Analysis and Performance Characteristics of Cryptosystem using Image Files

A. Naresh Reddy
M.Tech Scholar,
Sri Vasavi Engineering College
Tadepalligudem

Rakesh Nayak
Assoc. Professor Department of IT
Sri Vasavi Engineering College, Tadepalligudem

S. Baboo, Ph.D
Reader, HOD, Department of Computer Science,
Sambalpur University, Odisha.

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 RSA and NTRU belong to the category of asymmetric key cryptosystem. 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) is a collection of mathematical algorithms based on manipulating lists of very small Integers. NTRU is the first secure public key cryptosystem. 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 image required by implementing the algorithms of the asymmetric key cryptosystems. This paper presents the comparative study of RSA and NTRU algorithms for images 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, NTRU, RSA.

1. INTRODUCTION
The two major reasons which made Public-Key cryptographic algorithms more reliable are the areas of greater confidentiality, ease of key generation and authentication. These algorithms are based on mathematical calculations rather than substitutions and permutations like the symmetric cryptosystems. Further these algorithms use two keys in contrast to symmetric algorithms which uses only one key. Public-Key algorithms rely on one key for encryption and a different but related unique key for decryption. It is computationally infeasible to determine the decryption key given only the knowledge of cryptographic algorithm and the encryption 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. 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 RSA and NTRU systems for different image files and finally compared the computational running times to find the suitable method for the business applications.

2. INTRODUCTION TO RSA AND NTRU 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 $de = 1 \pmod{\Ø(n)}$ Finding the large prime numbers is usually done by testing random numbers of the right size with probabilistic primarily 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 $C=M^e$ mod 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 $M=C^d$ mod n.

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 = pfq*g (modulo q). B's private key is the pair of polynomials f and fp. B's public key is the polynomial h.

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.

Performing the polynomial multiplication of h* r and adding the message m then taking modulo q.

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 Buses 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, shifting the coefficients of a into the range (-q/2; q/2).
2. Performing b= a mod p, and shifting the coefficients of a into the range (-p/2; p/2).
3. Performing the polynomial multiplication of c = a * fp mod p.

3. APPROACH AND PERFORMANCE

The proposed System paper presents the comparative study of RSA and NTRU algorithms for image files as input and the results were observed, analyzed and

compared so as to identify which method is appropriate to the business needs.

Block Diagram of Encryption and Decryption Process:

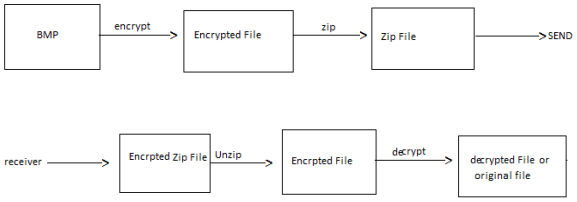


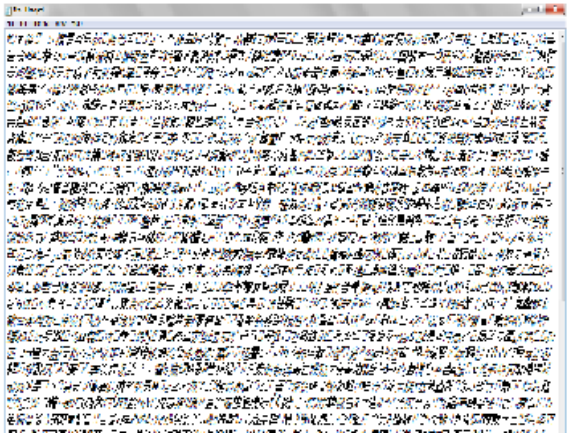
Fig.1: **Encryption and Decryption Process**:

Here we are taking image as a input file after we are converting that image file into BMP formant. And we are encrypting that image by using public key crypto system algorithm and here we get encrypted file as a output. We zip that encrypted file and sender send that file to receiver. In receiver system first system unzip that file after by using same public key crypto system we decrypt the encrypted file here we get original file as an output.

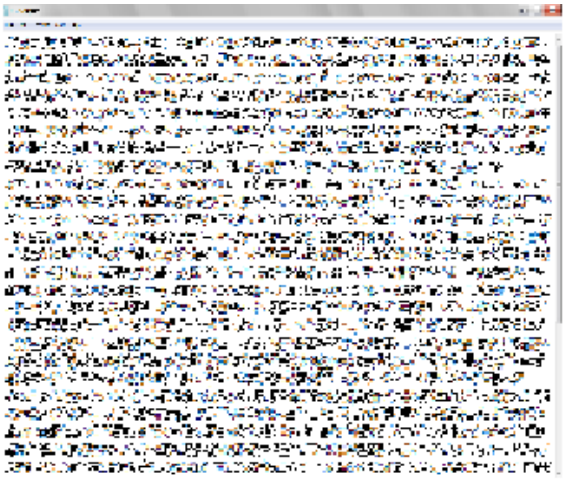
Virginal 5KB Image File:

Fig2: virginal image

Encrypted and Zipped Data of Image File by Using RSA Algorithm:



Encrypted and Zipped Data of Image File by Using NTRU Algorithm:



3.1 Performance with RSA

3.1.1 RSA encryption and decryption methods Computational execution timings in nano sec.

Image Size	Encryption	Decryption
5KB	0.00028	0.000255
50KB	0.00475	0.00226
100KB	0.0048	0.00232
300KB	0.0051	0.00241
500KB	0.00552	0.00257
800KB	0.00584	0.00261
1MB	0.00609	0.00273

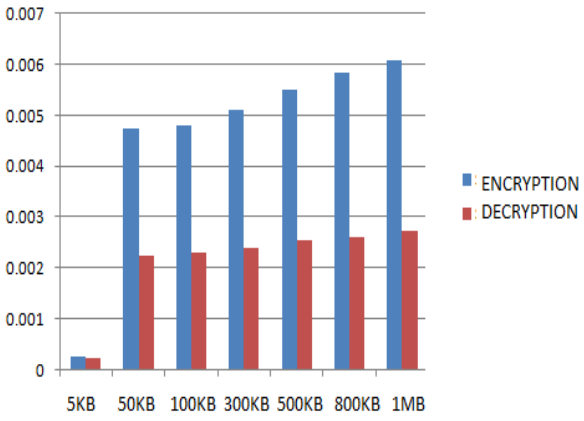


Fig3: Performance on encryption and decryption Timings of RSA

3.2 Performance with NTRU

3.1.2 NTRU encryption and decryption methods Computational execution timings in nano sec.,

Image Size	Encryption	Decryption
5KB	0.000266	0.0001
50KB	0.000291	0.000254
100KB	0.000305	0.000257
300KB	0.000322	0.000261
500KB	0.000375	0.000267
800KB	0.000458	0.00028
1MB	0.000497	0.000315

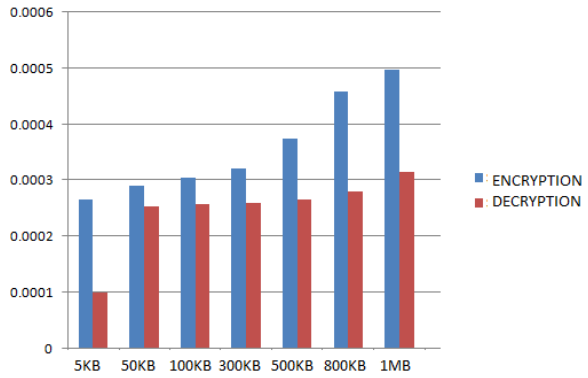


Fig 4: Performance on encryption and decryption Timings of NTRU

4. OBSERVATIONS & ANALYSIS

4.1 Encryption Analysis RSA, NTRU

METHOD		RSA	NTRU
SIZE	5KB	0.00028	0.000266
	50KB	0.00475	0.000291
	100KB	0.00480	0.000305
	300KB	0.00510	0.000322
	500KB	0.00552	0.000375
	800KB	0.00584	0.000458
	1MB	0.00609	0.000497

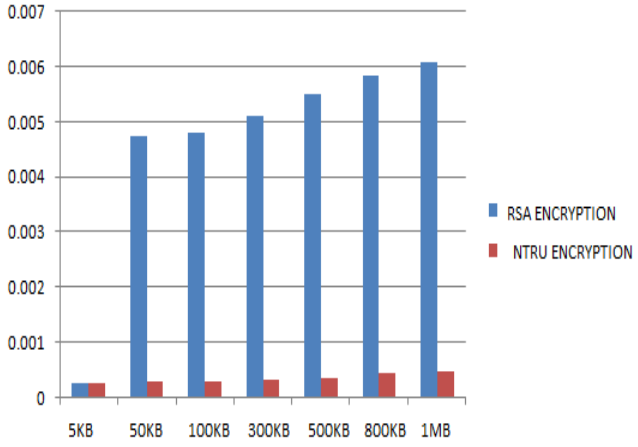


Fig 5: Encryption Analysis RSA and NTRU

4.2 Decryption Analysis RSA, NTRU

METHOD		RSA	NTRU
SIZE	5KB	0.000255	0.0001
	50KB	0.00226	0.000254
	100KB	0.00232	0.000257
	300KB	0.00241	0.000261
	500KB	0.00257	0.000267
	800KB	0.00261	0.000280
	1MB	0.00273	0.000315

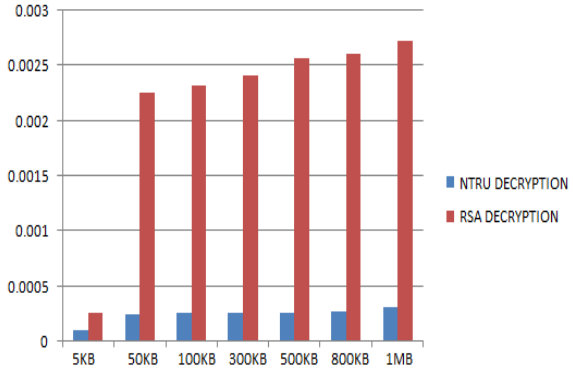


Fig 6: Decryption Analysis RSA and NTRU

5. DISCUSSION

Performance analysis and comparison of Asymmetric key cryptosystems:

METHOD	RSA	NTRU
Approach	Asymmetric	Asymmetric
Encryption	Slow	Faster
Decryption	Slow	Faster
Complexity	O(N^3)	O(N Log N)
Security	Highest	High
Nature	Open	Open

6. CONCLUSIONS

Using publicly available cryptographic methods, we performed the performance comparison for image as input. An analysis on computational running times results in significant difference among the methods. In this paper, we proposed and performed the test cases on the two public key crypto system 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.

REFERENCES

[1] Whitefield Diffie, Martin E Hellman " New directions in Cryptography " IEEE Information theory , June 2325, 1975.

[2] Joffrey Hoffstein , Jill Pipher , Joseph HSilverman " 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

[10] Challa Narasimham Jayaram Pradhan Evalution Of Performance Characterstictis of Cryptosystem Using Text Files1

[11] Rakesh Nayak, C.V.Sastry, Jayaram Pradhan," An algorithmic Comparison between Polynomial base and Matrix based NTRU cryptosystem "International Journal of Computer and Network Security, (IJCNS) Vol. 2, No. 7, July 2010.

[12] Jeffry Hoffstein, Jill Pipher and Joseph H. Silverman "NTRUA High Speed Public Key Cryptosystem", PrePrint Presented AHe Hump Session Of Euro Crypt 96,1996