J. of Comp. and I.T. Vol. 3(1&2), 1-6 (2012).

Comparative Analysis for Performance acceleration of Modern Asymmetric Crypto Systems

RAJ KUMAR[1] and V.K. SARASWAT[2]

[1,2]Department of Computer Science, ICIS
Dr. B.R. Ambedkar University, Agra (INDIA)

(Acceptance Date 2nd February, 2012)

Abstract

The demand of secure communication is growing day by day. e-commerce and mobile applications over internet has attracted the investigators of public-key cryptography to ensure the secured transaction[2]. RSA has been favorable public key cryptosystem over last 25 years. But the need of enhanced key size and possibility of attacks on RSA to break-down the security has forced the cryptographer to invent alternative to RSA. This paper introduces NTRU as an alternate algorithm to RSA and investigate for the accelerated asymmetric cryptosystem[1,5].

Key words : One way cipher, modular arithmetic, polynomial rings, encryption, decryption, cryptography

1. Introduction

Rapid progress in computer based communication technology has promoted new dimension to information security. Number theory of mathematics has turned out to be the most useful methodology in devising trusted communication system[2]. Sensitive data on web such as credit card details can be protected by encrypting the data.

The process of encryption becomes very slow when data size is large. In this case the security becomes a vexing, costly and complex business[3]. The need of faster crypto-system has been the demand of many applications such as e-mail, online baking etc.

The vast majority of products and standards use asymmetric cryptosystem. It is based on two different keys to perform encryption and decryption[4].

The present paper describes the performance of classical cryptosystem RSA which has been used in network security application as digital signature[5]. But RSA has several drawbacks along with the case of huge data size[4,9].

The further need of larger bit length keys for secured RSA forced the cryptographer to implement some new model of crypto-system[11,12].

Modern cryptographer use polynomial rings to implement asymmetric cipher rather than congruency, factoring, modulas and exponential, as used in RSA[6]. Such ciphers are called NTRU public key cryptosystem. To encrypt and decrypt a message of block length N, NTRU only required O (N^2) time where as the cipher like RSA requires O (N^3) time[7-9]. Further NTRU uses a very short key size of O (N) and requires less memory to be implemented in devices like smart cards[10].

The present paper gives the performance analysis of NTRU cryptosystem along with RSA. We also conclude the utility of NTRU mechanism in developing future day's secured wireless communication such as MANET and VANET.

2. Organization :

The research work of this paper has been organized as following. In section 3 we have derived the basic working of the classical asymmetric cryptosystem RSA. We have also enumerated the possible attacks on RSA to make it breakable and hence un-secured. In section 4, we have represented the model of modern asymmetric cryptosystem NTRU and explain the working of NTRU algorithm. In section 5, we have concluded the comparative result of encryption and decryption of NTRU along with RSA on variable data size and with different key length.

The section 6 of the present paper summarize the out comes as conclusion and further scope of development of the ciphers.

3. The RSA Cryptosystem :

Algorithm of RSA cryptosystem is based on generation of large prime number, multiplication and factoring computation of larger number.

The trust ability of RSA is higher as it is very time consuming to find the prime factors of 1024 bit or 2048 bit longer number.

3.1 Mathematical facts and conjecture used in RSA asymmetric cryptosystem.

Fact 1. Prime number generation is easy

Prime number of any size are very common and it is easy to test if a number even very large is prime.

Fact 2. Multiplication is easy :

RSA starts with computation of N which is the product of two longer selected prime.

Fact 3. Factorization is hard :

For hundred years it has been a complex operation to find the prime factors p, q for a given number n.

Fact 4. Modular exponentiation is easy :

For given m, n and e one can compute $c = m^e$ mod n.

Normally e^m mod n is result of multiplying e copies of m, and dividing by n and getting the residue.

Fact 5. Reverse of modular exponentiation is also easy:

The value of m can be recovered from c by a modular exponentiation operation using another odd integer d, thus m = $(m^e)^d$, thus

m = $(m^e)^d$ mod n

Conjecture 6. Modular root extraction is as had as NP problem.

3.2 Key generation by RSA

To encrypt a message by RSA, one requires public key. It consists a pair number (n, e) called modulus and public exponent. Similarly to decrypt the message one use another number pair (n, d) known as private key. Private key consists modulus and private exponent.

3.3 RSA Algorithm :

Following is the algorithm used to encrypt and decrypt a message with RSA[5,8].

1. Consider a pair of large primes p, q.
2. Compute modulas n = p x q
3. Select an odd public exponent e between 3 to n – 1 which is relatively prime to (p - 1) and (q - 1).
4. Compute the private exponent d such that d. e mod n $\cong 1$
5. The result (n, e) is public key and (n,d) is the private key.
6. Encryption operation is
   C = encrypt (message, m)
   i.e. C= m^e mod n where C is called cipher text
7. The decryption operation is
   m = decrypt (c)
   m = c^d mod n

4. Model of NTRU cryptosystem :

NTRU algorithm performs the encryption and decryption of the message, designed with following components[6,10,11].

(i) ***Key generation :*** It is responsible to generate the public and private keys.

(ii) ***Encryption :*** It convert the plain message into cipher text.

(iii) ***Decryption :*** It reproduce the cipher text into original plain text.

(iv) ***Polynomial operation :*** It perform mathematical operation such as multiplication and inversion.

(v) ***Random Polynomial generator :*** It is used to construct random polynomial.

(vi) ***Analyzer :*** It contains text routines.

(vii) NTRU algorithm uses polynomial addition and multiplication in the ring R, where R is $Z[x]/(x^n - 1)$. Any polynomial f in ring is written as a vector
f = $[F_0, F_1, F_2 ......]$

$$f = \sum_{i=0}^{n-1} fi.x^i$$

The addition used by NTRU is a regular polynomial and the multiplication is a cyclic convolution product denoted by $\otimes$

$H = F \otimes G$

The NTRU process uses four sets Lf, Lg, Lr and Lm of polynomials of degree N – 1 with integer coefficients.

4.1 Working of NTRU

4.1.1 Key generation:

Two small polynomials f and g are randomly chosen from the set Lf and Lg. The inverses of polynomial f modulo p and modulo q are denoted as Fp and Fq

where $Fp \otimes f \cong i \bmod p$

and $Fq \otimes f \cong 1 \bmod q$

The polynomials f and Fp are used as the private key and the polynomial h given below becomes the public key.

$h = p \otimes Fq\, g \bmod q$

4.1.2 Encryption Process :

The message 'm' must be formed as a polynomial from set Lm. To encrypt 'm', a random polynomial 'r' is chosen from the set Lr. Then encrypted message is computed as followed :

$e = r \otimes h + m \pmod q$ where

h is the public key.

4.1.3 The Decryption process :

To decrypt the cipher message 'e', the polynomial 'a' is computed as below

$a \cong f \otimes e \pmod q$

The coefficient of the polynomial 'a' is chosen from [-q/2, q/2]. The original message is computed as following.

$m = Fp \otimes a \pmod q$

4.1.4 Performance Enhancement of NTRU:

Following are the facts to enhance the performance of NTRU over other classical asymmetric cryptosystem like RSA.

Fact 1 : Invertibility of f modulo p :

The polynomial f can be chosen as $f = 1 + p \otimes f1$ where f1 is a random polynomial. Thus we need not to compute Fp in key creation and at the second multiplication in decryption.

Fact 2 : Consider p to be a polynomial :

We can consider 'p' to be a polynomial rather than an integer, eg. the convenient polynomial is a "small" polynomial like p=x+2

This makes easy to encrypt the message.

Fact 3 : Low hamming weight products:

We can make the process of encryption and decryption faster by using small hamming weight products. As we have observed the products $r \otimes h$ and $e \otimes f$ are required to do encryption and decryption which can be made faster by using "small" polynomial.

5. Performance comparison of NTRU and RSA :

The following is the table to do the comparison of performance acceleration of NTRU with classical RSA[12,15].

Table 5.1. Performance comparison of RSA with NTRU

System	**Basic operation**	**Order of complexity**	
		Encryption Decryption	Key generation
RSA	Modular Multiplication	O (N^3)	O (N^2)
NTRU	Convolution Product	O (NlogN)	O (N)

The following table and plot of graph gives more clear idea about the performance acceleration of RSA and NTRU crypto-system[14,15].

Table 5.2. Execution time of NTRU with message size (source internet)

Text size	**Encryption**	**Decryption**
128 bits	0.0000001	0.0000001
256 bits	0.0000001	0.05490
512 bits	0.05494	0.05494
1 K	0.10989	0.05494
2 K	0.27472	0.05494
5 K	0.65934	0.16484
10 K	0.311868	0.36100

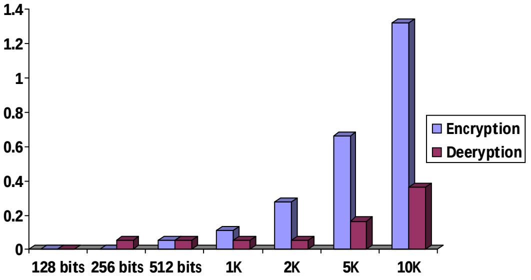

Fig. 5.1. Performance of NTRU over message size

Table 5.3. Execution time of RSA with message size (source internet)

Text size	**Encryption**	**Decryption**
128 bits	0.054945	0.0000001
256 bits	0.054946	0.0000001
512 bits	0.070976	0.00052
1 K	0.1418	0.0010
2 K	0.2835	0.0020
5 K	0.6816	0.0084
10 K	0.3601	0.0142

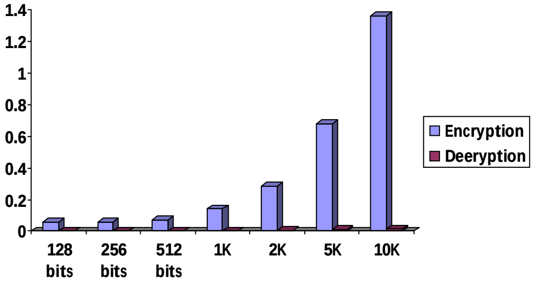

Fig. 5.2. Performance of RSA over message size

6. Conclusion

Performance analysis and comparison of RSA with NTRU is derived as following[9,13,15].

Parameter	RSA	NTRU
Approach	Asymmetric	Asymmetric
Encryption	Slow	Faster
Decryption	Slow	Fast
Key Distribution	Easy	Easy
Security	Higher	High
Nature	Open	Open

We can deduce from this paper that encryption and decryption is very faster in NTRU than RSA, but security of NTRU is little moderate than RSA while using the key of same size.

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 implémentations"
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. N. Ferguson, R. Schroeppel, D. Whiting, "A simple algebraic representation of Rijndael ", Selected Areas in Cryptography, Proc. SAC 2001, *Lecture Notes in Computer Science* 2259, pp. 103–111, Springer Verlag (2001).
11. K. Aoki and H. Lipmaa, "Fast Implementations of AES Candidates", Third Advanced Encryption Standard Candidate Conference, pages 106–120 (2000).
12. H. Lipmaa, Fast Implementations of AES and IDEA fro Pentium 3 and 4, October 2005, http://home.cyber.ee/helger/implementations.
13. A. Hodjat, I. Verbauwhede, "A 21.54 Gbit/s fully pipelined AES processor on FPGA", Field–Programmable Custom Computing Machines 2004 (FCCM'04), 12th Annual IEEE Symposium, pages 308 – 309.
14. B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall and N. Ferguson, "Performance Comparison of the AES Submissions", Proc. Second AES Candidate Conference, NIST, pp. 15-34 (1999).
15. A. Lenstra, Key Length, Contribution to "The Handbook of Information Security", http://cm.bell-labs.com/who/akl/key_lengths.pdf (2004).