The Hardware Implementation of Improved RSA Algorithm

Ankita Nag, Vinay Kumar Jain

Abstract- RSA algorithm is an asymmetric key cryptography. It is a block cipher. RSA has stronger security than single key cryptography. RSA has a pair of key- a private key a and public key. Sender sends the message encrypting it with the public key of receiver. Receiver receives the message by decrypting it with its private key. RSA provides authentication and integrity. So it is used in SSL for key exchange. At present 512 bit is considered insecure after the implementation of General Field Sieve Number [1].So in reference paper the idea of bit stuffing is introduced. RSA is bit stuffed after encryption that means a random number is appended to the cipher text and sent. At receiver, stuffed bit that is that random number is removed and then the cipher text is decrypted. Bit stuffing is suggested as a logic or measure to be used instead of increasing the number of bits in RSA. Since larger bit numbers will require more time and effort for calculation, bit stuffing will save time and effort. In this paper, this idea is implemented in hardware. Same security as with larger bit number say 1024 can be get in almost same time with lesser bit numbers say 512 bits with lesser band width requirement. In this paper, improving the SSL using modified RSA algorithm, coding is done in MATLAB.

Keywords: - RSA, Bit Stuffing, public key cryptography, public key, private key, prime number

I. INTRODUCTION

The RSA algorithm is a public key cryptography or asymmetric key cryptography. It is named after Rivest, Shamir, Adleman. It has a pair of key- public key or encryption key (e) and private key or decryption key (d). Sender uses public key of receiver to send him message as cipher text. Receiver uses his private key to decrypt the cipher text.

A. METHODOLOGY OF RSA

Here p and q are two prime numbers. Both are large bit numbers say 512 bits.

(i) Select two large bit prime numbers p and q.
(ii) Calculate **n** = p x q.
(iii) Calculate **z** = (p-1) x (q-1)
(iv) Select encryption key **e : 0 <e< n**
(v) Select the private key (i. e. decryption key), **d** such that ed mod z = 1
(vi) Publish public key- {e, n}
(vii) Keep private key secret {d, p, q}
(viii) Encryption -Cipher Text, C = M^**e** mod n
(ix) Decryption - Plain Text, M = C^d mod n

The real challenge lies in selecting the two large bit prime numbers. This must give larger bit number n. This larger bit number must be tough to factorize.The larger the bit tougher the factorization. Here lies the strength of RSA.

Manuscript Received on September 2014.

Ankita Nag, Department of Communication, Shri. Shankaracharya Group of Institutions (SSGI), Bhilai, India.
Vinay Kumar Jain, Department of Communication, Shri. Shankaracharya Group of Institutions (SSGI), Bhilai, India.

RSA Algorithm is generally used in SSL for key exchange not for message exchange since it is 1000 times slower than symmetric key. For secure exchange of symmetric key, it is exchanged between client and server using asymmetric key.
This paper is organized as follows: Section II past work related to the project and experimental result of classical RSA Section III describes the proposed algorithm, Section IV describes the Modified RSA, which depends on BIT STUFFING which will be more secure than classical RSA we use. Finally section V concludes the work.

II. PAST WORK RELATED TO THE PROJECT

In the reference paper "Improving the SSL using RSA algorithm" the idea of bit stuffing is introduced. Here RSA algorithm is improvised to make it more secure. In this paper author has explained SSL Protocol uses RSA algorithm for key exchange. In this way SSL provides authentication and integrity to the message. But if the attacker is somehow able to detect the public key then he can see the whole message. So the message is disclosed completely. The attacker can misuse the information, and can create misunderstanding between the users. So there arises a need to make the message uncomprehending even after it is disclosed. In the paper the method introduced involves message encryption by RSA algorithm and then a bit stream is added to it. This bit stream is a random number not an integer. This is how bit stuffing is done in the message. Since RSA is a block cipher the message is broken into blocks. Each block is encrypted and bit stuffed and sent for the receiver. At the receiver the received cipher text with stuffed bit is processed. That is the stuffed bit is removed from the cipher text. And then this cipher text is decrypted to get the plain text or the original message. Here if the attacker sees the message, it remains ambiguous to him because of the bit stuffing. At present the use of 512 bits keys are considered as insecure 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]. At present 2048 bits is considered secure. It concludes that the increasing the number of bits is considered as a measure to obtain security. But this also increases processing time as shown in experimental results below:

