International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 4| ISSN: 2456-3307

# Web Security and Enhancement Using SSL

**Ajay Singh, Ramesh Loar**
Department of Computer Science and Engineering, Rao Pahlad Singh Group of Institutions, Balana, Mohindergarh, Haryana, India

## ABSTRACT

With the development of e-commerce, ssl protocol is more and more widely applied to various network services. It is one of key technologies to keep user's data in secure transmission via internet. This document majorly focuses on sslstrip which generates the most recent attack in the secure network connections. It strips out all the secure connections to unsecure plain connection. In this article we depict this attack and to nullify it, we have proposed a technique cum practical solution to strengthen data security by developing mozilla-firefox add-on and servlet code which will strengthen our defense against the https hijacking attacks. Internet users today depend daily on HTTPS for secure communication with sites they intend to visit. Over the years, many attacks on HTTPS and the certificate trust model it uses have been hypothesized, executed, and/or evolved. Meanwhile the number of browser-trusted (and thus, de facto, user-trusted) certificate authorities has proliferated, while the due diligence in baseline certificate issuance has declined. We survey and categorize prominent security issues with HTTPS and provide a systematic treatment of the history and on-going challenges, intending to provide context for future directions.

**Keywords :** HTTPS, SSL, SSLSTRIP

## I. INTRODUCTION

Cyber security is very useful in every field of today's world such as military, government and even in our daily lives. [1] Today, everything is connected to internet from simple shopping to defense secrets as a result there is huge need of cyber security. Billions of dollars of transactions happens every hour over the internet, this need to be protected. Even a small unnoticed vulnerability in a network can cause serious damage. In every field of Internet, whether it is financial, personal or business everyone wants to know whom they are communicating with, ensuring that their data can be sent securely, and whether it has reached the destination correctly. Cyber security is the continuing effort to protect electronic data and computer systems from unwanted intrusions. Transmission of data over a network implies a possible loss of confidentiality, message integrity or endpoint authentication.

In This chapter we define fist we use http protocol to make a secure connection but after some time we see it is not secure properly we have to need some new protocol it is not work on dedicated IP in which the algorithm use which is commonly use and hacker known about these algorithm so which can easily hack the all information it is use encryption method to established a connection between the client and server.

After that we use new protocol which is HTTPS which is make secure connection between the client and server it can be use SSL Certificate.

## II. Related Work

**Er. Prabhjot Kaur, Er. Gurjeet Kaur, May 2017**

In this paper, we focus on SSL because it can secure millions of peoples' data every second, during online transactions or when transmitting confidential information over Internet. With the data encryption up to 256-bits, SSL protocol converts data into virtually incomprehensible code that is safe from hackers and identity thieves and increases the confidence of users during transactions. It also provides confidence in the integrity and security in online business and network infrastructure. Thus, we can say that SSL is the backbone of secure Internet.

**Ahmed Elnaggar, October 2015 (**network engineer for the ministry of communication and information technology, Egypt)

The Secure Sockets Layer (SSL) protocol uses a combination of public-key and symmetric-key encryption. Symmetric-key encryption is much faster than public-key encryption; however, public-key encryption provides better authentication techniques. An SSL session always begins with an exchange of messages called the SSL handshake. The handshake allows the server to authenticate itself to the client by using public-key techniques, and then allows the client and the server to cooperate in the creation of symmetric keys used for rapid encryption, decryption, and tamper detection during the session that follows. Optionally, the handshake also allows the client to authenticate itself to the server.

**Mohammed A. Alnatheer , Sept 2014**

SSL is very computational intensive. The increase in total processing time, a result of decrypting a message that was encrypted with a public-key algorithm, is quite CPU intensive. Furthermore, SSL handshakes are performance-intensive operations because of the cryptographic operations using the public and private keys. So, Handshake processing takes up a lot of CPU time. The aforementioned are the most influential reasons for increasing the percentage of the total processing time.

## III. Proposed Work

SSL/TLS is a deceptively simple technology. It is easy to deploy, and it just works--except when it does not. The main problem is that encryption is not often easy to deploy *correctly*. To ensure that TLS provides the necessary security, system administrators and developers must put extra effort into properly configuring their servers and developing their applications.

