International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
Article Received: *20 May 2023* ***Revised:*** *28 May 2023* ***Accepted:*** *16 August 2023*

Chinese Remainder Theorem based Performance Analysis of RSA Cryptosystem

Yogesh Suryawanshi
Department of Electronics Engineering,
Yeshwantrao Chavan College of Engineering,
Nagpur, India
yogesh_surya8@rediffmail.com

Sachin Khade
Department of Electronics & Telecommunication Engineering,
Yeshwantrao Chavan College of Engineering, Nagpur, India
Nagpur, India
sac_mob@ rediffmail.com

D.B. Bhoyar
Department of Electronics & Telecommunication Engineering,
Yeshwantrao Chavan College of Engineering,
Nagpur, India
dinesh.bhoyar23@gmail.com

Arvind R. Bhagat Patil
Department of Computer Technology,
Yeshwantrao Chavan College of Engineering, Nagpur, India
Nagpur, India
arbhagatpatil@gmail.com

Pravin Zode
Department of Electronics Engineering,
Yeshwantrao Chavan College of Engineering,
Nagpur, India
pravin.zode@gmail.com

Abstract—Message security and authenticity is a very important issue which cannot neglect in wireless network. This paper explains how the RSA algorithm can be used to achieve both, over a wireless network. The RSA-CRT technique was premeditated for data decryption and operative illustration of cryptograpghy using the Chinese Remainder Theorem (CRT) for message security which is nearly four times faster.

Keywords- Encryption, Decryption, RSA, and CRT

I. INTRODUCTION *(HEADING 1)*

The RSA algorithm is predicated on the notion that factoring very big integers is inefficient. As a result, deducing an RSA key necessitates a significant amount of computer processing power and time [2].

A. KEY GENERATION ALGORITHM

This algorithm finds extreme applications in cryptography. Some of the notations used in this algorithm are as follows.

- n - modulus.
- e - exponent.
- d - furtive exponent

Key Generation Algorithm s as shown in Figure 1.

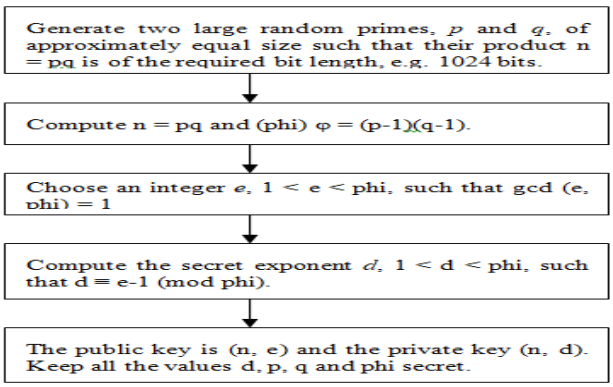

Figure 1. Key Generation Algorithm

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
Article Received: *20 May 2023* ***Revised:*** *28 May 2023* ***Accepted:*** *16 August 2023*

B. ENCRYPTION

Sender A follows the below stpes:

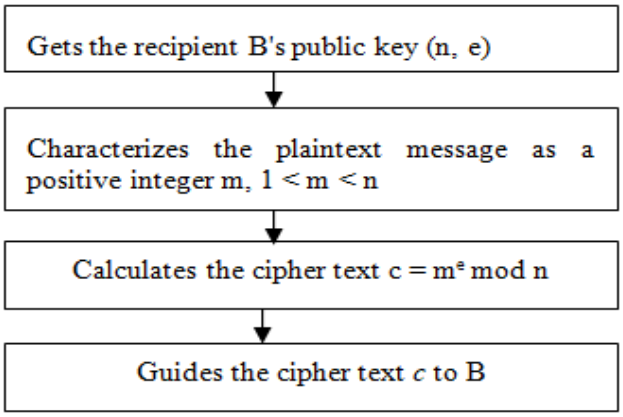

Figure 2: Encryption

C. DECRYPTION

Recipient B does the succeeding steps:-

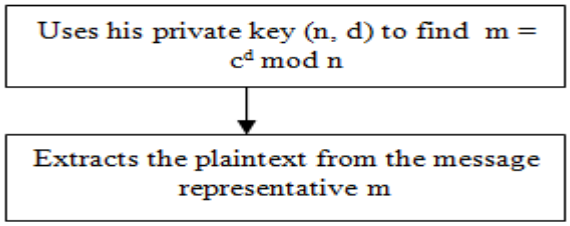

Figure 3: Decryption

D. EXAMPLE

Figure 4 shows an example of RSA encryption and decryption.

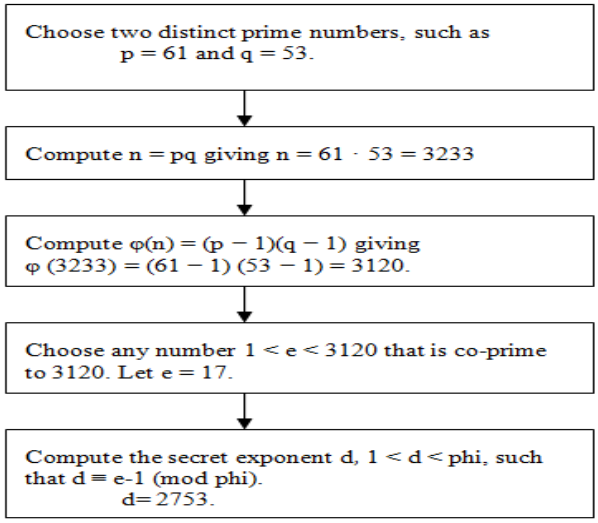

Figure 4: Example

E. EXTENDED EUCLIDEAN ALGORITHM

Calculating the inverse of the modulus function of one integer to another is common in cryptography systems.As an example, $x=y^{-1}\text{mod}(n)$. Calculating x for any given y can be challenging, as one might anticipate.

It is easy to calculate the inverse function x by using the Exte nded Euclidean Algorithm. In this Algorithm, inverse of the modulus function have calculated if gcd(x, n) = 1, then only inverse of x, y, exists. In other words, there exists an integer y for which xy + sn = 1, where y is the modular inverse of x mod n, for which xy + sn = 1.

F. WORKING EXAMPLE

To obtain d = e-1 mod n, first locate integers x and y such that xy + sn = 1, then calculate the inverse d.

We may determine d = 3-1 mod 20 using our technique above by finding gcd(20, 3) and checking that it is one (otherwise, the inverse cannot be found).

20=3x6+2

3 is equal to 2x1+1 resulting in gcd(20, 3) = 1. The numbers in this computation are then used to find xy + sn = 1.

1=3-1x2

1=3-1x(20-6x3)

1=-1x20+7x3

As x (the coefficient of 3) is 7, the inverse is also 7..

Hence the secret exponent d has been computed using Extended Euclidean algorithm which has been explained in the above section. We can observe from above example as modulus is large, to decrypt original message using private key consumes large amount of time. So it decreases computational efficiency.

If you able to reduce computational time we can improve computational efficiency.

The computational time elapsed in RSA algorithm functioning can be reduced to almost four times If we modify RSA algorithm using Chinese Remainder Theorem.

The application of the Chinese Remainder Theorem to increase decryption speed is a critical performance booster. The Chinese Remainder Theorem is an algorithm in arithmetic computation that allows modular arithmetic to be performed efficiently when the modulus is big (typically several hundred bits). The brief idea of Chinese Remainder Theorem is given in the next section

II.CHINESE REMAINDER THEOREM

The theorem's original version is a statement regarding modular arithmetic's simultaneous congruences. Integers of 1024, 2048, or 4096 bits are often utilised, which makes calculations take a long time. In this article, illustrate how using three extra values pre-computed from the prime factors

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
Article Received: *20 May 2023* ***Revised:*** *28 May 2023* ***Accepted:*** *16 August 2023*

of n, the CRT representation of numbers in Zn can be used to do modular exponentiation about four times faster.

A. RSA CALCULATIONS WITH CRT

The CRT provides a more efficient ay to calculate

$m = c^d \bmod n$.

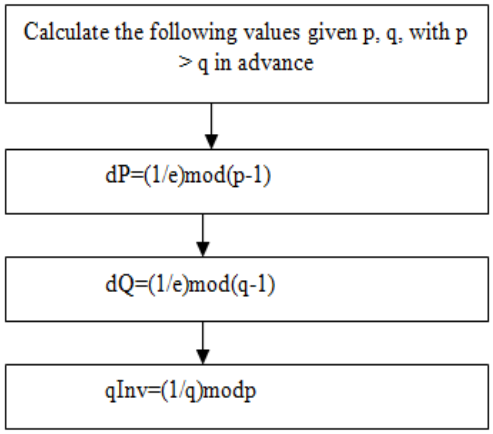

Figure 5: Steps in calculation

Do the following to compute the message m given c

$m_1 = c^{dP} \bmod p$
$m_2 = c^{dQ} \bmod q$
$h = qInv.(m_1 - m_2) \bmod p$
$m = m_2 + h.q$

We keep the private key in a quintuple (p, q, dP, dQ, qInv).

B. A SIMPLE EXAMPLE

This example comes from our RSA Algorithm.

Let p=137,q=131,n=137.131,e=3,d=11787,m=513

$c = 513^3 \bmod n = 8363$.

We might easily compute cd mod n to decode c:

m = 836311787 mod17947=513.

On a pocket calculator, this is quite difficult. Let's try the CRT way now.Take note of how much lower and more manageable the exponent and modulus numbers are. By breaking the RSA calculation down into two smaller ones, this simple (but insecure) example illustrates how easy it is to do the RSA calculation.

With a pocket calculator, we can compute this same result. Let us observed the results for 1024, 2048, 4096 bits RSA algorithm with and without CRT.

III. RESULTS & DISCUSSIONS

TABLE 1 . RSA ALGORITHM WITHOUT CRT

S.N	Key length (bits)	Time required for key generation (ms)		Generation time in ms		Verification time in ms	
		256R, p4	512R, p4	256 R, p4	512 R,p4	256R, p4	512R,p4
1	1024	32	16	0	0	16	16
2	2048	16	0	0	0	94	93
3	4096	16	31	0	0	609	625

TABLE 2. RSA ALGORITHM WITH CRT

S.N	Key length (bits)	Time required for key generation (ms)		generation time in ms		verification time in ms	
		256 R, p4	512 R,p4	256 R, p4	512R,p4	256 R, p4	512R,p4
1	1024	0	32	0	0	47	47
2	2048	16	16	0	16	297	297
3	4096	47	32	0	0	2297	2297

IV. CONCLUSION

It is evident from the results obtained for RSA algorithm with and without Chinese remainder theorem on 1024, 2048and 4096 bits respectively. Under Chinese remainder theorem RSA algorithm performs better and becomes almost four times faster as the computational efficiency improves.

REFERENCES

[1] C. Lamprecht "Investigating the efficiency of Cryptographic algorithm in Online Transaction's 1473-804X online, 1473-8031 I.J. of Simulation Vol.7, No.2.

[2] http://www.tml.hut.fi/studies/T110.501/2001/papers/index.htm.

[3] Ghasem S. Alijani, "Design and Implementation of an Information Security Model for E-Business", Information System Education Journal, Vol 4, no.4 Feb, 8, 2006.

[4] A. Sengupta, "E-Commerce Security- A Lie Cycle Approach", Sadhana Vol.30, Parts 2 & 3, April 2005 pp.119-140.

Dirk Balfanz, "A Security Infrastructure or Distributed Java Application".

[5] L. Ertaual and N. Chavan, "Security of Ad Hoc Networks and Threshold Cryptography", in MOBIWAC 2005.

[6] Alexander May, "Cryptanalysis of Unbalanced RSA with Small CRT-Exponent", CRYPTO 2002, LNCS 2442, pp 242-256, 2002.

[7] JohannnesBlomer, Martin Otto, "a new CRT-RSA Algorithm Secure Against Bellcore", CC'03, October 27-30,Washington, DC, USA.

[8] Dan Boneh and Hovav Shacham, winter/Spring 2002. Fast Variants o RSA.CryptoBytes- Vol 5, No. 1 Winter/Spring 2002. Pg 1-9

[9] Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2005, in press.

[10] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullende