Journal of Theoretical and Applied Information Technology



www.jatit.org

# EVALUATION OF PERFORMANCE CHARACTERISTICS OF CRYPTOSYSTEM USING TEXT FILES

[1]Challa Narasimham, [2]Jayaram Pradhan

[1]Assoc. Prof., Dept of Computers, MVGR College of Engineering, Vizianagaram, India-535 005

[2]Professor, Dept of Computer Science, Berhampur University, Orissa, India.

E-mail: narasimham_c@yahoo.com, jayarampradhan@hotmail.com

**ABSTRACT**

In order to achieve the security for the e-business application, generally the organizations follow the cryptographic methods. The two widely accepted and used cryptographic methods are symmetric and asymmetric. The DES ideally belongs to the category of symmetric key cryptosystem and RSA, NTRU belongs to the category of asymmetric key cryptosystem. Symmetric key ciphers use the same key for encryption and decryption, or the key used for decryption is easily calculated from the key used for encryption. Symmetric key ciphers can be broadly grouped into block ciphers and stream ciphers. Symmetric encryption has a troublesome drawback i.e., two people who wish to exchange confidential messages must share a secret key. The key must be exchanged in a secure way. Key distribution is difficult among the parties. .RSA is one of the oldest and the most widely used public key cryptographic algorithms. It was the first algorithm known to be suitable for signing as well as encryption. The system works on two large prime numbers, from which the public and private keys will be generated. NTRU ($N^{th}$ degree truncated polynomial ring units) is a collection of mathematical algorithms based on manipulating lists of very small integers. NTRU is the first secure public key cryptosystem not based on factorization or discrete logarithmic problems. The keys are generated by having small potent polynomials from the ring of truncated polynomials. Finally we proceed with the key generation, encryption and decryption of the plain text required by implementing the algorithms of both the cryptosystems. Though the limitations are there in the symmetric key cryptosystem, we proposed and tested all the methods for variable sized text files. This paper presents the comparative study of DES, RSA and NTRU algorithms for variable sized text files as input and the results were observed, analyzed and compared so as to identify which method is appropriate to the business needs.

**Keywords**: Asymmetric Key, Data Encryption Standard (DES), $N^{th}$- Degree Truncated Polynomial Ring Unit(NTRU), Rivest- Shamir- Adleman(RSA), Symmetric key.

## 1. INTRODUCTION

The symmetric key cryptography uses conventional encryption algorithms. Although numerous conventional encryption algorithms have been developed since the introduction of DES, the Data encryption Standard (DES) remains the most widely used conventional encryption algorithm. Usually the structures of conventional encryption algorithms are very complex and cannot be explained as easily as asymmetric encryption algorithms. Usually all the symmetric encryption algorithms follow block cipher mode of operation. The other type of cryptography i.e., public key cryptography is the greatest and perhaps the only true revolution in the entire history of Cryptography. Public-Key cryptography provides a radical departure from all that has gone before. The two major reasons which made Public-Key cryptographic algorithms more reliable in the areas of confidentiality, key distribution and authentication. These algorithms are based on mathematical calculations rather than substitutions and permutations like the symmetric

cryptosystems. These algorithms use two keys in contrast to symmetric algorithms which uses only one key. These public key cryptosystems evolved from an attempt to attack two of the most difficult problems of conventional encryption, one being the problem of Key distribution and the other problem was associated with the digital signatures for the purpose of authenticity of data and messages. Public-Key algorithms rely on one key for encryption and a different but related key for decryption. It is computationally infeasible to determine the decryption key given only the knowledge of cryptographic algorithm and the encryption key. The two keys in Public-Key Cryptographic algorithms are referred as public key and private key. Invariably the private key is kept secret and is only known to the user that holds it. The two most important public key cryptographic algorithms are the RSA and NTRU which have been accepted and are widely used now-a-days. In the next sections, we presented the implementations of DES, RSA and NTRU systems for different text files and finally compared the computational running times to find the suitable method for the business applications.

## 2. PERFORMANCE EVALUATION OF ASYMMETRIC KEY CRYPTOSYSTEMS

### 2.1 RSA

RSA scheme is a block cipher in which the plain text and cipher text are integers between 0 and n-1 for some n. That is, the block size must be less than or equal to $\log_2(n)$; in practice, the block size is $2^k$ bits, where $2^k < n <= 2^{k+1}$. Encryption and Decryption are of the following form, for some plain text M and cipher text $C = M^e \bmod n$
$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$.
Both the sender and the receiver must know the value of n. The sender knows the value of e, and only the receiver knows the value of d. Thus, this is a public key encryption algorithm. The public key consists of n, the modulus, and e, the public exponent. The private key consists of n, the modulus, which is public and appears in the public key, and d, the private exponent, which must be kept secret. We are now ready to state the RSA scheme. The following are the steps to generate the public and the private keys. Choose two large prime numbers p, q such that p is not equal to q, randomly and independently of each other.

Compute n = p * q
Compute the quotient Ø (n) = (p-1) (q-1)
Choose an integer e such that 1< e <Ø (n) which is co prime to Ø (n)
Compute d such that de = 1 (mod Ø (n))
Finding the large prime numbers is usually done by testing random numbers of the right size with probabilistic primality tests which quickly eliminate virtually all non-primes. p and q should not be 'too close', lest the Fermat factorization for n be successful. Further more if p-1 and q-1 has only small prime factors, n can be factored quickly and these values of p and q should therefore be discarded as well. It is important that the secret private key d should be large enough.

#### 2.1.1 RSA Encryption

RSA is a block cipher mechanism. So we divide the input binary text into 8 bit apart. We will convert the first 8 bit text into an integer form. After that we take a public key from key generator and perform encryption operation for that integer. For example 'M' is an integer then we encrypt 'M' by performing $C=M^e \bmod n$. After calculating the value of C we will convert C into binary format. After that we will make binary value of C as 16 bit length and print that result in ciph.txt. Now we will take another 8 bit text and repeat the above process.

#### **2.1.2** RSA Decryption

Divide the input binary text into 16 bit apart. We have converted the first 16 bit text into an integer form. After that we take a private key'd' from key generator and perform decryption operation for that integer. For example 'C' is an integer then we encrypt 'C' by performing $M=C^d \bmod n$.

#### 2.1.3 RSA Performance

Encryption key size: 22 bits
Decryption key size: 10 bits

TABLE 1
RSA encryption and decryption methods computational execution timings in sec.,

| Text Size | Encryption | Decryption |
|---|---|---|
| 128 bits | 0.0549 | 0.0549 |
| 256 bits | 0.1098 | 0.0549 |
| 512 bits | 0.2197 | 0.1648 |
| 1K | 0.3846 | 0.3296 |
| 2K | 0.7142 | 0.6593 |
| 5K | 1.7032 | 1.7032 |
| 10K | 3.402 | 3.402 |

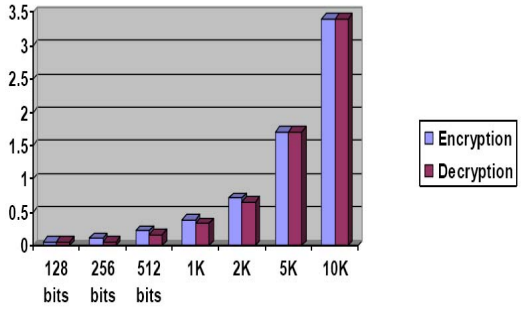



Fig 1 Performance on encryption and decryption timings of RSA

2.2 NTRU

NTRU is a collection of mathematical algorithms based on manipulating lists of very small integers. This allows NTRU to achieve high speeds with the use of minimal computing power. NTRU is the first public key cryptosystem not based on factorization or discrete logarithmic problems. Encryption and Decryption take speeds of O (n log(n)). This is compared with RSA's O ($n^3$) operations. NTRU is "non trivial ring units" or "n $^{th}$ degree truncated polynomial ring units "or "Number Theory Research Units". The basic collection of objects used by the NTRU is the ring R that consists of all truncated polynomials of degree N-1 integer coefficients. NTRU is the latest in the line of Public Key Cryptographic Systems. It is relatively new and was conceived by Jeffrey Hoffstein, Jill Pipher and Joseph. H. Silverman. NTRU uses polynomial algebra combined with the clustering principle based on elementary mathematical theory. The security of NTRU comes from the interaction of polynomial mixing modulo two relatively prime numbers.

2.2.1 NTRU Key Generation

User B wants to create a public/private key pair for the NTRU PKCS. B first randomly chooses two small polynomials f and g in the ring of truncated polynomials R. A small polynomial is small relative to a random polynomial mod q. In a random polynomial the coefficients are much smaller than q. B must keep the values of the polynomials f and g private, since anyone who knows the value of either of them will be able to decrypt messages sent to B. B's next step is to compute the inverse of the f modulo q and the inverse of f modulo p. Thus B computes polynomial fp and fq with the property that f*fq = 1 (modulo q) and f*fp = 1 (modulo p). If by some chance if the inverse does not exist, B will need to go back and choose another f. For information about computing inverses in the ring of truncated polynomials, now B computes the product h = p fq*g (modulo q). B's private key is the pair of polynomials f and fp. B's public key is the polynomial h.The CreateKey function shown in algorithm

1.Creating the inverse polynomial of the secret key modulo q, Fq,
2.Creating the inverse polynomial of the secret key modulo p, Fp,
3.Creating the Public Key, h = p * ((Fq )*g)mod q.

2.2.2 NTRU Encryption

User A has a message to transmit to B , So A first puts the message in the form of a polynomial m whose coefficients is chosen modulo p say between as –p/2 and p/2. Next A randomly chooses another small polynomial r. This is the binding value which is used to obscure the message. A uses the message m, randomly chosen polynomial r, and B's public key h to compute the polynomial e = r*h + m (modulo q). The polynomial e is the encrypted message which A sends to B.

1. Performing the polynomial multiplication of h * r and
2. Adding the message m and the modulo reduction is performed by extracting the lower w bits.

2.2.3 NTRU Decryption

User B has received A's encrypted message e and B wants to decrypt it. B begins by using the private polynomial f to compute the polynomial a = f*e (mod q). Since B is computing a modulo q, can choose the coefficients of a to lie between -q/2 and q/2. In general B will choose the coefficients of a to lie in an interval of length q. The specific interval depends on the form of the small polynomials. It is very important that B does this before performing the next step. B next computes the polynomial b = a (mod p). That is, B reduces each of the coefficients of a modulo p. Finally B uses the other private polynomial fp to compute c = fp*b (modulo p). The polynomial c will be A's original message m.

The decryption procedure is executed by the following three steps

1. Performing the polynomial multiplication of a = f * e mod q,
2. Shifting the coefficients of a into the range (-q/2; q/2).

3. Performing the polynomial multiplication of d = a * Fp mod p.

2.2.4 NTRU Performance

Encryption key size : 51 bits
Decryption key size : 20 bits

TABLE 2
NTRU encryption and decryption methods computational execution timings in secs.,

| Text size | Encryption | Decryption |
|---|---|---|
| 128 bits | 0.0000001 | 0.0000001 |
| 256 bits | 0.0000001 | 0.05490 |
| 512 bits | 0.05494 | 0.05490 |
| 1K | 0.10989 | 0.05490 |
| 2K | 0.27472 | 0.05490 |
| 5K | 0.65934 | 0.16484 |
| 10K | 1.31868 | 0.36100 |

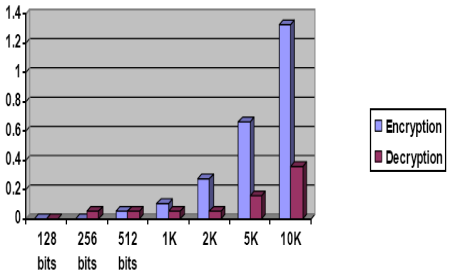



Fig 2 Performance on encryption and decryption timings of NTRU

## 3. SYMMETRIC KEY CRYPTOSYSTEM DATA ENCRYPTION STANDARD (DES)

Key size: 56 bits

TABLE 3
DES encryption and decryption methods computational execution timings in sec.,

| Text size | Encryption | Decryption |
|---|---|---|
| 128 bits | 0.054945 | 0.00001 |
| 256 bits | 0.054946 | 0.00001 |
| 512 bits | 0.070976 | 0.00052 |
| 1K | 0.1418 | 0.0010 |
| 2K | 0.2835 | 0.0020 |
| 5K | 0.6816 | 0.0084 |
| 10K | 1.3601 | 0.0142 |

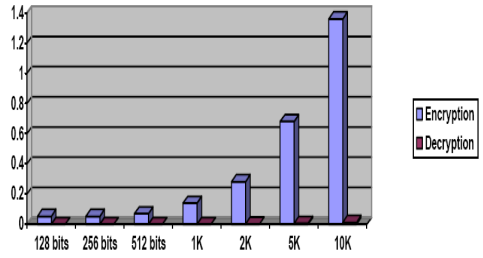



Fig 3 Performance on encryption and decryption timings of DES

## 4. OBSERVATIONS & ANALYSIS

4.1 Encryption Analysis DES, RSA, NTRU

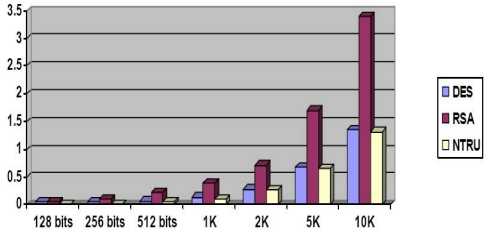
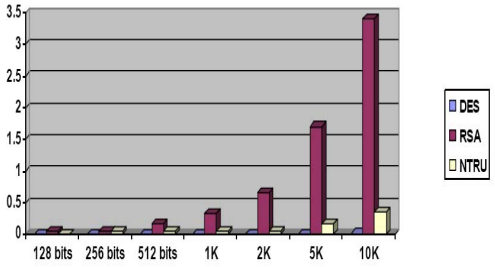



Fig 4 Performance analysis on encryption for DES, RSA and NTRU methods

4.2 Decryption Analysis DES, RSA, NTRU

Fig 5 Performance analysis on decryption for DES, RSA and NTRU methods




## 5. DISCUSSION

Performance analysis and comparison of Symmetric and Asymmetric key cryptosystems

| Method | DES | RSA | NTRU |
|---|---|---|---|
| Approach | Symmetric | Asymmetric | Asymmetric |
| Encryption | Faster | Slow | Fastest |
| Decryption | Fastest | Slow | Faster |
| Key Distribution | Difficult | Easy | Easy |
| Complexity | O(Log N) | $O(N^{3})$ | O(N Log N) |
| Security | Moderate | Highest | High |
| Nature | Closed | Open | Open |

## 6. CONCLUSIONS

Using publicly available cryptographic methods, we performed the performance comparison for variable sized text files as input. An analysis on computational running times results in significant difference among the methods. We believe in that the performance of DES, especially in decryption method is very high than the alternatives. Despite the key distribution, DES is more suitable to the application, which has the decryption as the highest priority. There is no doubt that, an Asymmetric key cryptographic system provides high security in all ways. In this paper, we proposed and performed the test cases on the two PKCS methods i.e., RSA and NTRU Though the encryption, decryption and complexity are high in NTRU, the RSA provides the highest security to the business application. We presented all these parameters with computational running times for all the methods, so as to select the appropriate method.

## 7. REFERENCES

[1] Whitefield Diffie, Martin E Hellman " New directions in Cryptography " IEEE Information theory , June 23-25 , 1975.
[2] Joffrey Hoffstein , Jill Pipher , Joseph H Silverman " NTRU – A ring based public key cryptosystem".
[3] Joffrey Hoffstein , Joseph H Silverman " Optimizations for NTRU"
[4] Collen Marie O'Rourke " Efficient NTRU implementations"
[5] Wikipedia , the free encyclopedia " NTRU Cryptosystems Inc.,"
[6] A. Huffman, "A method for the construction of minimum redundancy codes," Proc. IRE, vol. 40, pp. 1098–1101, Sept. 1952.
[7] R.L.Rivest , A.Shamir, L.Adleman "A method for obtaining digital signatures and Public-Key Cryptosystems".
[8] www.ntru.com
[9] DI management - RSA Algorithm