In 2009, we began our work on SSL Labs because we wanted to understand how TLS was used and to remedy the lack of easy-to-use TLS tools and documentation. We have achieved some of our goals through our global surveys of TLS usage, as well as the online assessment tool, but the lack of documentation is still evident. This document is a step toward addressing that problem.

Our aim here is to provide clear and concise instructions to help overworked administrators and programmers spend the minimum time possible to deploy a secure site or web application. In pursuit of clarity, we sacrifice completeness, foregoing certain advanced topics. The focus is on advice that is practical and easy to follow. For those who want more information, Section 6 gives useful pointers.

**Use 2048-Bit Private Keys**

For most web sites, security provided by 2,048-bit RSA keys is sufficient. The RSA public key algorithm is widely supported, which makes keys of this type a safe default choice. At 2,048 bits, such keys provide about 112 bits of security. If you want more security than this, note that RSA keys don't scale very well. To get 128 bits of security, you need 3,072-bit RSA keys, which are noticeably slower. ECDSA keys provide an alternative that offers better security and better performance. At 256 bits, ECDSA keys provide 128 bits of security. A small number of older clients don't support ECDSA, but modern clients do. It's

possible to get the best of both worlds and deploy with RSA and ECDSA keys simultaneously if you don't mind the overhead of managing such a setup.

### 1) Use Strong Certificate Signature Algorithms

Certificate security depends *(1)* on the strength of the private key that was used to sign the certificate and *(2)* the strength of the hashing function used in the signature. Until recently, most certificates relied on the SHA1 hashing function, which is now considered insecure. As a result, we're currently in transition to SHA256. As of January 2016, you shouldn't be able to get a SHA1 certificate from a public CA. The existing SHA1 certificates will continue to work (with warnings in some browsers), but only until the end of 2016.

### 2) Use Secure Protocols

There are five protocols in the SSL/TLS family: SSL v2, SSL v3, TLS v1.0, TLS v1.1, and TLS v1.2:

SSL v2 is insecure and must not be used. This protocol version is so bad that it can be used to attack RSA keys and sites with the same name even if they are on an entirely different servers (the DROWN attack).

SSL v3 is insecure when used with HTTP (the POODLE attack) and weak when used with other protocols. It's also obsolete and shouldn't be used.

TLS v1.0 is also a legacy protocol that shouldn't be used, but it's typically still necessary in practice. Its major weakness (BEAST) has been mitigated in modern browsers, but other problems remain.

TLS v1.1 and v1.2 are both without known security issues, but only v1.2 provides modern cryptographic algorithms.

SSL v1.2 should be your main protocol because it's the only version that offers modern authenticated encryption (also known as AEAD). If you don't support SSL v1.2 today, your security is lacking.

In order to support older clients, you may need to continue to support TLS v1.0 and TLS v1.1 for now. However, you should plan to retire TLS v1.0 in the near future. For example, the PCI DSS standard will require all sites that accept credit card payments to remove support for TLS v1.0 by June 2018.

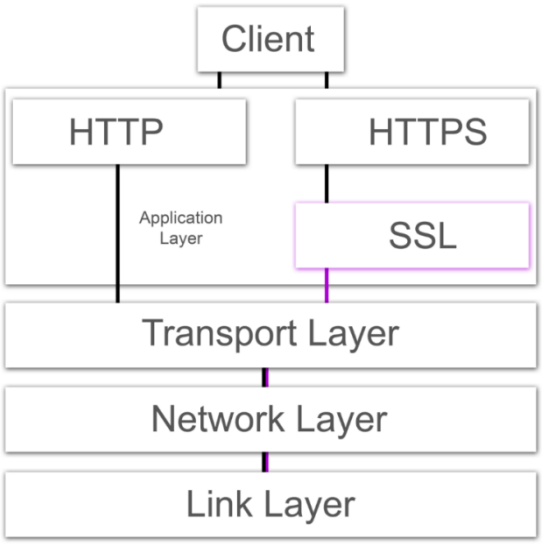


Fig 1: SSL Architecture

SSL uses these protocols to address the tasks as described above. The SSL record protocol is responsible for data encryption and integrity. It is also used to encapsulate data sent by other SSL protocols, and therefore, it is also involved in the tasks associated with the SSL check data. The other three protocols cover the areas of session management, cryptographic parameter management and transfer of SSL messages between the client and the server. Prior to going into a more detailed discussion of the role of individual protocols and their functions let us describe two fundamental concepts related to the use of SSL.

