Volume 2, Issue 9, September 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Improvement of NTRU Cryptosystem

Ranjeet Ranjan* *Department of SoICT Gautam Buddha University India*
Dr. A. S. Baghel *Departmentof SoICT Gautam Buddha University India*
Sushil Kumar *Departmentof SoICT Gautam Buddha University India*

Abstract— The NTRU cryptosystem can be used in a range of application which involves security in a network. NTRU is depending upon the algebraic structures of certain polynomial rings. The NTRU Encrypt is a public-key cryptosystem which is based on the shortest vector problem. Its main characteristics are the low memory and computational requirements as providing a high security level. It is a very well-organized public-key cryptosystem based on polynomial arithmetic. In this research paper, we introduce the description of NTRU cryptosystem, its analysis and some improvement for the security.

Keywords— modulo, convolution operation, NTRU cryptosystem, public key cryptography, polynomial.

I. INTRODUCTION

The improvement of public-key cryptography (PKC) is the best and true revolution in the whole history of cryptography. PKC gives a fundamental different approach from all that has security to securing network. For individual thing about the public-key algorithms are depended upon the mathematical functions rather than on permutation and substitution. The most important, PKC is asymmetric encryption in which encryption and decryption are performed using the different keys. In symmetric encryption is a different form of cryptosystem in which encryption and decryption are performed using the same key. The use of encryption and decryption keys has profound outcome in the field of authentication, confidentiality and key distribution. The NTRU is an acronym for N-th degree truncated polynomial ring which was simply called NTRU. It was developed in 1996 by three mathematicians J. Hoffstein, J.H. Silverman and J.Pipher. These mathematicians in contact with each other with D. Lieman founded the NTRU Cryptosystems, Inc. and were granted a patent on the cryptosystem. NTRU is relatively new cryptosystem [1][2]. The mathematics which used in NTRU is based on lattice-based cryptography it has different cryptographic properties from RSA and ECC[3]. The strength of cryptographic NTRU performs valuable private key operations much faster in comparison to RSA. NTRU's comparative performance increases with the level of security necessary. While in each case N represents a natural block size, RSA etc. require $O(N^3)$ where as NTRU requires $O(N^2)$ for encryption and decryption. Though NTRU also has a small block size O(N) and low memory requirements which makes it ideal. Speed and quantum computing resistance are two characteristics of NTRU that make it interesting as an alternative to RSA and Elliptic Curve Cryptography.

II. NTRU DESCRIPTION

NTRU is based on the algebraic composition of definite polynomial rings therefore it provides very fast computation to encrypt and decrypt the message, NTRU only requires $O(N^2)$. It has a following notation which is a part of parameter of NTRU implementation.

Where n is the dimension of the polynomial ring, p is a positive integer define a ring Z/pZ, q is a positive integer define a ring Z/qZ it is used in creating a public key, k is a security parameter, d_f is distribution of the coefficient of polynomial f, d_g is distribution of the coefficient of polynomial g, d_r is the number of 1s and -1s that use in certain random polynomial.

There are some other notations which we will use:

f : polynomial in $Z[X]/(x^n-1)$
f_p : polynomial in $Z[X]/(p, X^n-1)$
f_q : polynomial in $Z[X]/(q, X^n-1)$
L_f : set of polynomial in $Z[X]/(x^n-1)$ whose satisfy d_f
g : polynomial in $Z[X]/(q, X^n-1)$
L_g : set of polynomial in $Z[X]/(x^n-1)$ whose satisfy d_g
L_r : set of polynomial in $Z[X]/(x^n-1)$ whose satisfy d_r
f_p^{-1} : inverse of fp in $Z[X]/(p, X^n-1)$
f_q^{-1} : inverse of fq in $Z[X]/(q, X^n-1)$
h : public key, in $Z[X]/(q, X^n-1)$
H : hashing function
G : generating function

r : polynomial in $Z[X]/(q, X^n-1)$
the ring $R=Z[X]/(x^n-1)$
where $f \in R$ will be as a polynomial,

$$f = \sum_{i=0}^{n-1} f_i x^i = [f_0, f_1, \dots \dots \dots \dots \dots \dots \dots f_{n-1}]$$

we use $\otimes$ to multiplication in R
$f \otimes g=h$ with

$$h_k = \sum_{i=0}^{k} f_i g_{k-i} + \sum_{i=k+1}^{n-1} f_i g_{n+k-i} = \sum_{i+j \equiv k} f_i g_j$$

