UACEE International Journal of Advances in Computer Networks and its Security – IJCNS
Volume 3 : Issue 2 [ISSN 2250 – 3757]
Publication Date : 05 June 2013

Matrix Formulation of NTRU Algorithm using multiple Public keys from Matrix Data Bank for High Degree polynomials

Vinay Kumar, Mohan Rao Mamdikar, Dr. D. Gosh

***Abstract*— In many business sectors sending secure information over public channels has become a challenging task. Data encryption is not the most efficient method to counter attacks for adversaries. One form of encryption called the symmetric key encryption uses the same key for encryption at sender's end and for decryption at the receiver's end. World-wide encryption standards such as DES and AES are used in Government and public domains. However symmetric key encryptions are prone to attacks by intruders. Asymmetric key encryption has been proposed in the Jofferey et al. [4] wherein the key to be used to send a message to R is made public and R uses his own private key to decrypt the encrypted message. NTRU is one such Public key Cryptosystem (PKCS), which is based on polynomial Ring Theory. The security of NTRU depends on the lattices. Several studies have suggested that it is difficult to know whether a polynomial is invertible or not. Nayak et al.[2] introduced a new method as a matrix solution to solve the problem. In this paper we propose the usage of multiple public keys to encrypt a text message. This method will be more secure and robust as there will be a unique private key for each public key. We also discuss about the safe exchange of the public key sequence numbers along with encrypted text block which is essential for correct decryption by the receiver. The security is further enhanced in the proposed method by taking high degree of polynomials from the matrix data bank created by us N = 225 i.e. 15x15 in which each matrix has an inverse and we use a matrix from data bank randomly. The method is extensively tested algorithm design and proved and faster since we can encrypt and decrypt a large number of blocks simultaneously.**

Keywords—NTRU, Private Key, Public Key, Encryption, Decryption, Modular Operation, matrix databank.

I. Introduction

The NTRU encryption system and the related signature scheme are both built on polynomial algebra. The basic object are truncated polynomials in the ring $R = Z[x]/(x^{N-1})$ and the basic tool is the reduction of polynomials with respect to two relatively prime moduli. The security of the systems is based on the difficulty of finding a "short" factorization for such polynomials. This latter problem is equivalent to finding a short vector in a certain N dimensional lattice, a commonly known and also widely studied NP hard problem. NTRU polynomials a(x) are frequently reduced modulo p and q, the small and large moduli. The large modulus q is an integer, so reduction of $a(x) = a_0 + a_1x + a_2x^2 + \text{-----} + a_{N-1}x^{N-1}$ mod q means just reduction of each a_i modulo q. The small modulus p can also be an integer. It is required that p and q are relatively prime. The main objects in the systems are "small" polynomials; i.e. polynomials with small coefficients. The public key h is defined by an equation f*h = p*g (mod q), where f and g are small polynomials. The polynomial f should always have inverses modulo p and q, $f*f_p \equiv 1$ (mod p) and $f*f_q \equiv 1$(mod q). Moreover, the parameters N, p and q are also public, and can be used as common domain parameters for all users. Polynomials f and g are private to the key owner. The polynomial g is needed only in key generation. The owner Alice chooses two small polynomials f and g in the ring of truncated polynomials and keeps f and g private. He then computes inverse of f (mod p) [f_p] and inverse of f (mod q) [f_q], where p and q are relatively prime to each other. He then computes $h = p*f_q*g$(mod q), which becomes the public key for Alice and the pair of polynomials f and f_p forms his private key pair. The message is also represented in the form of a truncated polynomial. Let it be m. The sender Bob encrypts using the public key of Alice i.e., h as E = h*R + m (mod q), where R is a random polynomial basically used to obscure the message. This encrypted message may be sent in a public channel. Alice decrypts the encrypted message using his private key pair by performing the following operations:

A = f *E (mod q)
B = A (mod p)
$C = f_p * B$(mod p) and C is the original message:
$C = f_p *[f * (p * f_q *g *R +m)(\text{mod } q)](\text{mod } p) = m$ using
The identities $f *f_p = 1$ and $f * f_q = 1$

II. Proposed Algorithm

In this paper we propose to use multiple set of public-private key pairs in form of matrices of order N x N as compared to a single key pair used by Nayak et al.[2].The public key sets are generated by the owner of the keys and made available to everyone. The sender who wishes to send a message to the receiver will store these public keys in an array

Vinay Kumar
National Institute of Technology, Durgapur
India
vk841232@gmail.com

Mohan Rao Mamdikar
National Institute of Technology, Durgapur
India
mohanrao.mamdikar@gmail.com

