Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 24 (2016) 1581 – 1586

Procedia Technology

International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015)

Dual Modulus RSA based on Jordan-Totient function

Balram Swami[a*], Ravindar Singh[b], Sanjay Choudhary[c]

[a]*Computer Science Department, Government Engineering College, Ajmer-305002, India*
[b]*Computer Science Department, Government Engineering College, Ajmer-305002, India*
[b]*Master in Computer Application Department, Government Engineering College, Ajmer-305002, India*

Abstract

A public key cryptosystem consists of a public key, which is used for encryption, and a private key, used for decryption. Therefore, anybody can encrypt plain-text since the encryption key is available to everyone, but only the holders of the private key corresponding to the public key used for encryption can decrypt the cipher-text. RSA is the most popular Asymmetric cryptosystem as it uses pair of keys, one of which is used to encrypt the data in such a way that it can only be decrypted with the other key. The keys are generated by a common process, but they cannot be feasibly generated from each other. The security of the RSA system is based on the assumption that factoring of large number is difficult. But if one could factor a large number into its prime factors then he could break the security. So a new algorithm is developed to increase the security of RSA called Dual Modulus RSA based on Jordan-Totient function (DMRJT). DMRJT algorithm is more secure as compared to RSA algorithm as it uses dual modulus based double encryption and decryption with the use of Jordan function. It is shown here that dual modules play an important role in increasing the complexity of decomposing them into its factors and Jordan function increase the size of the private key hence increases the security. DMRJT algorithm uses double encryption and decryption using double private and public keys to provide security against Brute-force attacks.

Peer-review under responsibility of the organizing committee of ICETEST – 2015

Keywords: RSA; DMRJT; Key; Jorden-Totient;

1. Introduction

Cryptography is the science that uses mathematical concepts for encoding message in non readable format. Original text is known as plain Text which is readable and after encoding the plain text the resultant is known as cipher text which is unreadable. Encryption and Decryption required a key and an algorithm to do this.

Peer-review under responsibility of the organizing committee of ICETEST – 2015
doi:10.1016/j.protcy.2016.05.143

1.1. Encryption and decryption

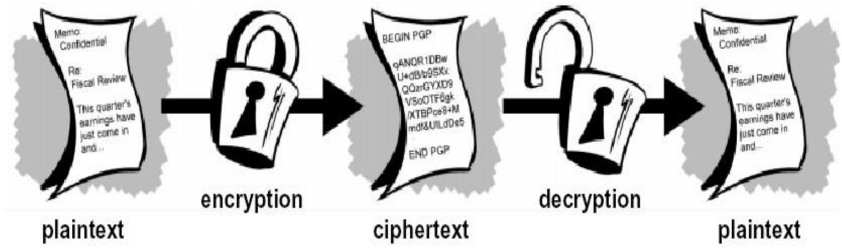

Fig. 1. Encryption and Decryption of Text

- Plain text- It is the text which is in the human readable form.
- Cipher text- It is an encoded text which is not in human readable form. Cipher text is Encoded text which cannot be easily understood by the human being
- Encryption- It is a technique to encode the plain text in some other form called cipher text with the help of a key and algorithm, which is unreadable to any unauthorized person who does not have key.
- Decryption- it is technique to convert a cipher text into plain text

There are many well known cryptography algorithms. We can divide them according to the number of keys they used and working scenario for encryption and decryption of plain text.

In this paper we are going to introduce a modified version of RSA algorithm with the use of Jorden-Totient function. Which have high security than RSA algorithm.

Section 2 will describe the types of cryptography briefly. Section 3 contains the explanation of RSA algorithm. Section 4 will be the description of encryption and decryption techniques of DMRJT and section 5 contains the comparison between the RSA and DMRJT. And last one is the conclusion of the research. And section 7 contains the references.

2. Types of cryptography

There are many type of cryptography and different techniques have different capacity, advantages, disadvantages and security.

- Brute Force Attack- Try and Error i.e try some code to decode if success then gets the original Plain Text otherwise Error is called Brute Force Attack.
- Man-in-Middle Attack- A man within the middle can hear the conversation and get some clue or key without notifying the two authorized party is called man-in-middle attack.

2.1.Substitution Techniques

In the Substitution technique replaces the characters of the plain by other characters, numbers or special symbols.

Substitution algorithms

- Caesar cipher
- Modified version of Caesar cipher
- Mono-alphabetic cipher
- Homophonic substitution cipher
- Polygram substitution cipher
- Poly-alphabetic substitution cipher
- Play fair cipher
- Hill cipher

2.2.Transposition Techniques

In this technique the characters of the plain text is replaces with some others text and some permutation also done over the plain text.

Transposition algorithms

- Rail fence technique
- Simple columnar transposition technique
- Vernam cipher (One-time pad)
- Book cipher/Running key cipher

2.3. Symmetric key cryptography

In this cryptography the encryption and decryption are done using same key. So this is a simple technique but there is problem in this known as key exchange problem. i.e. how the sender and receiver will agree on the same key for the communication and there is possibilities of man-in-middle attack when sender and receiver agree on the key someone may listen their key so the communication will not secure. So in the symmetric key cryptography uses the Deffie-Hellman key exchange algorithm.

If there are N parties involve in the communication then number of keys required

Keys = $N*(N-1)/2 = (N^2-N)/2$

- Prime numbers are those integer numbers which divided only by 1 and number itself.
  Eg. 2, 3, 5, 7, 11, 13, 17, 19

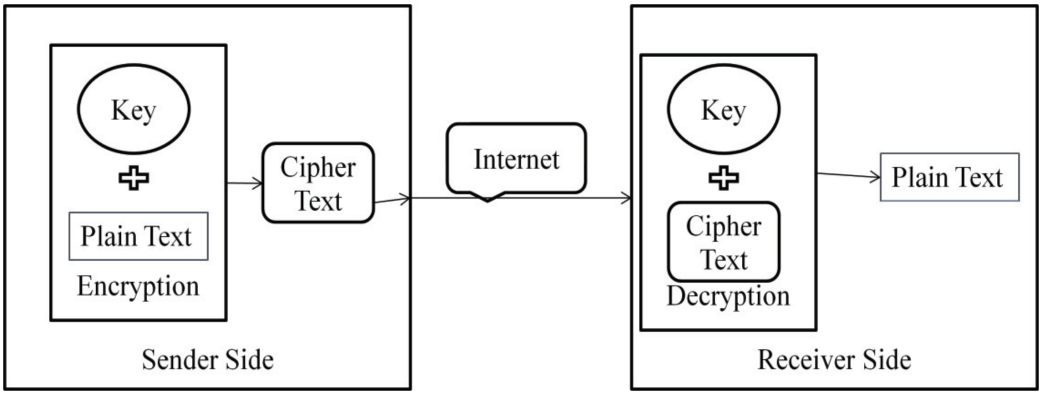

Fig. 2. Working Scenario of Symmetric key cryptographic

- *Defffie-Hellman key exchange /Agreement Algorithm*

This algorithm is for key exchange not for the cryptography. It is based on the principles of mathematics.

Algorithm

- Let John and Henry want communicate then firstly they have to agree on two large prime number say **n** and **g**. These two integers do not need to keep secret.
- John choose another large random number say **x** and calculates A with the formula: $A=g^x \bmod n$
- John sends his calculated number A to Henry
- Henry chooses independently another large number y and calculates B with the formula: $B=g^y \bmod n$
- Henry sends his calculated number B to John.
- John calculates the secret key k1 as $k1= B^x \bmod n$
- Henry calculate his secret key k2 as $k2= A^y \bmod n$

We can see that both John and Henry have same key that is $k1 = k2 = k = (g^{xy} \bmod n)$. But with this algorithm key exchange is not fully secured because man-in-middle attack can be possible.

Some popular symmetric key algorithms are:-

- DES – Data Encryption Standard

- DDES – Double Data Encryption Standard
- IDEA – International Data Encryption Algorithm
- RC4 – Ron Rivest 4
- RC5 – Ron Rivest 4
- Blowfish
- AES – Advanced Encryption Standard

2.4. Asymmetric key cryptography

In this scheme cryptography uses two kinds of keys one is “Public key” and second is “Private key”.

- Public key- it is used to encrypt the plain text. This key does not need to keep secret.
- Private key-it is used to decrypt the plain text. This key is kept secret for security reasons.

Man-in-middle attack and Brute force attack cannot possible in this type of cryptography technique.

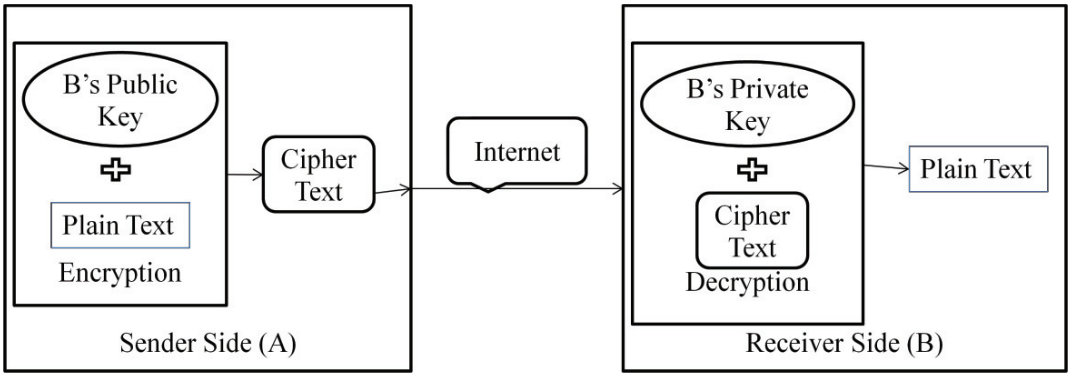

Fig. 3. Working Scenario of Asymmetric key cryptographic

Some popular Asymmetric key algorithms are

- RSA algorithm
- Digital Signatures
- Knapsack Algorithm

3. RSA algorithm

Invented by Ron **R**ivest, Adi **S**hamir and Len **A**dleman in 1977. RSA is based on theoretical work of Diffie-Hellman key exchange algorithm. It is based on the fact that it is easy to find and multiply large prime numbers but it very difficult to find the factor their product.

It is very simple algorithm but the real challenge is to generate the public key and private key which are based on large prime numbers.

Algorithm:-

- Choose two large prime numbers say P and Q
- Calculate N= P*Q
- Select the public key E such that it does not a factor of (P-1) and (Q-1)
- Select the private key D which satisfy the equation (D*E) mod (P-1)*(Q-1) = 1
- For Encryption generate the Cipher Text CT = PT^E mod N
- For Decryption generate the Plain Text PT = CT^D mod N

If we choose small prime numbers then it may be possible Brute force attack but we choose large number of prime numbers then it takes very long time.

4. DMRJT Algorithm and key generation

DMRJT stand for Dual Modulus RSA based on Jordan-Totient function. DMRJT public cryptosystem is almost same as RSA cryptosystem with some modification. As it is an asymmetrical cryptosystem, it uses two pair of keys; one key pair is used to encrypt the data in such a way that it can only be decrypted with the other key pair. Following are the processes of DMRJT cryptosystem.

4.1. Key Generation Algorithm:

- Generate large random primes, p_1, p_2 and q_1, q_2 of approximately equal size.
- Compute $n_1 = p_1$ x p_2 and $n_2 = q_1$ x q_2
- Compute ***Jordan-Totient function*** $J_s(n) =(p_1^s-1)*(p_2^s-1)$ and $J_t(n)= (q_1{}^t-1)* (q_2{}^t-1)$.
- Choose two integer e_1 and e_2 such that GCD $(e_1, J_s(n)) = 1$ and GCD $(e_2, J_t(n)) =1$.
- Compute the secret exponent d_1 and d_2, such that e_1 x d_1 mod $J_s(n) = 1$ and e_2 x d_2 mod $J_t(n) = 1$.
- The public key is (n_1, n_2, e_1, e_2) and the private key is (n_1, n_2, d_1, d_2). Keep all the values d_1, d_2, p_1, $p_{2,}$ q_1, $q_{2,}$ J_t and J_s secret.
- n_1 and n_2 are known as the *modulus.*
- e_1 and e_2 are known as the *public exponent* or *encryption exponent* or just the *exponent*.
- d_1 and d_2 are known as the *secret exponent* or *decryption exponent*.

4.2. Encryption:

Sender A does the following:-

- Obtains the recipient B's public key (n_1, n_2, e_1, e_2).
- Represents the plaintext message as a positive integer m.
- Computes the cipher text $c = ((m^{e1} \bmod n_1)^{e2} \bmod n_2)$.
- Sends the cipher text c to B.

4.3. Decryption:

Recipient B does the following:-

- Uses A’s private key (n_1, n_2, d_1, d_2) to compute $m = ((c^{d2} \bmod n_2)^{d1} \bmod n_1)$.
- Extracts the plain text/clear text from the message Representative m

5. Comparison between RSA and DMRJT

Table.1 Comparison between RSA and DMRJT

Parameters	RSA	DMRJT
Execution time	Less than DMRJT	Slightly More than RSA
Key generation time	Less than DMRJT	More than RSA because two pair of keys
Public key size effect	If size increases than time for key generation and execution time also increases	If size increases than time for key generation and execution time also increases and it takes more than RSA
Speed	Faster than DMRJT	Slower than RSA
Security	High secure if takes large prime numbers	DMRJT has double security than RSA because it has double private key
Complexity	Not so complex easy step of working	It has complex steps than RSA
Number of keys	Two keys used say public and private	Two pair of public and private keys used

6. Conclusions

Double modulus modified RSA algorithm based on Jorden-Totient function. As we comparison of RSA and DMRJT then we can see DMRJT is slower than RSA but DMRJT has high security than RSA as it uses double pair of private key and public key and also it is based on Jorden-Totient which makes it double security than RSA with some limitation like it slow down the working speed of DMRJT. If there are any third party tries to break the RSA in N combination and private key is x bit long than DMRJT will takes N^x combination.

DMRJT provide more security but there are many ways to make it faster with high security.

7. References

[1]. Atul Kahate Cryptography and Network Security Second Edition. Tata McGraw Hill Education Private Limited. Thirteenth reprint 2011.

[2]. S. Thajoddin & S. Vangipuram. A Note on Jordan’s Totients Function. Indian J. pure appl. Math. 19(12): 1156-1161, December 1988.

[3]. Harish Sharma, Dr. shiv kumar & Kavita Sharma. Dual mode public key cryptosystem. IJARCSSE Volume 8-August 2014

[4]. Mykola karpinskyy & Yaroslav Kinakh. Reliability of RSA Algorithm and its Computational Complexity. IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 8-10 September 2003.

[5]. R. Rivest, A. Shamir and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, 21 (2), February 1978, pages 120-126.