III. SOME OTHER PKC'S

A. PKC's

Public key cryptography is cryptography in which a pair of keys is used to encrypt and decrypt a message so that it arrives securely. In the beginning, a network user receives a public and private key pair from a certificate authority (CA). Any other user who wants to send an encrypted message can get the intended recipient's public key from a public directory. They use this key to encrypt the message, and they send it to the recipient. When the recipient gets the message, they decrypt it with their private key, which no one else should have access to.

Public-key cryptography denotes to a cryptographic system and it requires separate keys, these keys are used to encrypt and decrypt the plaintext and ciphertext simultaneously, for the security purpose there needed one of the two keys must be secret [8][9]. It must be impractical to decipher a message if other information is not available. Knowledge of the algorithm, samples of ciphertext plus one of the keys must be inadequate to determine the other keys [10].

B. RSA

RSA algorithm developed by Rivest, Shamir and Adleman that is based on the presumed difficulty of factoring large integers. RSA algorithm uses two prime numbers and then calculates their product of prime numbers, user takes two random prime as a private key and then publishes there product as a public key. There is a problem in factoring that no one can derive the private key from the public key [11]. To protect the encryption, the minimum number of bits in n should be 2048. The RSA cryptosystem is based on the difficulty of factoring large integers.

C. Elliptic Curve

Most of standards and products which use public-key cryptography for digital signatures and encryption use RSA. In recent years as we have seen the key length for secure RSA use has increased. Recently, elliptic curve cryptography (ECC) a challenging system has started to challenge RSA.

For Public-Key Cryptography ECC is viewing up in standardization efforts, including the IEEE P1363 Standard. The principal magnetism of ECC, in compare to RSA, is that it seems to offer the same security for a far lesser key size, thus keep down processing overhead. The assurance level in ECC is not yet as high like that in RSA. ECC is basically more difficult to give explanation than either RSA or Diffie-Hellman. Elliptic curves are different from ellipses. They are thus named for the reason that they are explained through cubic equations, alike to those uses for calculating the circumference of an ellipse.

Cubic equations for elliptic curves
$y^2 + axy + by = x^3 + cx^2 + dx + e$

The main advantage promised by ECC is a small key size, transmission requirements and reducing storage, a 256bit ECC public key should provide comparable security to a 3072bit RSA public key.

An elliptic curve is a plane curve, equation for the elliptic curve is the form of
$y^2 = x^3 + ax + b$

The degree of this equation is 3 or said to be cubic which define an elliptic curve, it is a single element and called the point at identify.

D. McEliece

The McEliece cryptosystem is an asymmetric encryption algorithm, this cryptosystem based on an error correcting code. McEliece has three algorithms: deterministic decryption algorithm, a probabilistic encryption algorithm and probabilistic key generation algorithm which produces a public and a private key. It has advantages over RSA, the encryptions and decryptions are much faster and with the increase of the key size, the security becomes much faster. The main disadvantage of McEliece is that the public and private keys are used to large matrices.

The following table shows the comparison in NTRU, RSA and McEliece where N represent block size parameter.

TABLE I
COMPARISON WITH NTRU SIZES

	NTRU	RSA	McEliece
Encryption Speed	N^2	N^2	N^2
Decryption Speed	N^2	N^3	N^2
Public Key	N	N	N^2
Private Key	N	N	N^2
Message Expansion	varies	1-1	2-1

IV. NTRU ALGORITHM

A. Key Creation

FiguresTo create NTRU key (public and private key), there randomly chooses two polynomials from the set of $f \in L_f$ and $g \in L_g$[12][13]. Polynomial f must satisfy the requirements and it must have inverse modulo p and modulo q. It is denoted by f_p^{-1} and f_q^{-1}.

