Modifications in RSA through Positive Justification of Random Components

Pallavi Jindal
Student
Rayat Institute of Engineering and Information Technology

Vikas Gupta
Associate Professor
Rayat Institute of Engineering and Information Technology

ABSTRACT
There is a surge of interest today in Wireless Networks both for organizations and individuals. It's emerging as general consensus that the current generation as well as next generation will have more of wireless connectivity. However, wireless networks have many security problems. Effective Wireless Security policies are required as wireless networks have been found to be relatively easy to break in by the hackers. The various techniques are being used for counteracting security risks. The most prevalent algorithm warranting security is RSA .RSA, a public - key cryptography algorithm is based on the factorization of large prime numbers. Presently the use of 512 bits keys are considered as insecure keys after the successful attack and by implementing the General Number Field Sieve (GNFS) algorithm .It has still got gaps of security that needs to be handled. In this, modifications has been proposed in RSA 512 bit and modified RSA through positive justification of alphabets through appending in the plain text and prime numbers in cipher text through multiplication to make the communication more secure. As the large prime numbers are not easily perishable, this modification will provide reliability over the network .This paper proposes a modification in classical RSA and modified RSA in which random numbers are appended in cipher text leading to improvement in previous results. The introduction of positive justification in this paper will make access to the message cumbersome even after having the access to the private key.

Keywords
Positive Justification, RSA cryptography, Random Prime numbers, Random Alphabets

1. INTRODUCTION

Cryptography is the study of techniques for secure communication in the presence of third parties. [8]More often, it is about constructing and analyzing protocols that overcome the influence of adversaries and which are related to various aspects in information security such non repudiation, integrity, authentication. Applications of cryptography include e commerce, computer passwords, and Credit cards. In cryptographic systems, the term key refers to a numerical value used by an algorithm to make changes in information, making that information secure and visible only to individuals who have the corresponding key to recover the information. Transport Layer Security and its predecessor, Secure Sockets Layer [3][6](SSL) are cryptographic protocols that provide communication security over the Internet. The both protocols encrypt the segments of application layer for transport layer using asymmetric cryptography for key exchange. SSL uses three interdependent cryptographic functions to perform a secure connection that allows secure communication. The first function is authentication. It lets the client to identify the server and optionally allow the server to identify the client. 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 .Confidentiality is the second function in SSL. Its goal is to keep the communications confidential. This is performed using symmetric encryption scheme. In Symmetric encryption mechanism, both parties use the same key. Integrity is the last function used in SSL. Its job is to ensure the integrity of the data against interfering. This is performed using message digests. There are many public key cryptographic algorithms that could be used to achieve authentication such as RSA, Diffie Hellman. Presently the use of 512 bits keys are considered as insecure keys after the successful attack and by implementing the General Number Field Sieve (GNFS) algorithm. The GNFS [6] [7] algorithm is used to factorize n, where n is the multiplication of two large prime numbers p and q .This paper will show modifications in classical RSA of 512 bits from an area of integers to an area of positive justification to be applied to the first function of SSL that would give more secure communication and that would make the plain text and cipher text more complex so that it cannot be easily decrypted**.**

2. BACKGROUND

RSA is an algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem .A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message .Whether breaking RSA encryption is as hard as factoring is an open question known as the RSA problem. RSA involves a public key and a private key. [1][7][10]The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The problem that was in classical RSA algorithm is that it was cracked by some hacker. And this way the security was broken. The RSA algorithm was very secure in previous years and was very useful for the security in the communication process. Basically this algorithm was designed for the authorization and encryption/decryption purposes. In this research paper, the method has been proposed for securing the communication through byte stuffing in the plain text and then doing the positive justification in the cipher text as

presently the use of 512 bits keys are considered as unsecured keys after for carrying bit streams that do not need fully have the same or analytically related bit rates up to a common rate, or to fill buffers or frames. The location of the justified 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. Positive justification [4] may be used to synchronize many channels before multiplexing but used it as the bits which are not important for messaging but will be useful for security .Like these extra bits will make the attacker confuse and then attacker will not determine the extra bits because only the authorized person will know which are the extra bits are there.