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

A. EXPERIMENTAL RESULTS OF CLASSICAL RSA

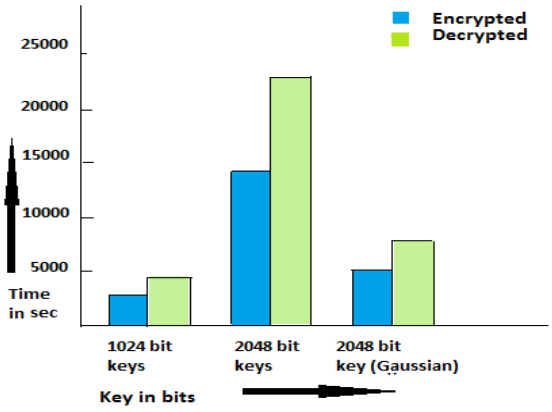

Figure 2 Time for Encryption and Decryption

In this section, the authors have compared and evaluated the classical and the modified authentication functions of SSL [12] by showing the run time results of three different examples for the 1024 bit key and the 2048 bit key generated using two prime numbers having 512 bit keys each and 2048 bit key generated using two prime numbers each with 512 bit keys (In this they have used Gaussian Integer). These are based on tested examples on messages and the corresponding results are shown in the above. 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]

III. PROPOSED ALGORITHM

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.[8] In this way we are making transmitted message more secure.

The following is the proposed diagram for this modifies communication which we designed.

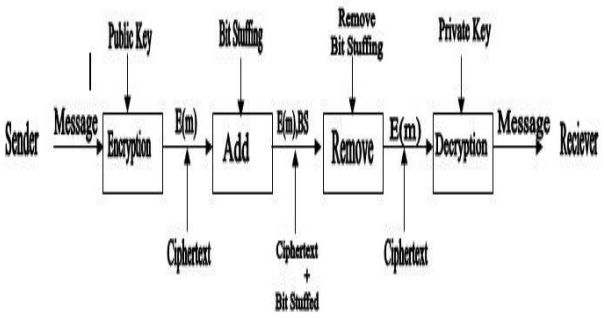

Figure 3 Encryption and Decryption

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 [8]. Improved RSA using bit- stuffing can be used for key exchange in SSL making it more secure.Improved RSA will take less time and effort to while using as compared to using number with more number bits. Bit stuffing technique has been used for frame synchronisation in communication system so improved RSA will provide good synchronisation. Bit stuffing is used for synchronizing bit rates or to fill buffers or frames. Bit stuffing may be used to synchronize several channels before multiplexing or to rate-match two single channels to each other.[14]Here appended bits is not part of message but its only purpose is security.

IV. METHODOLOGY OF IMPROVED RSA

Here p and q are two prime numbers. Both are large bit numbers say 512 bits.

(i) Select two large bit prime numbers p and q.
(ii) Calculate **n** = p x q.
(iii) Calculate **z** = (p-1) x (q-1)
(iv) Select encyption key **e : 0 <e< n**
(v) Select the private key (i. e. decryption key), **d** such that ed mod z = 1
(vi) Publish public key- {e, n}
(vii) Keep private key secret {d, p, q}
(viii) Encryption -Cipher Text, C = M^**e** mod n
(ix) Appending bits, C1 = [C Appended bits]
(x) At Receiver, appended bits are removed C = [C]
(xi) Decryption - Plain Text, M = C^d mod n

V. CONCLUSION

The message is successfully encrypted, appended with random number and sent to the receiver. At the receiver end original message is obtained after removal of the appended bit and decryption.

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/rsaauthentication-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/securesockets- 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

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

[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
[14] Purshotam, Dept. of Computer Engineering, Lovely Professional University, Punjab and Rupinder Cheema, PEC University of Technology, Chandigarh and Ayush Gulati, Lovely Professional University, Punjab ," Improving the Secure Socket Layer using modifying RSA algorithm" 2012
[15] Atul Kahate, Cryptography and Network Security, Tata McGraw-Hill Education, 2003

Ankita Nag, Branch: B.E. (Electronics and Telecom.), pursuing M.E. (Comm.), SSGI, Bhilai, Chhattisgarh, India.

Vinay Kumar Jain, Assoc. Prof., SSGI Bhilai, Chhattisgarh, India.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.