$f_p^{-1} \otimes f \equiv 1 \bmod p$

$f_q^{-1} \otimes f \equiv 1 \bmod q$

Polynomial h is the public key and it is given by

$h \equiv pf_q^{-1} \otimes g \bmod p$

B. Encryption

To encrypt plain text message m, random polynomial r is chosen from the set $r \in L_r$. Plaintext message m is polynomial in $Z[X]/(p, X^n-1)$ then encryption e is computed as

$e \equiv r \otimes h + [m + H(m, [r \otimes h]_p)X^{n-k} + G([r \otimes h]_p)]_p \pmod q$

C. Decryption

Message e is decrypt by a private key f.

$a \equiv f \otimes e \bmod q$

where the coefficients of a in the interval from [–q/2 to q/2]. If we treating a as a polynomial with integer coefficient, the temporary polynomial $t \in Z[X]/(p, X^n-1)$

$t = f_p^{-1} \otimes a \pmod p$

on computing the other temporary quantities

$b \equiv e - t \pmod p$

$c \equiv t - G(b) \pmod p$

D. Working Description

Polynomial a is computed as

$a \equiv f \otimes e$

$\equiv f \otimes r \otimes h + [m + H(m, [r \otimes h]_p)X^{n-k} + G([r \otimes h]_p)]_p \pmod q$

$\equiv f \otimes pr \otimes f_q^{-1} \otimes g + f \otimes [m + H(m, [r \otimes h]_p)X^{n-k} + G([r \otimes h]_p)]_p \pmod q$

$\equiv pr \otimes g + f \otimes [m + H(m, [r \otimes h]_p)X^{n-k} + G([r \otimes h]_p)]_p \pmod q$

For appropriate parameter choices, all of its coefficients lie between [–q/2 to q/2], it doesn't change if its coefficients are reduced modulo q. It means reduces the coefficients of $f \otimes e$ modulo q into the interval from [–q/2 to q/2], So on reducing the modulo p doesn't have an effect on the coefficients.

$a \equiv pr \otimes g + f \otimes [m + H(m, [r \otimes h]_p)X^{n-k} + G([r \otimes h]_p)]_p$ in R

Reducing a modulo p then it gives the polynomial

$f \otimes [m + H(m, [r \otimes h]_p)X^{n-k} + G([r \otimes h]_p)]_p$ in R

on multiplying by f_p^{-1}

$t = m + H(m, [r \otimes h]_p)X^{n-k} + G([r \otimes h]_p)]_p$ in $Z[X]/(p, X^n-1)$

Thus on computes $b = e - t$ and recovering $b = r \otimes h$

$c = m + H(m, [r \otimes h]_p)X^{n-k}$

V. IMPLEMENTATION OF NTRU

A. Choosing n

If the message is in the binary form, n is the number of bits that can be transported. For providing k bits of security and prevent some particular attacks, 2k bits can be transported. We set n to be the first prime number greater than 3k. It should be noted that n might need being changed if one cannot find appropriate values of the remaining parameters.

B. Choosing f, g, r and m

Let F, g and r to be binary polynomials with d_f , d_g and d_r number of 1s respectively. We take f =1+ pF so that the second convolution product in the decryption can be eliminated since f_p^{-1}. In addition, since security will be increases when h is invertible and also g to be invertible and we set $d_g = n/2$.

C. Choosing p and q

It is already noted that p and q should be relatively prime p is fixed to be the integer value of 2 so that we can work with binary polynomials. Also, q must have a higher order modulo n, i.e. the order of divisors of X^{n-1}modulo q should be high, for example (n-1) or (n-1)/2. In addition, to achieve better lattice security we must keep f and g as large as possible relative to q.

D. Encryption

$m \equiv (m` + bX^{n-k}) + H([\{r*h\}_q]_p)(mod)$

$e \equiv r*h + m \pmod q$

E. Decryption

$m=[f_p^{-1}[f*e]_q]_p$

for obtaining m`, we should need to compute the values

$x = e-m$ and $y=[m-H(\{[x]_q\}_p)]_p$

$y=y`+y``X^{n-k}$

if the condition is

$x=[G(y`+y``)h]_q$ and $y`B_{n-k}(d_m)$

satisfied, then ciphertext is valid.

and $y` = m`$ is the plaintext.