Dr. D. Ghosh
National Institute of Technology, Durgapur
India
profdg@yahoo.com

and choose them in a random sequence to encrypt the text message to be sent to the receiver. Each of these public keys will have a unique corresponding private key which will be known only to the receiver of the message. This same will be used for decryption to get back the original text. The owner Alice first creates a set of public/private key pair. He first randomly chooses two matrices f and g consisting of elements {1, 0, and -1}. Matrix f should be an invertible matrix (modulus p) i.e. determinant of f should be 1 and the number of 1's in f should be one more than number of -1's. Similarly the number of 1's and -1's in g should be equal but g's determinant need not necessarily be 1. Alice keeps the matrices f and g private, since anyone who knows either one of them will be able to decrypt messages sent to Alice. Alice's next step is to compute the inverse of f modulo q [f_q] and the inverse of f modulo p [f_p]. Alice's private key is pair of matrices f and f_p. Finally Alice computes h = p*fq*g(modulo q), which is his public key. Similarly he creates other sets of public and private key pairs. The proposed algorithm has been designed in three stages:

First stage - Key generation, second stage -procedure for encryption and third stage- procedure decryption as follows.

> ***Algorithm: NTRU with multiple message blocks***
> *{KEY GENERATION}*
> *1: Generate random matrix f and g*
> *2: compute inverse f(mod p)[f_p] and f(mod q)[f_q]*
> *3: compute public key h = (p * f_q * g) (mod q)*
> *{ENCRYPTION}*
> *4: text message convert into ternary representation:*
> *5: replace all 2's with -1*
> *6: generate random matrix R*
> *7: compute encrypted message E = h * R + m (mod q)*
> *8: stores message in array with index (if possible)*
> *{DECRYPTION}*
> *9: compute a (mod q) if possible*
> *10: compute A = f * E (mod q)*
> *11: choose coefficients of A into -q/2 to q/2*
> *12: compute B = A *(mod q)*
> *13: compute C = f_p*B (mod p) which is original message*

III. Proof of Algorithm

Bob's encrypted text block E = h * R + m (mod q), where h is the Alice's 1st public key, R is the blinding matrix used by sender Bob and m is the original text message. But Alice doesn't know the values of R and m. Alice 's first step is to compute f * E, where f is the private key corresponding to his public key h and then reduce the resultant modulo q. Since A's public key h was actually formed by multiplying p * fq * g, and reducing its coefficients modulo q i.e. h = p * fq * g(modulo q), where fq is inverse of f(modulo q) and g is the 2nd matrix corresponding to f generated by Alice during the private-public key generation process. So although Alice doesn't know R and m, when he computes A = f * E (modulo q), he is actually performing the following computation and the details of the computation is itself explanatory to establish that Alice is able to decrypt original message m back.

*A = f *E (modulo q)*
= f(R*h + m) (modulo q)*
*[Since E =R*h + m (modulo q)]*
= f(R*p*fq*g +m) (modulo q)*
*[Since h =p*fq*g (modulo q)]*
*= p*R*g + f*m (modulo q)*
*[Since f*fq = I (modulo q where I is the identity matrix]*

Next Alice computes
B = A (mod p)
*B = (p*R*g + f*m)(mod p)*
*= f*m (mod p)*
*[Since p*R*g (mod p) = 0]*

The last step is to find:
*C = fq*B (mod p)*
=fq f*m (mod p)*
*= m (mod p) [since f_p * f (mod p) = I]*
= m Q.E.D.

IV. Illustrations of the proposed Algorithm

A. *Generation of multiple public and private key using matrix databank*

The owner Alice uses the following parameters for obtaining the Public key and private key pair for encrypting the Plain Text message based on algorithm from step 1 to 3. f = 15 x 15 matrix, g = 15 x 15 matrix, p = 3 and q = 128, N = 225. f and g are two randomly generated matrices consisting of - 1's, 0's and +1's.The number of +1's is more than the number of -1's in f and the determinant of f equals to 1.Matrix g consists of equal number of -1's and +1's as shown in table I and II.

TABLE I. RANDOM MATRIX *f*

0	1	1	-1	1	0	0	-1	-1	0	-1	-1	0	0	1
1	0	-1	0	0	-1	1	0	-1	-1	0	1	1	-1	0
1	0	0	1	0	1	-1	0	-1	1	0	1	1	-1	0
-1	0	1	-1	1	-1	1	1	-1	0	1	0	0	1	1
-1	0	1	0	-1	-1	1	-1	1	0	-1	1	0	0	-1
0	1	1	0	1	1	-1	1	0	0	1	0	1	1	-1
0	-1	-1	0	-1	1	0	1	0	0	0	-1	1	-1	1
-1	-1	0	1	1	0	-1	1	0	1	0	-1	1	1	-1
1	0	-1	-1	1	-1	0	-1	0	0	0	-1	1	-1	0

0	0	1	0	-1	-1	0	1	-1	-1	-1	1	-1	-1	0
1	1	1	-1	1	0	0	0	0	-1	1	1	0	1	0
-1	0	0	0	-1	1	0	1	1	-1	-1	-1	0	-1	0
0	0	0	-1	0	0	0	0	0	-1	0	-1	1	1	0
0	0	0	-1	1	-1	0	0	1	0	-1	1	1	-1	-1
-1	1	0	-1	0	1	-1	0	0	1	0	1	0	-1	0

TABLE II. 2ND RANDOM MATRIX g

-1	0	1	-1	0	-1	1	-1	1	1	-1	0	-1	1	1
1	1	0	-1	-1	1	0	1	-1	0	0	-1	0	-1	0
1	1	-1	1	-1	-1	1	0	-1	0	-1	-1	1	1	-1
0	0	-1	1	0	0	-1	-1	0	-1	0	-1	1	1	-1
1	-1	1	1	0	0	0	-1	1	-1	0	0	1	-1	0
0	0	1	-1	1	-1	-1	0	-1	0	-1	1	0	-1	1
-1	0	1	-1	1	1	-1	1	1	0	-1	1	1	1	1
-1	1	0	0	1	1	-1	0	0	0	-1	1	-1	-1	-1
1	0	0	-1	1	0	-1	1	1	1	1	1	0	-1	0
1	0	0	0	0	-1	-1	-1	-1	0	0	-1	0	0	1
1	-1	1	0	0	-1	0	-1	-1	1	-1	1	-1	-1	1
0	0	1	0	1	0	0	0	1	0	-1	-1	-1	1	0
0	0	0	1	1	-1	0	0	0	-1	0	0	1	1	-1
0	0	1	-1	1	1	-1	1	1	0	1	1	0	0	-1
1	0	-1	1	-1	1	-1	-1	-1	1	0	-1	-1	1	0

The parameters f and g are private to Alice where as h and N are known to everyone i.e. public. Alice generates 10 such random f and g pairs (f1; g1) to (f10; g10). Alice next find the inverse of f(mod p)[fp] and f(mod q)[fq] as shown in the table III and table IV

TABLE III. INVERSE MATRIX fp

1	1	1	2	1	1	1	2	0	2	1	0	2	0	0
0	0	2	1	0	2	1	1	2	1	2	2	0	2	2
0	0	2	2	1	2	1	2	0	2	1	1	0	1	0
0	2	0	2	1	1	2	0	1	1	0	2	2	2	1
1	2	1	1	0	2	2	0	0	1	2	1	1	2	0
2	2	1	2	1	0	2	2	2	1	0	1	1	0	2
0	1	2	2	1	2	1	0	1	1	0	2	0	0	2
2	0	0	2	2	2	0	2	1	1	1	2	1	1	0
1	1	2	0	1	2	2	2	2	2	2	2	2	1	2
1	2	2	1	2	2	2	0	0	0	1	1	1	0	1
2	1	0	0	2	1	1	0	2	0	2	1	1	0	2
2	0	0	1	1	2	0	0	2	0	2	2	2	0	0
2	1	0	2	0	0	0	2	2	0	2	0	0	2	1
1	0	2	2	1	2	1	1	1	1	1	2	0	0	1
1	2	0	1	0	2	1	1	0	0	2	2	1	2	0

.

TABLE IV. INVERSE MATRIX f_q

4	105	85	26	53	108	44	93	89	26	117	53	98	73	72
108	12	5	100	125	61	45	11	15	103	123	76	76	115	124
34	96	24	76	14	81	21	114	68	96	110	97	6	9	20
94	36	27	12	0	81	40	49	12	28	43	77	55	51	111
29	70	98	95	71	26	36	3	95	115	36	56	63	21	88
93	64	103	54	59	7	50	97	77	9	53	5	89	72	98
7	50	102	56	115	114	96	82	111	70	47	125	0	75	74
113	51	8	26	66	104	103	92	36	29	77	45	15	117	70
89	34	93	62	93	127	111	48	60	48	18	48	11	74	4
42	48	100	25	113	23	81	30	122	127	108	97	10	40	37
11	103	23	82	12	32	112	80	3	58	77	21	36	81	99
37	73	13	108	109	4	60	57	120	16	60	102	71	10	1
1	23	2	86	112	60	122	34	1	40	74	120	43	69	15
44	6	10	62	113	45	59	102	125	0	9	28	96	35	3
42	119	40	16	70	49	71	67	56	124	3	64	40	63	56

. Alice then generates his public key h which is made available to everyone using the formula h = (p * fq * g) (mod q) which is given in table V.

TABLE V. MATRIX FOR PUBLIC KEY h

112	42	39	41	7	96	119	98	84	112	114	14	84	115	42
124	12	107	46	49	23	20	67	69	8	49	62	112	72	48
18	67	48	103	4	67	45	74	57	56	44	117	116	84	71
49	5	89	76	7	122	10	83	37	54	14	98	87	121	33
87	25	9	101	51	32	111	121	17	76	35	95	22	82	18
111	57	102	81	19	114	50	69	119	57	84	83	79	11	29
71	74	78	116	77	122	96	3	30	33	0	30	122	95	18
25	65	63	72	63	3	89	106	85	125	13	124	19	76	0
74	79	11	122	64	39	61	81	10	70	12	60	76	5	17
56	48	32	101	69	74	33	98	65	8	119	119	64	63	40
127	49	98	64	41	15	28	27	64	44	117	9	41	28	71
16	48	71	98	113	52	120	27	105	4	95	124	96	31	105
38	3	119	45	63	82	43	89	112	91	27	114	29	124	8
41	4	103	74	3	0	53	123	1	89	109	108	91	79	101
4	25	29	112	123	93	80	90	112	85	101	119	59	60	12

Alice generates 10 such public keys (h_1,h_2,h_3,h_4,h_5,h_6,h_7,h_8,h_9,h_{10}) using different sets of f and g. This public key is made available to everyone and anyone who wishes to send message to Alice can use this public key to encrypt his text message then send it to the receiver Alice. Alice's private key is a pair of matrices f and fp. There will be 10 such private key pairs each corresponding to a unique public key.

B. *Proposed method for encryption in the algorithm*

Now according to proposed algorithm, the sender Bob wants to send a large text message to Alice using Alice's public keys. For this Bob breaks down the plain text message into smaller blocks, each of size equal to dimension of Alice's public key. Bob next chooses the 1st sub text block to be encrypted (of length 45 characters) and finds the ASCII values corresponding to each character of the chosen sub text. Bob then converts ASCII values of each character to its corresponding ternary form i.e. base 3 form. This ternary representation of ASCII value of each character is a 5 bit number consisting of 0's, 1's and 2's. Bob next replaces all the 2's in the ternary representation with -1. Bob then arranges the resultant ternary representations of ASCII values into a 15 x 15 matrix, where each row represents ternary forms of three characters (the first 5 bits for the 1st character, the next 5 bits for 2nd character and the last 5 bits for the 3rd character). Since there are 15 such rows and each row consists of 3 characters Bob can represent 45 characters at a time using a 15 x 15 matrix.

For instance let the text message to be encrypted be "In many business sectors sending a secure messGe over public channels has become a challenging task. Data encryption is the most efficient method to counteract attacks by adversaries. One form of encryption is to use the same key by the sender for encryption as well as decryption. Worldwide encryption standards such as DES and AES are used in Government and public domains. However symmetric key encryption is prone to attacks by symmetric intruders. Key encryption has been proposed in the literature wherein the key to be used for encryption is different from that used for decryption."

Then the first sub matrix m= "In many business sector sending a secure mess". The ASCII values and ternary representations corresponding to the given text message in table VI.

TABLE VI. ASCII VALUES AND TERNARY REPRESENTATION

Character	ASCII Vaule of character	Ternary representation of ASCII value
	32	01020
a	97	10121
b	98	10122
c	99	10200
d	100	10201
e	101	10202
g	103	10211
i	105	10220
m	109	11001
n	110	11002
o	111	11010
p	112	11011
r	114	11020
s	115	11021
t	116	11022
u	117	11100
v	118	11101
w	119	11102
y	121	11111
I	73	02201

Bob next rearranges these ternary representations of ASCII values to obtain 15 X15 matrix m form in the table VII

TABLE VII. SUB TEXT MESSAGE m

0	2	2	0	1	1	1	0	0	2	0	1	0	1	2
1	1	0	0	1	1	0	1	2	1	1	1	0	0	2
1	1	1	1	1	0	1	0	1	2	1	0	1	2	2
1	1	1	0	0	1	1	0	2	1	1	0	2	2	0
1	1	0	0	2	1	0	2	0	2	1	1	0	2	1
1	1	0	2	1	0	1	0	1	2	1	1	0	2	1
1	0	2	0	2	1	0	2	0	0	1	1	0	2	2
1	1	0	1	0	1	1	0	2	0	0	1	0	1	2
1	1	0	2	1	1	0	2	0	2	1	1	0	0	2
1	0	2	0	1	1	0	2	2	0	1	1	0	0	2
1	0	2	1	1	0	1	0	1	2	1	0	1	2	1
0	1	0	1	2	1	1	0	2	1	1	0	2	0	2
1	0	2	0	0	1	1	1	0	0	1	1	0	2	0
1	0	2	0	2	0	1	0	1	2	1	1	0	0	1
1	0	2	0	2	1	1	0	2	1	1	1	0	2	1

Bob next replaces all the 2's in m with -1's to obtain m as given in table VIII.

TABLE VIII. SUB TEXT MESSAGE AFTER REPLACING 2'S WITH -1

0	-1	-1	0	1	1	1	0	0	-1	0	1	0	1	-1
1	1	0	0	1	1	0	1	-1	1	1	1	0	0	-1
1	1	1	1	1	0	1	0	1	-1	1	0	1	-1	-1
1	1	1	0	0	1	1	0	-1	1	1	0	-1	-1	0
1	1	0	0	-1	1	0	-1	0	-1	1	1	0	-1	1
1	1	0	-1	1	0	1	0	1	-1	1	1	0	-1	1
1	0	-1	0	-1	1	0	-1	0	0	1	1	0	-1	-1
1	1	0	1	0	1	1	0	-1	0	0	1	0	1	-1
1	1	0	-1	1	1	0	-1	0	-1	1	1	0	0	-1
1	0	-1	0	1	1	0	-1	-1	0	1	1	0	0	-1
1	0	-1	1	1	0	1	0	1	-1	1	0	1	-1	1
0	1	0	1	-1	1	1	0	-1	1	1	0	-1	0	-1
1	0	-1	0	0	1	1	1	0	0	1	1	0	-1	0
1	0	-1	0	-1	0	1	0	1	-1	1	1	0	0	1
1	0	-1	0	-1	1	1	0	-1	1	1	1	0	-1	1

Bob next chooses a random matrix R consisting of 0's, 1's and -1's and of same order as f, g and m with equal number of 1's and -1's. This is a 'blinding value' which is used to obscure the text message in table IX.

TABLE IX. RANDOM MATRIX R OF SAME ORDER OF f, g

1	1	1	1	-1	-1	0	-1	1	-1	-1	-1	0	0	-1
1	1	-1	1	0	1	1	0	-1	1	1	-1	0	0	0
-1	-1	1	0	-1	0	0	1	1	0	0	0	0	-1	1
-1	-1	0	-1	1	1	-1	0	0	1	0	0	1	1	-1
-1	1	-1	1	0	-1	-1	-1	0	0	1	0	0	0	0
0	1	0	1	1	1	1	1	-1	-1	-1	0	0	1	-1
1	0	1	0	0	0	-1	0	0	-1	0	-1	0	1	1
0	0	1	0	1	-1	0	0	0	0	1	-1	1	-1	-1
-1	0	0	0	0	-1	1	-1	0	0	0	0	1	-1	0
0	0	1	0	0	-1	1	0	1	-1	1	1	0	1	-1
0	0	0	1	0	-1	-1	0	0	0	0	0	-1	1	-1
0	-1	-1	-1	-1	0	1	-1	0	0	0	0	1	0	0
1	0	-1	0	-1	0	1	1	0	0	-1	1	-1	0	1
1	-1	0	0	0	-1	1	0	1	1	0	-1	-1	1	-1
1	0	0	-1	0	0	0	0	0	0	-1	0	0	-1	1

Bob then encrypts the text message m to obtain the encrypted text E by using formula E = (h * R + m)(mod q) as given table X.

TABLE X. ENCRYPTED MATRIX E

15	32	70	108	116	41	57	92	103	21	123	62	101	56	107
108	11	111	1	36	86	65	75	1	39	11	17	16	8	12
5	20	77	15	111	64	105	26	115	76	52	84	103	38	62
81	45	71	54	17	109	118	108	82	113	29	69	0	55	3
77	55	21	121	5	78	80	63	39	107	54	20	1	44	89
46	118	7	118	125	49	108	93	79	30	71	107	19	125	84
47	56	66	78	17	98	103	113	90	22	123	17	124	88	76
78	11	40	15	41	36	126	123	111	101	64	81	118	123	21
16	67	75	118	125	50	118	68	15	50	121	88	115	107	8
27	92	111	47	86	27	12	14	109	96	79	25	116	115	88
67	79	111	34	104	23	52	44	39	52	76	91	121	108	106
30	119	84	73	73	33	60	104	126	75	15	106	117	67	108
73	47	28	49	14	95	91	0	29	83	91	91	55	14	74
62	73	99	124	30	69	93	97	39	82	34	107	49	98	0
82	97	50	19	73	16	100	25	100	41	66	59	97	106	100

.

Similarly Bob uses other public keys (h_2 to h_{10}) along with the correspondingly generated random matrices(R_2,R_3,R_4,R_5,R_6,R_7,R_8,R_9,R_{10}) to obtain other encrypted sub text matrices blocks (E_2,E_3,E_4,E_5,E_6,E_7,E_8, E_9,E_{10}) which will be sent to the receiver (Alice) along with sequence number of the public keys used for the encryption of the text blocks. The sequence number can be sent as an array of 10 random numbers. The receiver has to perform the modulo q operation on these numbers to obtain the corresponding sequence numbers. Since only the legitimate sender and the receiver know the exact value of 'q' therefore this method ensures the correct and safe exchange of key sequences

C. . *Decryption methodology of the Algorithm*

The receiver Alice receives the following matrix (1st encrypted message) from the sender (Bob), along with an array of 10 numbers from Bob [257,514,899,388,1157,1414,1927,1032,2441,1162].

Alice chooses the 1st number in an array i.e. 257 and performs *a* (mod) *q* operation on it (*q*=128) to obtain value 1. Hence he comes to know that the 1st public key has been used to encrypt the 1st text block. Similarly he performs the (mod) *q* operations on all the numbers of array to get the corresponding public key sequence number used for encryption of subsequent sub text blocks.

A next uses his private key f_1 to compute $D_1=f_1*E$ (mod 128), which is given table XI.

TABLE XI. DECRYPTED MATRIX D_1

0	118	6	118	121	121	4	126	10	0	6	1	2	120	17
9	12	7	14	127	8	3	11	119	123	123	121	123	126	123
19	1	3	1	3	9	4	5	0	8	2	120	120	11	124
4	1	113	8	126	7	2	2	126	11	125	5	122	6	124
117	126	3	125	118	2	119	127	3	127	113	3	2	122	3
3	5	122	124	120	2	5	127	6	123	8	10	5	108	16
125	123	117	123	126	122	22	4	127	4	126	4	3	113	8
116	8	117	3	6	12	7	124	117	0	13	4	12	124	0
119	4	125	7	122	110	2	116	3	118	3	3	7	120	120
1	3	5	4	120	126	121	2	3	8	117	126	118	1	126
0	124	7	119	116	2	126	123	15	115	1	9	8	121	9
118	2	10	0	5	120	126	126	6	4	3	121	3	123	127
125	120	117	127	124	123	122	127	7	9	2	126	127	3	3
114	4	126	12	124	118	6	124	126	125	5	0	2	124	114
6	6	7	3	12	6	2	4	1	2	123	0	122	18	106

Since Alice is computing D_1 modulo q, he chooses the coefficients of D_1 to lie between –q/2 and q/2. When Alice reduces the coefficients of f_1*E_1 (mod 128), he chooses values lying between -63 and 64 and not between 0 and 127.

Therefore after centre lifting the coefficients of matrix D_1 to lie between -63 and 64 he gets D_1 in table XII.

TABLE XII. DECRYPTED MATRIX D_1 AFTER CENTER LIFTING

0	-10	6	-10	-7	-7	4	-2	10	0	6	1	2	-8	17
9	12	7	14	-1	8	3	11	-9	-5	-5	-7	-5	-2	-5
19	1	3	1	3	9	4	5	0	8	2	-8	-8	11	-4
4	1	-15	8	-2	7	2	2	-2	11	-3	5	-6	6	-4
-11	-2	3	-3	-10	2	-9	-1	3	-1	-15	3	2	-6	3

3	5	-6	-4	-8	2	5	-1	6	-5	8	10	5	-20	16
-3	-5	-11	-5	-2	-6	22	4	-1	4	-2	4	3	-15	8
-12	8	-11	3	6	12	7	-4	-11	0	13	4	12	-4	0
-9	4	-3	7	-6	-18	2	-12	3	-10	3	3	7	-8	-8
1	3	5	4	-8	-2	-7	2	3	8	-11	-2	-10	1	-2
0	-4	7	-9	-12	2	-2	-5	15	-13	1	9	8	-7	9
-10	2	10	0	5	-8	-2	-2	6	4	3	-7	3	-5	-1
-3	-8	-11	-1	-4	-5	-6	-1	7	9	2	-2	-1	3	3
-14	4	-2	12	-4	-10	6	-4	-2	-3	5	0	2	-4	-14
6	6	7	3	12	6	2	4	1	2	-5	0	-6	18	-22

Next Alice reduces the coefficients of D_1 modulo p (p = 3) and obtains $B_1 = D_1$ (mod p), and Bob finally uses fp, the 2nd part of his private key to compute $C_1 = fp * B_1$ (mod p), which turns out to be C_1 given in table XIII.

TABLE XIII. ORIGINAL MATRIX

0	2	2	0	1	1	1	0	0	2	0	1	0	1	2
1	1	0	0	1	1	0	1	2	1	1	1	0	0	2
1	1	1	1	1	0	1	0	1	2	1	0	1	2	2
1	1	1	0	0	1	1	0	2	1	1	0	2	2	0
1	1	0	0	2	1	0	2	0	2	1	1	0	2	1
1	1	0	2	1	0	1	0	1	2	1	1	0	2	1
1	0	2	0	2	1	0	2	0	0	1	1	0	2	2
1	1	0	1	0	1	1	0	2	0	0	1	0	1	2
1	1	0	2	1	1	0	2	0	2	1	1	0	0	2
1	0	2	0	1	1	0	2	2	0	1	1	0	0	2
1	0	2	1	1	0	1	0	1	2	1	0	1	2	1
0	1	0	1	2	1	1	0	2	1	1	0	2	0	2
1	0	2	0	0	1	1	1	0	0	1	1	0	2	0
1	0	2	0	2	0	1	0	1	2	1	1	0	0	1
1	0	2	0	2	1	1	0	2	1	1	1	0	2	1

This is the initial 15X15 matrix M, which was the ternary representation of the ASCII values of the text characters to be encrypted. A next selects the first five elements of the 1st row of matrix C_1 and finds its decimal equivalent (ASCII) value. Next he finds the alphabet or character corresponding to this ASCII value. Then he selects the next five elements of the 1st row (6 to 10) and repeats the same procedure to obtain the 2nd character. Lastly he selects the last 5 elements (11 to 15) of the 1st row to obtain the 3rd character.

He repeats this entire procedure for all the remaining rows of matrix C_1 and decrypts all the characters and eventually obtains the original text message. Hence Alice successfully decrypts the 1st encrypted sub text block which was encrypted using Alice's 1st public key h and sent by sender Bob in an encrypted form of random 15X15 matrix using his private key matrices pair f and fp.

TABLE XIV. TERNARY VALUES AND CORRESPONDING CHARACTER

Five bits of matrix B	ASCII Value	Corresponding character
01012	32	
10121	97	a
10122	98	b
10200	99	c
10201	100	d
10202	101	e
10210	102	f
10211	103	g
10220	105	i
11001	109	m
11002	110	n
11010	111	o
11011	112	p
11020	114	r
11021	115	s
11022	116	t
11100	117	u
11101	118	v
11102	119	w
11110	120	x
11111	121	Y
02201	73	I

Alice repeats this entire decryption procedure for all the other encrypted sub text blocks received from the sender (Bob) and eventually obtains the original message that was sent by the sender.

V. Analysis of Algorithm

In our paper we have proposed a method for encryption and decryption using matrices instead of polynomials as proposed in the original NTRU cryptosystem by Joffrey Hoffstein, Jill Piper and Joshep Silverman. It uses 10 different sets of public keys to encrypt different blocks of text messages in a random sequence. This method is more secure and efficient compared to the polynomial method as there is no way to determine whether a truncated polynomial in NTRU Cryptosystem is invertible or not. We can use this method because a matrix is invertible only when its determinant is found (which is relatively easier to find out). Also this method has a high degree of security as each block of text message is encrypted using different set of public keys instead of using just a single key which is highly vulnerable as the intruder might be able to figure out a single public key.

In our paper we used following parameter and corresponding execution time for existing method and proposed method as shown in table XV and XVI respectively.

TABLE XV. ESTIMATED EXECUTION TIME FOR EXISTING ALGORITHM

Degree of polynomial (N)	Estimated Execution time
100 (10X10)	7.2312 sec
225(15X15)	4 hrs 22 min 21 sec 233ms
400 (20X20)	7 days 6 hrs 46 min 44 sec 194 ms

TABLE XVI. ESTIMATED EXECUTION TIME FOR PROPOSED ALGORITHM

Degree of polynomial (N)	Estimated Execution time
100 (10X10)	5.0052 sec
225(15X15)	5.7999 sec
400 (20X20)	6.9195

Advantages of the algorithm over existing standard method as proposed by Nayak et al.[2]

- This method is more secure than the proposed method by Nayak et al.[2] and also more efficient due to high degree of polynomials . Further, since for each block separate public keys are associated this can go 10 assigned randomly. Therefore it will increase the security to 10 fold than available in normal NTRU cryptosystem.
- In proposed method we do not required each and every time to generate private keys i.e. random matrices f and g. It saves huge computational time to generate these higher degree polynomials. The computational time for different higher degree polynomials (executed in MATLAB 7.8.0 (R2009a) on Hp Z600 workstation) are as follows.