.

3. RELATED RESEARCHES

3.1 Classical Algorithm

The keys for the RSA algorithm are generated the following way:[7]

1. Choose two distinct prime numbers *b* and *f*.
    - For security purposes, the integer's *b* and *f* should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primarily test.
2. Compute *n* = *bf*.
    - *n* is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
3. Compute $\varphi(n) = \varphi(b)\varphi(f) = (b-1)(f-1)$where φ is Euler's totient function.
4. Choose an integer *e* such that 1 < *e* < φ(*n*) and gcd(*e*, φ(*n*)) = 1; i.e. *e* and φ(*n*) are co prime.
    - e is released as the public key exponent.
5. Determine *d* as $d^{-1} \equiv e\ (mod\ (\varphi\ (n))$i.e., *d* is the multiplicative inverse of *e* (modulo φ(*n*)).
    - This is more clearly stated as solve for *d* given $d*e \equiv 1\ (mod\ (\varphi\ (n))$
    - This is often computed using the extended Euclidean algorithm.
    - *d* is kept as the private key exponent.
6. By construction, $d \cdot e \equiv 1\ (mod\ (\varphi(n))$ The public

key consists of the modulus *n* and the public (or encryption) exponent *e*. The private key consists of the modulus *n* and the private (or decryption) exponent *d*, which must be kept secret. *p*, *q*, and φ(*n*) must also be kept secret because they can be used to calculate *d*.

3.1.1 Case Study

(i) Generation of a key

As cited above RSA involves [7] public key and private key . Public key is used for encoding and private key is used for decoding of message. The key generation takes place in the following way:

1. Two prime numbers *b* and *f* are chosen.
   For example let's take *b* = 5 and *f* = 7
2. Compute n by using following formula
   $n = b * f$
   Thus, n = 5 * 7
   n = 35
3. Compute $\varphi(n)$:

$\varphi(n) = (b-1)\ (f-1)$Here, phi($\varphi$) is Euler's totient

In our example:

$\varphi\ (n)$ = (5 – 1) * (7 – 1)

$\varphi\ (n)$ = 4 * 6

$\varphi\ (n)$ = 24

4. Choose the public key exponent e such that 1 < e < $\varphi\ (n)$ and e and $\varphi\ (n)$are co prime which means that gcd (e, φ (n)) = 1

Thus, we will choose e in the following range:

1 < e < 24

Let e = 5 as its co-prime to 24

5.Compute private key exponent d through following formula:

$d = e \wedge -1\ mod\ (\varphi\ (n))$

It means that d is the multiplicative inverse of e mod (phi (n)). Thus d can be found out as follows

$e * d = 1.mod(b-1)(f-1)$

So $5 * d = 1 * mod(24)$

d = 5

So, the public key consists of public key exponent e,n . And private key consists n and private key exponent d.

Public key: (n, e) = (35, 5)

Private key: (n, d) = (35, 5)

(ii)Encryption

For encrypting a message first the given message is turned to an integer number by using a suitable padding scheme. Then following formula is used to generate encrypted message c:

$C = M^e\ (mod\ n)$

Let's take m = 5 for example. Thus our encrypted message is:

C = 5 ^ 5 mod (35)

C = 3125 mod (35)

C = 10

(iii)Decryption

Following formula is used to decrypt the given message:

M = c ^ d mod (n)

M = 10 ^ 5 mod (35)

M = 100000 mod (35)

M = 5

Thus, we saw that we get the same message which was sent in an encrypted form after decryption using RSA algorithm.

3.2 Modified RSA

The classical RSA algorithm [6] has been modified from the domain of integers to the domain of BIT STUFFING. It was proposed as this modification is reliable and more secure than the classical RSA. The modified algorithm is as follows:

1) Find two large primes s and t and compute their product in $n= s * t$ with bit stuffing mechanism.

2) Find an integer d i.e. co-prime to $\varphi(n) = (s-1)\ (t-1)$

3) Compute e from $e * d$=1.$mod(s-1)\ (t-1)$

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) Encode each message, m, using the public key by applying the rule C= $M^e\ (mod\ n)$

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 decodes the message using the rule m =$C^d (mod\ n)$

4. PROPOSED ALGORITHM

In this proposed algorithm shown in Fig 1, the positive justification of random prime numbers in the cipher text and positive justification of random alphabets in the plain message is done. Positive justification [4] is referred to the insertion of non-information bits .It should not be confused with overhead bits .In this algorithm, random alphabets have been generated and stuffed in plain message {M} to obtain appended message M' and on then encryption is applied and then generated two random prime numbers on which multiplication operations is applied. After applying the multiplication operations on these random numbers the result has been multiplied with the cipher text on the sender side during encryption .And then on the decryption time, the same positive justification is removed firstly from the cipher text by dividing it and then from the appended message. In this way the hacker will have difficulty in accessing the message even after having access to the private key. This scheme will make the communication more secure and in this we have compared it with classical RSA and the modified RSA on the basis of total time taking for the decryption.

1) Find two large primes r and s and calculate their product in $t= r * s$

2) Find an integer d that is co-prime to $\varphi(t) = (r-1)\ (s-1)$

3) Compute e from $e * d$=1.$mod(r-1)(s-1)$

4) Transmit the public key, that is, the pair of numbers (e, t).

5) Exhibit the plain text message to be transmitted, M.

6) Generate the random alphabets and append in plain message to be encrypted say M'

$M'= M + DHRYH$

7) Encode each message, M, using the public key by applying the rule $CT = M'^e\ (mod\ t)$

8) Generate the random prime numbers into Cipher text using multiplication applying positive justification of non-repeatable random prime numbers

$CT'= CT * (R1 * R2)$

9) To decrypt, firstly remove the random prime numbers by dividing from CT' to obtain CT

$CT = CT' / (R1 * R2)$

10) To obtain the appended message $M' = CT^d (mod\ t)$

11) Now original plain message M the user transmitted is obtained by removing appended alphabets from M'

$M = M' - DHRYH$

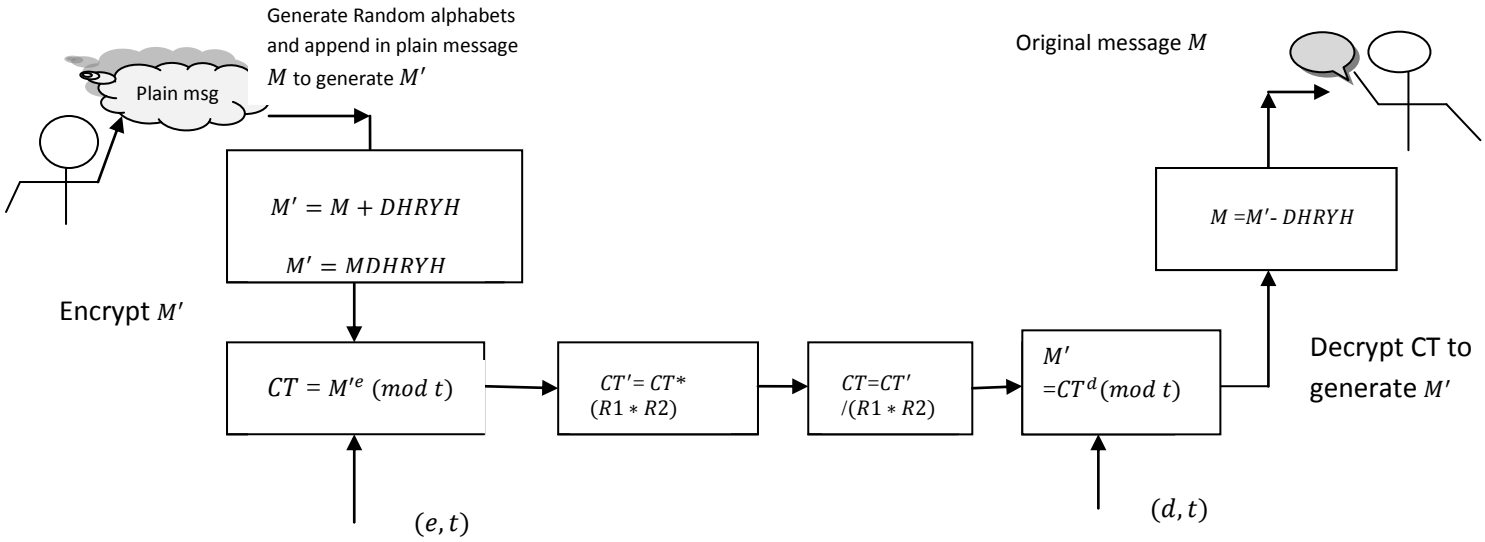

Fig.1 Shows the working of proposed algorithm

4.1 Case Study of Proposed RSA

1. Compute $t = r * s$ = 11*3 = 33 phi =$(r-1)$ $(s-1)$ = 10*2 = 20
2. Now choose the value of e =3, Check gcd$(e, r-1)$ = gcd(3, 10) = 1 because 3 and 10 have no common factors except 1 and check gcd$(e, s-1)$ = gcd(3, 2) = 1, therefore gcd(e, phi) = gcd$(e,\ (r-1)\ (s-1))$= gcd(3, 20) = 1
3. Now we have to compute d such that $e^* d \equiv 1$ (mod φ (t)), compute $d = e - 1\ mod\ \varphi$ = 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 $e^* d - 1$ = 3*7 - 1 = 20, which is divisible by phi
4. Now the Public key = (t, e)= (33, 3) and Private Key = (t, d) = (33, 7).
5. The plain text message M =123456 can be encrypted as follows in next step.
6. In the plain message the random alphabets say DHRYH is appended to generate M' $M' = M + DHRYH$ or M'=123456 $DHRYH$
7. Now convert this M' into an integer form using suitable padding scheme
8. Now encrypt the following padded message using $CT = M'^e\ (mod\ t)$ using the above values and generate Cipher text
9. After this ,do the positive justification of random prime numbers by multiplying into cipher text to obtain CT'= CT^* $(R1 * R2)$
10. On the decryption side remove the random prime numbers by dividing CT=$CT'/(R1 * R2)$
11. Now decrypt the cipher text using M' =$CT^d (mod\ t)$ to obtain M'
12. To obtain the original plain message do, M =M'- $DHRYH$ to get 123456

5. RELATED WORK RESULTS

In this section, the comparison and evaluation of the run time of the classical, modified and proposed RSA by showing the run time results by taking 2 different examples shown in Fig 2 and Fig 3. In the said figures above, we have taken 2 different messages. In the figure 2,

1) The time taken to encrypt the message "Hello clients, I need to check if you are in operation "using classical RSA of 512 key is 0.014399388 nanoseconds and the average time needed to decrypt the same message is 0.02660152 nanoseconds.

2) The time needed to encrypt the message "Hello clients, I need to check if you are in operation" using modified algorithm is 0.0151407 nanoseconds while the time needed to decrypt the message is 0.029399829 nanoseconds.

3) The time needed to encrypt" Hello clients, I need to check if you are in operation" using proposed algorithm by generating using two primes is 0.015068347 nanoseconds while the time needed to decrypt the message is 0.030832734 nanoseconds.

Similarly, in the Fig 3 the message is "Hello server, I am in operation" and computed the encryption time and decryption time. From the run times, it can see that decryption time of the proposed algorithm is almost more than half of the decryption time taken by classical and modified RSA. And it is already known if the decryption time of security algorithm is more it's said to be good.

