An Improved and Secured RSA algorithm using CRT for text encryption

Paka Srinidhi

July 4, 2021

AN IMPROVED AND SECURED RSA ALGORITHM USING CRT FOR TEXT ENCRYPTION.

Paka Srinidhi
B.Tech, Dept. of CSE
KITS, Warangal
Telangana, India
pakasrinidhi@gmail.com

Abstract—**Every day, terabytes of data is produced in this internet and smart phone age. Data protection during contact over the internet is a significant challenge. Cryptography is an essential aspect of the internet's data protection system. Now-a-days intruders are trying to attack the data. In order to provide security for the data, we must provide some cryptographic algorithm, which involves encryption and decryption that plays a key role. Cryptography is defined as study of encipherment and decipherment techniques. Authorized people can access the data. Cryptography is of two types symmetric key and asymmetric key. An asymmetric key is also termed as public key. Confidentiality and authentication are the two cryptographic services offered by public key cryptosystems. The RSA algorithm is a well-known and widely used public key cryptosystem in which keys are generated using two large and distinct integers. We'll talk about an improved RSA algorithm for text encryption and decryption which uses Chinese Remainder Theorem (CRT) and provides more safety for the data.The proposed Enhanced and Secured RSA(ESRSA) algorithm generates public/private key pairs using four prime numbers rather than two primes, to increase the complexity .The encryption and decryption functions in the proposed RSA algorithm have been updated to improve security.**

***Keywords—** Euler totient function, Public key, Private key , Encryption, Decryption, RSA algorithm.*

I. INTRODUCTION

This Data protection and network security are becoming increasingly important in human life for a variety of hardware and software applications. Many human operations are now automated, and more places will become part of the network infrastructure in the future. As a result, the majority of devices would connect to the internet, making it critical to ensure data protection when being transmitted. Cryptography is an important aspect of information security in today's world, making the virtual world a safer place. In general, there are two types of cryptographic schemes that are commonly used: secret key (symmetric) cryptography and public-key (or asymmetric) cryptography, which are both listed below.

A) Symmetric Key Cryptography:-A single key is used for both encoding and decoding in secret key cryptography.For eg, suppose the sender P encrypts the plaintext message M with the key K and sends the ciphertext Ct to the receiver. To decode the cipher text Ct and recover the plaintext message M, the receiver uses the same key K. Secret key cryptography is also known as symmetric encryption because it uses the same key for both functions. The key must be identified by both the sender and the receiver in this form of cryptography; that is to be the secret. The main distribution is, of course, the most challenging aspect of this method. DES, AES, TRIPLE DES, and are some examples.

B) Public-Key Cryptography:- encryption and decryption are carried out using two keys: one public and one private. These keys are mathematically connected, but knowing one does not allow anyone to easily figure out the other. If the sender P encrypts the plaintext with the receiver Q's public key, for example. The recipient Q must then decode the cypher text Ct using his or her own private key and recover the plaintext message M. This method is known as asymmetric cryptography because it requires a pair of keys. The RSA, ECC, and Diffie-Hellman key exchange algorithms are the most popular examples of public key algorithm.

- Enhanced and Secured RSA Algorithm (ESRSA):

The RSA cryptosystem is a commonly used public key cryptosystem which was proposed by Rivest, Shamir and Adlemen in 1978 at MIT, that generates keys using two large and distinct integers. The three main functions of the RSA cryptosystem are key generation, message encryption, and message decryption. Many of Modified RSA algorithms are found in the literature but the security and complexity were the major issues in most of them. So, Using the Chinese remainder theorem, we proposed an improved and efficient RSA public key cryptosystem (ESRSA) algorithm in this paper. To increase the system's complexity, the proposed ESRSA algorithm generates the public/private key pairs using four prime numbers instead of two primes. The encryption and decryption functions in the proposed ESRSA algorithm have been updated to improve security. Since the encryption and decryption functions use an additional key parameter and are not wholly reliant on the public and private key pairs, the proposed scheme is extremely safe.

II. RELATED WORK

In the literature, there are numerous modified RSA key generation algorithms for secure communication. In most known modified RSA systems, the security and complexity of encryption and decryption algorithms are the primary concerns. It is vital to examine the many ways available in the literature in order to deal with these challenges. The following phrases provide an overview of some recently suggested fundamental changes to the classic RSA cryptosystem. Ivy et al. [2] proposed a modified RSA cryptosystem based on 'n' prime prime numbers that provides great security for confidential data sent over the network. The suggested cryptosystem employs 'n' prime numbers to guarantee maximal security. For safe file transmission, Jamgekar and Joshi [3] provide a modified RSA public key cryptosystem. Four prime numbers are employed in this suggested cryptosystem, and the key generation method is identical to that of the classic RSA cryptosystem. The encryption and decryption parts of this technique were made more difficult. The complexity of encryption and decryption functions, on the other hand, is extremely great. Furthermore, the system's overhead is increased because multiple new parameters are included without justification. In comparison to the typical RSA cryptosystem, the suggested technique is efficient in a number of aspects. Firstly it only requires one extra multiplication to implement encryption and decryption operations. The public/private key generation procedure is equivalent to the classic RSA cryptosystem and is based on four prime numbers. The encryption and decryption functions complexity is reasonable and also in order to guarantee excellent security, only one more parameter 'mu' must be computed. Because the encryption and decryption functions are not solely dependent on the public and private key parameters, the proposed scheme is extremely secured.

III. METHODOLOGY

The public/private key pairs produced by our modified RSA Algorithm uses four prime numbers instead of two primes. The proposed ESRSA uses a method identical to conventional RSA to produce the keys. In the proposed algorithm, the encryption and decryption functions are modified. The encryption function makes use of an additional parameter called encryption key, which is developed using secret parameters. To decrypt the code, the decryption function uses one extra parameter, either k1 or k2. In comparison to the conventional RSA cryptosystem, the proposed ESRSA Algorithm only needed one extra multiplication to perform the encryption and decryption operations. Therefore the encrypting and decrypting functions are dependent on these extra parameters which ensures more security. we have the 3 main functions in ESRSA as 1.Keygenertion, Message encryption AND Message decryption, which are explained in detail as follows.

- ALGORITHM:

A)Key-generation

Step 1: Firstly, we need to select randomly four large and distinct prime numbers i.e., u, v, w and z

Step 2: Then calculate n = u x v, m= w x z and A= n x m.

Step 3: Calculate Euler's totient function for n, m and A as follows

$\varphi(n) = (u-1) \times (v-1)$, $\varphi(m) = (w-1) \times (z-1)$ and $\varphi(A) = \varphi(n) \times \varphi(m)$.

Step 4: Now, select one more large prime number 'Pu' that is public key between 2 and φ(A). i.e., 2< Pu< φ(A) and GCD (Pu, φ(A)) = 1.

Step 5: Then compute the private key 'R' using 'Pu' and Phi_A. R such that (Pu x R)-1 = 0 mod φ(A), using (EEA).

Step 6: Choose two more new values p1 and q1 such that p1=A/n or 'm' and q1=A/m. Also, Calculate their multiplicative inverse i.e., p1_inv and q1_inv.

Step 7: Choose another two unique prime numbers N1 & N2 and find N1_inv which is multiplicative inverse of N1 & n and find N2_inv which is multiplicative inverse of N2 & m.

Step 8: Now, generating the new parameter used in this algorithm that is mu using the formula.

8.1. mu = ((N1 * p1 * p1_inv) + (N2 * q1 * q1_inv)) % A

Step 9: Finally, we have public and private key pairs as (mu, Pu, A) and (N1_inv, R, n) or (N2_inv, R, m) respectively.

Step 10: Now perform Encryption and decryption using the below formulas and get cipher text or encrypted message as EM and original plaintext as OM.

B) ESRSA Message Encryption

EM = mu*M ^Pu mod A

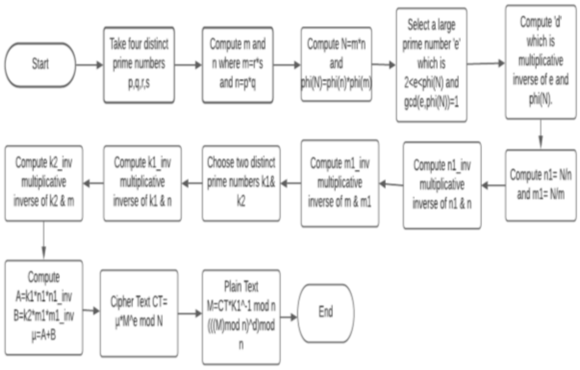
C)ESRSA Message Decryption

OM = (((EM*N1_invmod n) mod n) ^R) mod n

(Or)

OM= (((EM*N2_invmod m) mod m) ^R) mod m

- FLOW CHART:


Start
Take four distinct prime numbers p,q,r,s
Compute m and n where m=r*s and n=p*q
Compute N=m*n and phi(N)=phi(n)*phi(m)
Select a large prime number 'e' which is 2<e<phi(N) and gcd(e,phi(N))=1
Compute 'd' which is multiplicative inverse of e and phi(N).
Compute n1= N/n and m1= N/m
Compute n1_inv multiplicative inverse of n1 & n
Compute m1_inv multiplicative inverse of m & m1
Choose two distinct prime numbers k1& k2
Compute k1_inv multiplicative inverse of k1 & n
Compute k2_inv multiplicative inverse of k2 & m
Compute A=k1*n1*n1_inv B=k2*m1*m1_inv μ=A+B
Cipher Text CT= μ*M^e mod N
Plain Text M=CT*K1^-1 mod n (((M)mod n)^d)mod n
End

EXAMPLE:

- Message = " hi "
- Four prime numbers - p = 17 , q = 59 , r = 37 , s = 19
- n = p * q =1003 , m = r * s = 703
- N = n * m = 705109
- Euler's Totient of n , m ,N
- phi_n = (p - 1)*(q - 1) = 928,

- phi_m = (r - 1)*(s - 1) = 648 and
- phi_N = phi_n * phi_m = 601344
- e = 269341 , d = mul_inv (e , phi_N) = 167989
- n1 = N / n = 703 , m1 = N / m = 1003
- n1_inv = mul_inv (n1 , n) = 224 and
- m1_inv = mul_inv (m1 , m) = 546
- Select two random primes k1 = 53 , k2 = 11
- k1_inv = mul_inv (k1 , n) = 757 and
- k2_inv = mul_inv (k2 , m) = 64
- µ = ((k1 * n1 * n1_inv) + (k2 * m1 * m1_inv)) % N
- Therefore µ = 267854
- For each letter in message perform,
- ct = ((µ % N) * ((m % N)**e)) % N
- Each letter will be assigned to a specific value here for ‘a’ it is 100, similarly for ‘b’ it is 101 and this goes on.
- For ‘h’ cipher text will be 483112
- For ‘i’ cipher text will be 482140
- Similarly for each cipher text value decryption is performed using the below formula

- dm = (((ct * (k1_inv % n)) % n)** d) % n

- Two values for the two cipher text values after applying decryption will be 107 and 108, which changes to ‘h’ and ‘i’
- Final output after decryption will be “hi”

IV. REQUIREMENTS:

A. HARDWARE REQUIREMENTS:

Intel Pentium Dual core processor,
2 GB RAM and 200GB Hard-disk drive

B.SOFTWARE RQUIREMENTS:

Programming Language: python
OS: Windows7 or above

V. RESULTS AND DISCUSSION

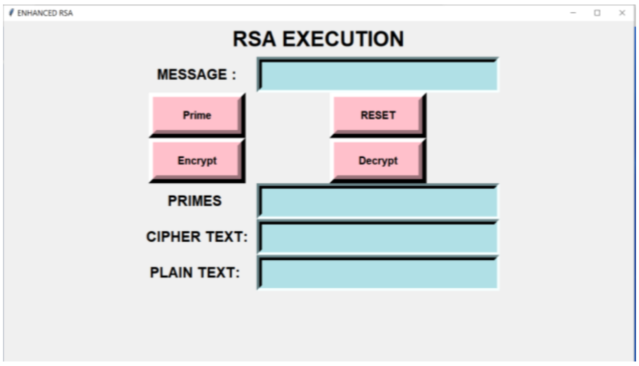
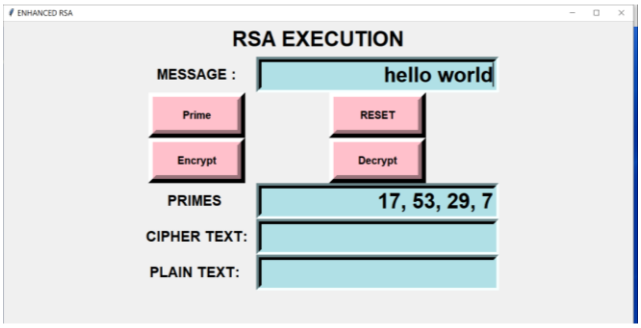


Figure-1

On executing the ESRSA algorithm using python we get the frame as shown in figure-1

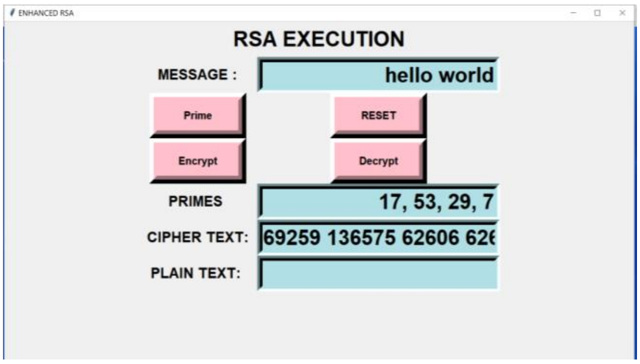
.



MESSAGE : hello world
PRIMES 17, 53, 29, 7

Figure-2

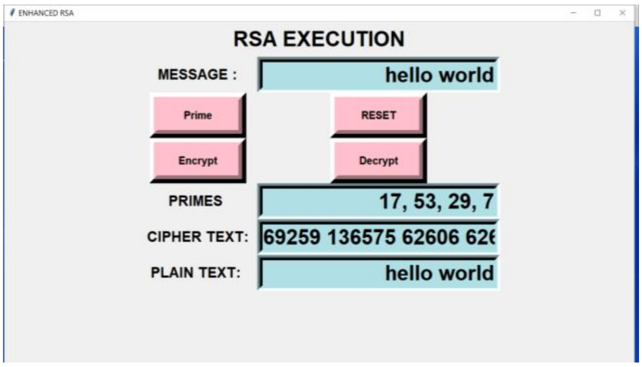
Now we need to give the message(here it is hello world) to be encrypted and generate primes using the prime buttonshown in figure-2. Later to encrypt we need to click on encrypt button to get our ciphertext which is shown in figure-3.



MESSAGE : hello world
PRIMES 17, 53, 29, 7
CIPHER TEXT: 69259 136575 62606 62

Figure-3

Finally to get back our original plain text we use decryp button and get our output as hello world as shown in figure-4.



RSA EXECUTION

MESSAGE : hello world

PRIMES 17, 53, 29, 7

CIPHER TEXT: 69259 136575 62606 626

PLAIN TEXT: hello world

VI. CONCLUSION

The ESRSA that we have seen makes use of four prime numbers and the encryption and decryption functions are modified, they use some special secret parameters like mu on encryption side and N1 or N2 on decryption side. Because of the primes and one extra multiplication the complexity of the algorithm is increased but the increase in complexity is acceptable as it is ensuring higher level of security. The new parameter ‘mu’ is generated using CRT. Unlike many existing algorithms the encryption and decryption processes in this algorithm are not only dependent on public (Pu, n) and private key (R, n) pairs but also on the new parameters ‘mu’ and ‘N1’ or N2’. So, even if the attackers or intruders tries to crack it, they may get the private key R in log N time using wiener’s attack or any other attack but, in order to get the special keys either N1 or N2 one need to perform brute force attack if for one time it takes 1ns then to obtain either N1 or N2 it takes about 2^l ns where l is the no. of bits of N1 or N2 used, which is impossible or impractical. In this way this ESRSA using CRT is ensuring high security of the information being transmitted from sender to receiver.

VII REFERENCES

[1] Vinod Kumar, Rajendra Kumar, and S. K. Pandey “An Enhanced and Secured RSA Public Key Cryptosystem Algorithm Using Chinese Remainder Theorem” , June 2018

[2] Ivy, P.U., Mandiwa, P., Kumar, M.: A modified RSA cryptosystem based on ‘n’ prime numbers. Int. J. Eng. Comput. Sci. 1(2), 63–66 (2012)

[3] Jamgekar, R.S., Joshi, G.S.: File encryption and decryption using secure RSA. Int. J. Emerg. Sci. Eng. (IJESE) 1(4), 11–14 (2013)

[4] Thangavel, M., et al.: “An Enhanced and Secured RSA Key Generation Scheme (ESRKGS)”. J. Inf. Secur. Appl. 20, 3–10 (2015)

[5] Rivest, R.L., Shamir, A., Adleman, L.: “A method for obtaining digital signatures and public-key cryptosystems”. Commun. ACM 21(2), 120–126 (1978)