TABLE XVII. COMPUTATIONAL TIME TO GENERATE RANDOM MATRICES

Degree of polynomial (N)	Estimated Execution time
100 (10X10)	2.8678sec
225(15X15)	4 hrs 22 min 18 sec
400 (20X20)	7 days 6 hrs 48 min 37sec

VI. Conclusion

We have successfully performed the encryption and decryption operation for text message consisting of 450 characters by splitting it into 10 sub text blocks, each of size 45 characters and using 10 different set of private-public key pairs.

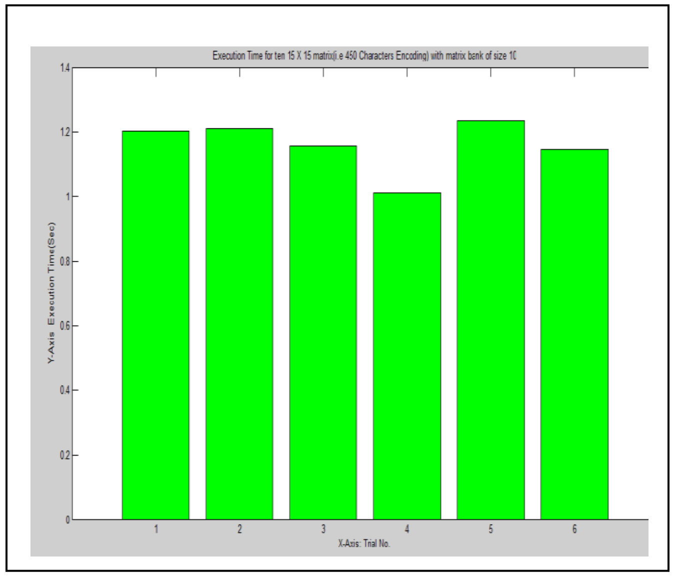

Figure 1. Average Eecution Time

We have compiled and run our code on MATLAB 7.8.0 (R2009a) and average execution time is to be 2.1234 sec.

Future scope of Work

In our proposed method, by using databank concept we can also perform encryption, decryption for the image and audio files more efficiently and securely.

References

[1] Whitefield Diffe, Martin E. Hellman "New Directions in cryptography "IEEE Information Theory, June 23-25, 1975 and IEEE International Symposium on Information theory, Sweden, June 21-24, 1976.

[2] Rakesh Nayak, C.V. Sastry and Jayaram Pradhan "A Matrix Formulation for NTRU cryptosystem" Proc. 16th IEEE International conference on Networks(ICON-2008), New Delhi, India 12-14 December, 2008.

[3] R.Nayak, C.V. Sastry and J. Pradhan, "Algorithmic Comparison between Polynomial Base and Matrix Base NTRU Cryptosystem", (IJCNS) International Journal of Computer and Network Security, vol.2, no.7, pp.58-63, July 2010.

[4] Jofferey Hoffstein, Jill Piper, Joshep H Silverman "NTRU A Ring based public key Cryptosystem" Lecture notes in Computer Science, Springer Verlag, Berlin 1433(1998),267-288.

[5] NTRU Cryptosystem, Technical Reports available at http:www.ntru.com

[6] C. Narsimham, Jayaram Pradhan "Performance analysis of Public key Cryptosystem RSA and NTRU" IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.8, August 2007,87-96.

[7] Xu-Ren Luo and Chen-Hui Jerry Lin, "Discussion on Matrix NTRU", IJCSNS International Journal of Computer Science and Network

[8] Joffrey Hoffstein, Joshep H Silverman "Optimization for NTRU" Proceedings of conference on Public key Cryptography and Computational number theory, Warsaw, De Gruyter, 200(Sep 11- 15) 77-88.

About Author (s):

Vinay Kumar was born in Siwan district of Bihar, India on 09/12/1983. He obtained his B.Tech degree in C.S.E in 2010 from Haldia Institute of Technology, West Bengal. His primary field of work includes NTRU Cryptosystem. He started his carrier from B.C.S., Bangalore as software developer. After wards he has worked in some other software company as Project Manager I.T. He is currently pursuing his M.Tech in C.S.E. from N.I.T Durgapur.

Mohan Rao Mamidkar was born in Bijapur district, Chhattisgarh, India on 09/02/1982. He obtained his B.Tech degree in C.S.E. in 2006 from N.I.T. Raipur. He is currently pursuing his M.Tech in C.S.E. from N.I.T Durgapur. His primary field of work includes NTRU Cryptosystem. Prior to M. Tech., He started his carrier from LIET, Hyderabad as Assistant professor after wards he has worked in LCIT Bilaspur Engineering College as Lecturer in C.S.E. department. He has 4 yrs experience in the field of teaching.