in the C on particular defined position as follows:

C`=C+BS

C`=13+27

8) To decrypt the cipher text, firstly we have to remove the BS from C` as
   C=C`-BS
   C=1327-BS
   C=13
9) To check decryption we compute as follows:
   $M = C^d \bmod n$

This way we can easily encrypt and decrypt the message and can secure the communication.

5 CONCLUSION

In this paper we have proposed a framework for securing the communication between the client and the server in SSL. The RSA algorithm has got several vulnerabilities that may be exploited, thus facilitating hacking of the algorithm. So, there is a need for devising security mechanisms for the same to thwart the exploited breaches. In this paper, modified RSA algorithm has been implemented which incorporates the use of bit stuffing. This mechanisms being followed will enhance the security as the generated number will not be repeated. Moreover, as with the implication of this novel algorithm, intruders irrespective of having access to the private will not be able to access the message as knowledge of bit stuffing is too required prior to accessing the message. So, this enhanced the security of the algorithm thus widening its domain of trusted usage.

REFERENCES

[1] Yogesh Joshi, Debabrata Das, Subir Saha, International Institute of Information Technology Bangalore (IIIT-B), Electronics City, Bangalore, India. "Mitigating Man in the Middle Attack over Secure Sockets Layer, 2009

[2] What is SSL and how the SSL works http://docs.oracle.com/cd/E17904_01/core.1111/e10105/sslconfig.htm

[3] A. J. Kenneth, P. C. Van Orshot and S. A. Vanstone, Handbook of applied Cryptography, CRC press, 1977.

[4] IT security web site, The Secure Sockets Layer Protocol Enabling Secure Web Transactions,http://www.verisign.com/ssl/ssl-information-center/how-ssl-security-works/index.html

[5] RSA website, 5.1 Security on the Internet, http://www.emc.com/security/rsa-securid/rsa-authentication-manager.htm

[6] IT security web site, the risks of short RSA keys for secure communications using SSL, http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4259828&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D4259828

[7] H. Otrok, Security testing and evaluation of Cryptographic Algorithms, M.S. Thesis, Lebanese American University, June 2003.

[8] Bit-Stuffing http://en.wikipedia.org/wiki/Bit_stuffing

[9] Cisco Systems, Introduction to Secure Sockets Layer, http://www.ehacking.net/2011/05/secure-sockets-layer-ssl-introduction.html

[10] A. 0. Freier, P. Karlton and P. C. Kocher, The SSL Protocol, version 3.0, http://www.cryptoheaven.com/Security/Presentation/SSL-protocol.htm

[11] W. Stallings, Cryptography and Network Security, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1999.

[12] H. Otrok, PhD student, ECE Department, Concordia University, Montreal, QC, Canada and R. Haraty, Assistant Dean, School of Arts and Sciences, Lebanese American University, Beirut, Lebanon and A. N. El-Kassar, Full Professor, Mathematics Department, Beirut Arab University, Beirut, Lebanon "Improving the Secure Socket Layer Protocol by modifying its Authentication functions" 2006

[13] Krishna Kant and Ravishankar Iyer Server Architecture Lab Intel Corporation, Beaverton, OR Prasant Mohapatra Dept. of Computer Science and Engineering Michigan state University, East Lansing, MI," Architectural Impact of Secure Socket Layer on Internet Servers" 2000

Authors

Name: Parshotam
Branch: B.Tech, M.tech CSE
Address: Village Ladian, p.o. Dosnajh kalan,
Tehsil Phillaur, Jalandhar,
Punjab-144502
Name: Rupinder Cheema
Assistant Professor
Lovely Professional University
Name: Aayush Gulati
Branch: B.Tech, M.tech CSE
Address: