Study of RSA Algorithm: Integer Factorization Approach for secure Data Transmission

Nilesh K. Kajale[1],Shubhashree S. Savant[2]

[1]Student MCA Department, MIT(E) ,
Dr.BabasahebAmbedkarMarathwada University Aurangabad, Maharashtra ,India.
nileshkkajale@gmail.com

[2]Assistant Professor, MCA Department, MIT(E),
Dr.BabashebAmbedkarMarathwada University Aurangabad, Maharashtra ,India.
Shubhashree.savant@mit.asia

Abstract:The Wired and wireless systems are the main technologies are implemented over the world as the technologies support easy data transactions and transmissions. There is no technology can be stated which is fully provided with all areas of security measures. Each System has its own loopholes and flaws. There is no existence of such systems which cannot be interrupted by Intruders. There may be security measures applied by the technologies but at some states they can be broken by someone. In some situations system is easy to break and in some cases it takes lots of effort for an intruder to enter in the system. There may be an uncounted loss for a group or organizations if any of their important information is theft by the unauthorized persons or their enemies. To avoid all this losses due to data theft and according activities there are many approaches presented by some inventors which can be used for better, efficient, and secure data transmissions over and after the networks. This paper is to state and provide information about one of the efficient and secured technique of data transmission called RSA algorithm.

Keywords: Euler phi function, Encryption, Decryption, Repeated square method.

1. Introduction

In the area of cryptography the RSA algorithm is considered as first of all approach of practicable approach of public-key cryptosystem and is used widely for secured data transmission amongst the public transformation channels. As it is the public-key cryptosystem it usually requires two of the keys namely public-key and private-key for its implementation. Where the public-key and private-key differs from each other in order to maintain its security metrics and the secret keys are kept secret in order to avoid explorations to vulnerabilities. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977. Clifford Cocks, an English mathematician, had developed an equivalent system in 1973, but it wasn't declassified until 1997.

As the RSA system belongs to the type asymmetric key cryptosystem the core process of Encryption and Decryption can be stated dynamically as follows :

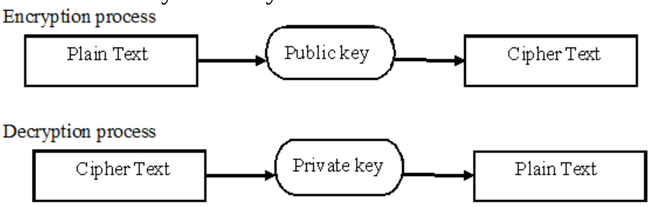

Figure 1: Encryption and Decryption Process

A user of RSA creates and then publishes a public key based on the two large prime numbers, along with an auxiliary value called secret key. The prime numbers 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. Breaking RSA encryption is known as the RSA problem. It is an open question whether it is as hard as the factoring problem [1].

2. The Risk Environment

Wireless networks and handheld devices are vulnerable to many of the same threats as conventional wired networks. Intruders who gain access to information systems via wireless communications can bypass firewall protection. Once they have accessed systems, intruders can launch denial of service attacks, steal identities, violate the privacy of legitimate users, insert viruses or malicious code, and disable operations. Sensitive information that is transmitted between two wireless devices can be intercepted and disclosed if not protected by strong encryption. Handheld devices, which are easily stolen, can reveal sensitive information [1] [2].

3. Key terms used in Implementation:

3.1 Euler phi function / Totient function :

The Euler's totient function or phi (φ) function is a very useful function to calculate the number of coprimes to each number. The totent function helps to calculate the exact number suggesting how many numbers are coprimes to the selected number. The totient $\varphi(n)$ of a positive integer n greater than 1 is defined to be the number of positive integers less than n that are coprimeto n. $\varphi(1)$ is defined to be 1. The following table shows the function values for the first several natural numbers:

n	$\varphi(n)$	numbers coprime to n
1	1	1
2	1	1
3	2	1, 2
4	2	1,3

5	4	1,2,3,4

Table 1: Calculating Coprimes to number by Euler phi function

3.1.1 **Major conditions to calculate Euler phi function:**

- **when*n* is a prime number** (e.g. 2, 3, 5, 7, 11, 13), **$\varphi(n) = n-1$**.
- **when*m* and *n* are coprime, $\varphi(m*n) = \varphi(m)*\varphi(n)$**.
- **If the prime factorisation of n is given by n $=p_1{}^{e}{}_1*...*p_n{}^{e}{}_n$, then $\varphi(n) = n *(1 - 1/p_1)* ... (1 - 1/p_n)$** [4].

3.2 Inverse of a number:

In general mathematics the inverse of a number A is calculated as 1/A.As we are calculating the multiplicative inverse of a number for implementation of RSA algorithm we have to check condition where a number x is multiplicative inverse of a number y if and only if mod of multiplication of x and y is = 1 i.e.

X * Y (mod n) = 1

Where ,Y is a number and X is a multiplicative inverse of Y.

3.3 Repeated Square method:

Repeated squaring is none other than calculating the power of any number e.g. 5 ^ 2 = 25.

As example we have taken the very small power calculation but when we are dealing with the security of the data we have idea that what sort of numbers we are going to deal with.

The calculation will be associated with 10,156,20,25 digits of number rest to the same digits.

Then it may become hectic or may be an impossible task to go with it manually. Hence to overcome the above problem the Repeated square method is used.

Following is an algorithm to implement the Repeated square method:

```
1. Set b = 1, if k = 0 then return b.
2. Set A = a.
3. If k(0) = 1 then set b = a.
4. For I from 1 to t do the following.
   4.1. Set A = A ^ 2 ( mod n ).
   4.2. If K(i)= 1 then b = A*b ( mod n ).
5. Return ( b ).
```

Where,

a is the base number.

k is the binary conversion of the power of a.

A and B are calculations to the repeated square method.

3.4 Prime number:

Prime number is a positive integer which must satisfy the condition that it can be divided by the 1 or the number itself. We use the prime numbers for cryptographic security techniques as they are secured for calculations and they are really hard to crack for the intruders.

4. Algorithm for RSA implementation:

4.1 Create Public and Private Keys

Before starting an actual processing i.e. Encryption and Decryption senders and receivers must calculate the respected values of public and secret key as:

4.1.1) select two large primes namely p and q, here we are going with p = 3 and q = 11

4.1.2) calculate n = p * q

i.e. 3 * 11 = 33

4.1.3) calculate ϕ(n) = (p - 1) * (q - 1)

i.e.(3 - 1) * (11 - 1) = 20

4.1.4) choose a prime number d, such that d is co-prime to ϕ(n). we pick d = 3.

4.1.5) So, the numbers n = 33 and d = 3 become the Server's private key.

4.1.6) Now, still done in advance of any transmission, the Server has to calculate its public key. Here is how.

4.1.7) e * d = 1 (modϕ(n))

4.1.8) e * 3 = 1 (mod 20)

i.e. e = 7

4.2 Encrypting the message

Here is the encryption function executed by the sender of message.

4.2.1 $\boldsymbol{C = M \,^\wedge\, e\,(mod\, n)}$

where,

"^" means "to the power of"

M is the Plain message we want to encrypt

e and e are Server's public key (see Section 1)

C is our Encrypted message we want to generate

After putting the values, this equation is solved as follows:

4.2.2 C = 2 ^ 7 (mod 33)

i.e 128 (mod 33)

C = 29

In this calculations we have got the Cipher text which is encrypted is C = 29 and in next section we are going to decrypt the message in order to get the reversed plain text.

4.3 . Decrypting the Message

Here is the decryption math the Server executes to recover the original Plain text

4.3.1 $\boldsymbol{M = C \,^\wedge\, d\,(\,mod\, n\,)}$

C is the Encrypted message just received

d is the Server's secret key

M is the Plain message we are trying to recover

n is Server's public key n=33 and d=7).

After putting the values this equation is solved as follows::

4.3.2 M = 29 ^ 3 (mod 33)

i.e. 24389 (mod 33)

M = 2, which is exactly the Plain text message that the Browser started with [5].

Flowchart

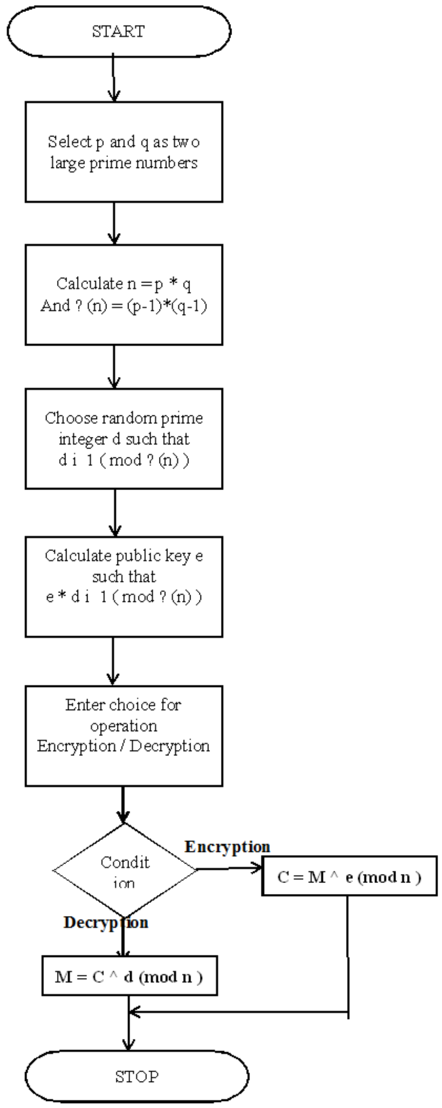

Figure 2: Flowchart for implementation of RSA

5. Conclusion

The paper is prepared and developed for the purpose of study and review of the RSA algorithm and the related techniques to calculate the Mathematics of the RSA algorithm. The algorithm is been developed and debugged with the c++ program and deployed using Turbo C 3. The paper meets its requirements as it is implemented as it was planned successfully.The algorithm is been developed and debugged with the C++ program and deployed using Code::Blocks (free, open source, cross-platform IDE which supports multiple compilers including GCC, Clang and Visual C++)

References

[1]. The RSA Algorithm Explained Using Simple Pencil and Paper Method August 14, 2003

[2]. http://www.itl.nist.gov/lab/bulletns/bltnmar03.htm

[3]. http://en.wikipedia.org/wiki/RSA_%28cryptosystem%29

[4]. *en.wikipedia.org/wiki/Euler's_totient_function*

[5]. Cryptography and network security, William Stallings.

[6] Rajan.S.Jamgekar, GeetaShantanuJoshi File Encryption and Decryption Using Secure RSA International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-1, Issue-4, February 2013

[7] B.Persis Urbana Ivy, PurshotamMandiwa.Mukesh Kumar A modified RSA cryptosystem based on 'n' prime numbers International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume1 Issue 2 Nov 2012

[8]. http://www.mathaware.org/mam/06/Kaliski.pdf

Author Profile

Nilesh K. Kajalehas completed his bachelor's degree in Computer Science and pursuing Masters of Computer Application (Final Year) from Marathwada Institute of Technology (Engineering), Aurangabad. His area of interest includes Network Security and Cryptography.

Shubhashree Savant is M.Sc. (Computer Science), MCA and M.Phil. (Computer Science) and is currently working in the capacity of Assistant Professor in Department of MCA at Marathwada Institute of Technology (Engineering), Dr. BabasahebAmbedkarMarathawada University, Aurangabad, Maharashtra. As a professional member she is associated with CSI. She has total experience of 14 years in teaching. Her area of specialization includes Medical Image Processing, Network security, Cryptography and Steganography.