1) NTRU encryption and decryption execution timings before modification

TABLE II
NTRU PERFORMANCE

Text Sizes	Encryption	Decryption
128 bits	0.000019	0.000019
256 bits	0.000018	0.049
512 bits	0.05697	0.048
1024 bits	0.189	0.0601
2048 bits	0.279	0.0612
5120 bits	0.65895	0.19586

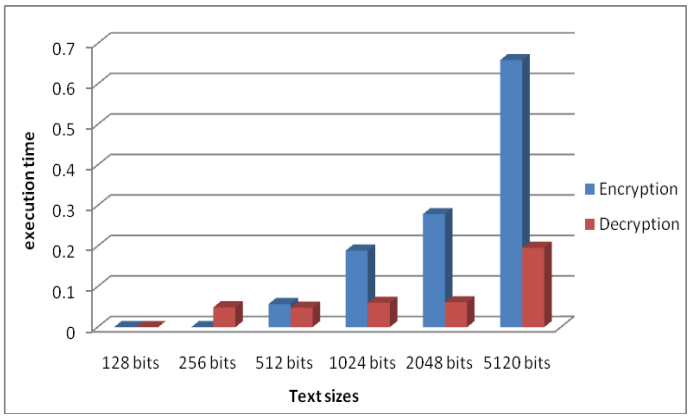

Fig. 1 NTRU Performances

2). NTRU encryption and decryption execution timings after modification

TABLE III
IMPROVED NTRU PERFORMANCES

Text Size	Encryption	Decryption
128 bits	0.00000009	0.00000009
256 bits	0.00000009	0.06101

512 bits	0.06202	0.06101
1024 bits	0.11077	0.06101
2048 bits	0.27562	0.06101
5120 bits	0.6425	0.16573

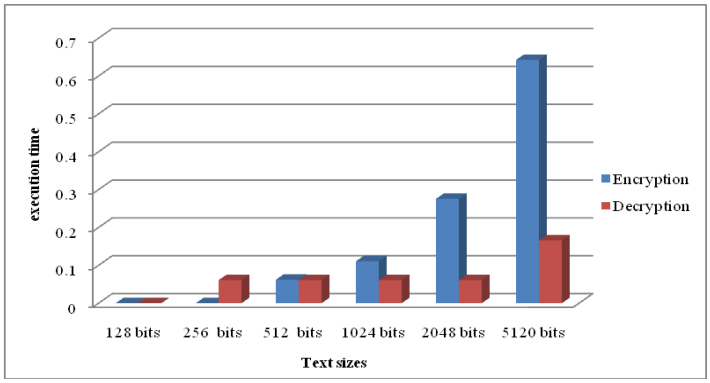

Fig.2 Improved NTRU Performances

3). Comparison of Improved NTRU encryption and NTRU encryption Performances

TABLE IV
COMPARISON OF IMPROVED NTRU ENCRYPTION AND NTRU ENCRYPTION PERFORMANCES

Text Size	Improved NTRU Encryption	NTRU Encryption
128 bits	0.00000009	0.000019
256 bits	0.00000009	0.000018
512 bits	0.06202	0.05697
1024 bits	0.11077	0.189
2048 bits	0.27562	0.279
5120 bits	0.6425	0.65895

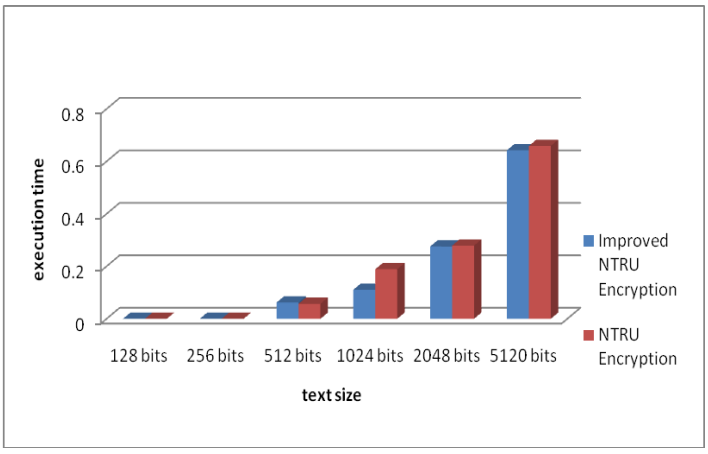

Fig.3 NTRU and Improved NTRU Encryption Performances

4). Comparison of Improved NTRU decryption and NTRU decryption Performances

TABLE V
COMPARISON OF IMPROVED NTRU DECRYPTION AND NTRU DECRYPTION PERFORMANCES

Text Size	Improved NTRU Decryption	NTRU Decryption
128 bits	0.00000009	0.000019
256 bits	0.06101	0.049
512 bits	0.06101	0.048

1024 bits	0.06101	0.0601
2048 bits	0.06101	0.0612
5120 bits	0.16573	0.19586

Fig.4 NTRU and Improved NTRU Encryption Performances

VI. CONCLUSIONS

In our proposed work we have modified the present existing NTRU algorithm and found out that modified NTRU works better than the present existing NTRU. A better and faster implementation of the algorithm is possible which would efficiently and quickly encrypt and decrypt large files. More efficient algorithms can be implemented for polynomial generation. The NTRU algorithm is a public key cryptosystem which is very fast cryptosystem and it is based on rings. It gives advantages for easy generation of keys, low memory and high speed. The entire operations contained in these procedures are convolution multiplication, addition and modular arithmetic. To be relevant NTRU cryptosystem appropriately, the analysis and description in this paper has vast importance. This paper gives brief description and analysis of the NTRU Cryptosystem and provides some help in the improvement of the cryptosystems for the network security.

REFERENCES

[1] J. Hoffstein, J. Pipper, and J. H. Silverman, "NTRU: A Ring Based Public Key Cryptosystem", in Proc. of Algorithmic Number Theory: Third International Symposium (ANTS 3) (J. P. Buhler, ed.), vol. LNCS 1423, Springer-Verlag, June 21-25 1998, pp. 267-288.

[2] http://en.wikipedia.org/wiki/NTRU_Cryptosystems,_Inc.

[3] Coppersmith and A. Shamir, "Lattice attacks on NTRU," in Proc. of EUROCRYPT 97, Lecture Notes in Computer Science, Springer-Verlag, 1997[CS97].

[4] J. Hoffstein, J. Pipper, and J. H. Silverman, "NTRU: A New High Speed Public Key Cryptosystem", Preprint, presented at the rump session of Crypto 1996. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Claredon, 1892, pp. 68-73

[5] N. Gama and N. Howgrave-Graham and P.Q. Nguyen, "Symplectic Lattice eduction and NTRU", In: Proc. of EuroCrypr2006, 2006.

[6] NTRU Cryptosystems, The NTRUEncrypt Public Key Cryptosystem: Advanced Topics.

[7] Asper Scholten and Frederik Vercauteren, "An Introduction to Elliptic and Hyperelliptic Curve Cryptography and the NTRU Cryptosystem", Available: http://www.math.unibonn.de/~saxena/courses/WS2010-ref4.pdf

[8] Joseph H. Silverman, Invertibility in Truncated Polynomial Rings, NTRU Cryptosystems Technical Report 9, available at http://www.ntru.com.

[9] NTRU Cryptosystems, The NTRUEncrypt Public Key Cryptosystem: Further Topics in Fast Implementation.

[10] NTRU Cryptosystems, The NTRU Encrypt Public Key Cryptosystem Basic Tutorial available: http://www.ntru.com/cryptolab/tutorial_pkcs.html

[11] Rashmi Jha, Anil Kumar Saini, "A Comparative Analysis & Enhancement of NTRU Algorithm for Network Security and Performance Improvement," in *IEEE*, 2011.

[12] NTRU Cryptosystems, The NTRU Encrypt Public Key Cryptosystem: EnhancementsII Available: http://www.ntru.com/cryptolab/tutorial

[13] NTRU Cryptosystems, The NTRU Encrypt Public Key Cryptosystem: EnhancementsI Available: http://www.ntru.com/cryptolab/tutorial_advanced.htmal_hamming.html

[14] Joseph H. Silverman, Almost Inverses and Fast NTRU Key Creation, NTRU Cryptosystems Technical Report 14, available at http://www.ntru.com.