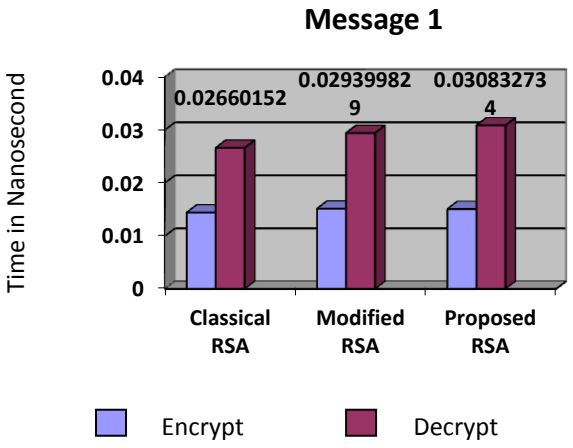

Fig.2 Shows comparison of Encryption and Decryption time taken in nanoseconds by these algorithms in Message 1

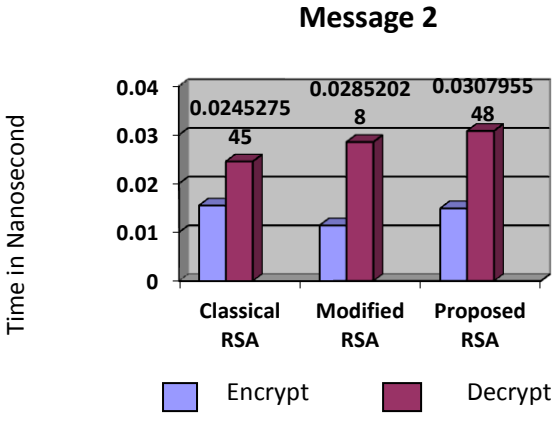

Fig.3 Shows comparison of Encryption and Decryption time taken in nanoseconds by these algorithms in Message 2

6. CONCLUSION

In this paper, we have proposed a method for coming over the several vulnerabilities that were faced in plain RSA and modified .As plain RSA is a deterministic encryption algorithm that is there is no random component and in the modified algorithm the numbers are appended .So in the proposed algorithm random components in the plain text and cipher text are generated. In this way the intruder will not come to know about the random alphabets and non – repeatable prime numbers and the operations applied on this random numbers. Going through the results, the proposed algorithm is giving enhanced results in terms of security as the time taken for decryption is more than the classical RSA and modified RSA as it is said if the cryptographic algorithm takes more time in decrypting encrypting its best algorithm. So proposed algorithm has outperformed. As RSA is also based on the prime numbers itself .This will make the communication between sender and receiver more secure.

7. REFERENCES

[1] Atul Kahate, Cryptography and Network Security, ISBN-10:0-07-064823-9, Tata McGraw Hill Publishing Company Limited, India, Second Edition, pages 38-62,152-165,205-240.

[2] Amit Kumar Gupta, Ravi Shankar Dhakar, Prashant Sharma, Modified RSA Encryption Algorithm (MREA), ACCT'12 proceeding of 2012 Second international conference on Advance computing and communication technologies page 426-429

[3] 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

[4] Wikipedia. (2013, March.). Bit Stuffing<Online> Available: http://en.wikipedia.org/wiki/Bit_Stuffing

[5] 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

[6] Parshotam, Rupinder Cheema and Aayush Gulati "Improving the Secure Socket Layer by Modifying the RSA Algorithm" International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, June 2012

[7] RSA website, 3.1 RSA, www.rsasecurity.com.

[8] Aayush Chhabra, Srushti Mathur "Modified RSA Algorithm: A Secure Approach" International Conference on Computational Intelligence and Communication Systems,2011

[9] Dhananjay Pugila, Harsh Chitral2, Salpesh Lunawat, **"**An Efficient Encryption Algorithm Based On Public Key Cryptography" International Journal of Engineering and Technology, July 2013

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