## IV. Result And Discussion

The SSL protocol is designed using three interdependent cryptographic functions 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 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 he use of RSA in SSL ensures the confidentiality of the data 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 Currently, the key size that is considered to be secure is of length 2048 bits.

## 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 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 used for authentication is as follows:

1) Firstly find two large primes pand q and compute their product n =p x q.
2) Secondly find an integer d that is coprime to φ (n) = (p1)(q1).
3) Compute e from e ≡l mod (p1)(q1).
4) Then broadcast the public key, which isthe 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 (mod n).
7) The receiver will decrypts the message using the rule m = C (mod n).

## Experimental Results of Classical Rsa

In this section, the authors have compare and evaluate the classical and the modified authentication functions of SSL 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).

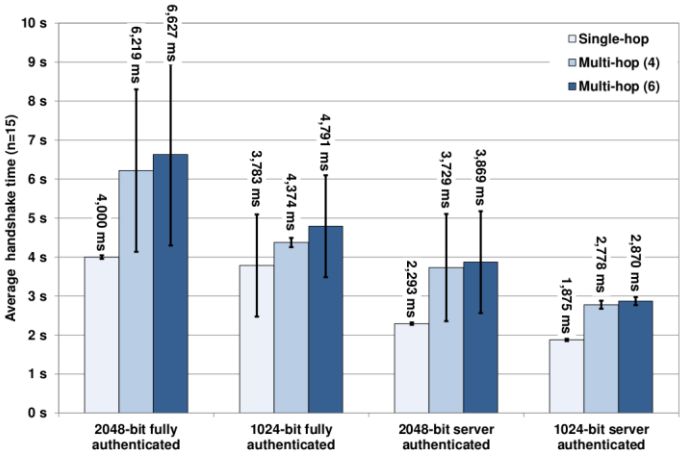

Figure 4.1 Time for Encryption and Decryption

They have tested examples on messages and the corresponding results are shown in the Above Figure 4.1 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 4.1, 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.

### 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

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: 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 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.

## V. Conclusion

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.

RSA claims that 1024-bit keys are likely to become crackable some time between 2006 and 2010 and that 2048-bit keys are sufficient until 2030. The NIST recommends 2048-bit keys for RSA. An RSA key length of 3072 bits should be used if security is required beyond 2030.

## VI. REFERENCES


1. Kartikey Agarwal and Dr. Sanjay Kumar Dubey, " Network Security : Attacks and Defence." IJCSE 2016
2. Mr. Pradeep Kumar Panwar and Mr. Devendra Kumar," Security through SSL ." in International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 12, December 2012.
3. Confidentiality integrity and availability CIA http://whatis.techtarget.com/definition.
4. Encryption and secret key cryptography cryptography/www.wikipedia.org.
5. Network Security: History, Importance, and Future by University of Florida Department of Electrical and Computer Engineering Bhavya Daya.
6. Mohammed A. Alnatheer , " Secure Socket Layer (SSL) Impact on Web Server Performance ." in Journal of Advances in Computer Networks, Vol. 2, No. 3, Sept 2014.
7. K Kant, R. Iyer, and P. Mohapatra, "Architectural impact of secure socket layer on

internet servers: A Retrospect" in Proc. International Conference on Computer Design.
8. K Kant, R. Iyer, and P. Mohapatra "Architectural impact of secure socket layer on internet servers" in Int. Conf. on Computer Design, pp. 7-14, 2000.
9. SSL Certificate Explained by Scion Solutions Ltd.
10. SSL Information Center/What is an SSL Certificatehttps://www.globalsign.com/en-in.
11. MS.Bhiogade Patni Computer Services, Secure Socket Layer InSITE - "Where Parallels Intersect" June 2002.
12. 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
13. What is SSL and how the SSL works http://docs.oracle.com/cd/E17904_01/core.1111/e10105/sslconfig.htm
14. A. J. Kenneth, P. C. Van Orshot and S. A. Vanstone, Handbook of applied Cryptography, CRC press, 1977.
15. 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
16. RSA website, 5.1 Security on the Internet, http://www.emc.com/security/rsasecurid/rsa-authentication-manager.htm
17. 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%2Fieeexplor.ieeee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D425982
18. H. Otrok, Security testing and evaluation of Cryptographic Algorithms, M.S. Thesis, Lebanese American University, June 2003.