Volume 5, Issue 2, February 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

NTRU with Gaussian Integer Matrix

[1]Ashok Kumar Nanda*, [2]Rakesh Nayak, [3]Lalit Kumar Awasthi

[1]Research Scholar, CSE Department, NIT, Hamirpur, India

[2]Department of CSE, Sri Vasavi Eginerring College, Tadepalligudem, India

[3]Head, Computer Centre NIT, Hamirpur, India

Abstract—Every person needs information at any time, at anywhere. This is possible only by using wireless communication and mobile computing. SMS is cheapest, fast and not disturbing while at work. The speed of data encryption and providing security to sensitive information are most important for SMS. Public cryptography is most suitable to SMS. In public cryptography, there are two keys are needed for encryption/decryption process. RSA and NTRU[2,3] belong to the category of public or asymmetric key cryptosystem[1]. It is already been shown [8,9] Matrix form of NTRU is faster than that of Polynomial from of NTRU.

In this paper we proposed to use Gaussian Integer matrix to implement NTRU cryptosystem, which will improve the security by many folds as compared to NTRU with integer matrix.

Keywords—Encryption, Decryption, NTRU, Gaussian Integer, Matrix

I. BRIEF INTRODUCTION TO MATRIX NTRU

Bob creates a public/private key pair. He first randomly chooses two matrices X and Y, where matrix X should be an invertible matrix (modulus p and q). Here p and q are two relatively prime numbers. Thus he computes matrix Xq and Xp which satisfies X*Xq = I (modulo q) and X*Xp = I (modulo p).

(Bob can ensure the existence of inverse of matrix X by checking X is non-singular and X is invertible mod p (i.e. [det[X]](mod p) ≠ 0) . Otherwise he needs to go back and choose another matrix X.). Now Bob computes the product H = p*Xq*Y (modulo q). Bob's private key is the pair of matrices X and Xp and his public key is the matrix H.

To send message M, Alice chooses a random matrix R (which is of same order as matrix X), and Bob's public key H to compute the matrix. E = R*H + M (modulo q).

Now Bob has received Alice's encrypted message E and he decrypt it. He begins by using his private matrix X to compute the matrix. A = X*E (modulo q).

Bob next computes the matrix B = A (modulo p).

That is, he reduces each of the coefficients of A (modulo p). Finally Bob uses his other private matrix Xp to compute C = Xp*B (modulo p). The matrix C will be Alice's original message M.

1.1 MODULAR ARITHMETIC ON GAUSSIAN INTEGER MATRICES

Let A be an n x n matrix defined as aij + ibij

$$A = \begin{bmatrix} a_{11} \pm ib_{11} & \cdots & a_{1n} \pm ib_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} \pm ib_{n1} & \cdots & a_{nn} \pm ib_{nn} \end{bmatrix}$$

And let p be an integer. Then we define A (mod p) as

$$A(\text{mod } p) = \begin{bmatrix} a_{11} \pm ib_{11} & \cdots & a_{1n} \pm ib_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} \pm ib_{n1} & \cdots & a_{nn} \pm ib_{nn} \end{bmatrix} (\text{mod } p)$$

$$= \begin{bmatrix} a_{11} \pm ib_{11}(\text{mod } p) & \cdots & a_{1n} \pm ib_{1n}(\text{mod } p) \\ \vdots & \ddots & \vdots \\ a_{n1} \pm ib_{n1}(\text{mod } p) & \cdots & a_{nn} \pm ib_{nn}(\text{mod } p) \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}(\text{mod } p) \pm ib_{11}(\text{mod } p) & \cdots & a_{1n}(\text{mod } p) \pm ib_{1n}(\text{mod } p) \\ \vdots & \ddots & \vdots \\ a_{n1}(\text{mod } p) \pm ib_{n1}(\text{mod } p) & \cdots & a_{nn}(\text{mod } p) \pm ib_{nn}(\text{mod } p) \end{bmatrix}$$

The following identities hold good

(i) [A (mod p) + B(mod p)](mod p) = (A + B)(mod p)

(ii) [A (mod p) * B (mod p)](mod p) = (A*B)(mod p)

(iii) A-1 (mod p) = (A (mod p))-1(mod p)

The inverse of Gaussian Integer Matrix can be found by the following method

Let $A = \begin{bmatrix} a_{11} \pm ib_{11} & \cdots & a_{1n} \pm ib_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} \pm ib_{n1} & \cdots & a_{nn} \pm ib_{nn} \end{bmatrix}$ be a n x n non singular matrix.

Then $A^{-1} = \frac{1}{\alpha \pm i\beta}\begin{bmatrix} A_{11} \pm iB_{11} & \cdots & A_{1n} \pm iB_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} \pm iB_{n1} & \cdots & a_{nn} \pm iB_{nn} \end{bmatrix}$, where α+iβ is Determinant of A.

We can write $A^{-1} = \begin{bmatrix} \frac{A_{11} \pm iB_{11}}{\alpha \pm i\beta} & \cdots & \frac{A_{1n} \pm iB_{1n}}{\alpha \pm i\beta} \\ \vdots & \ddots & \vdots \\ \frac{A_{n1} \pm iB_{n1}}{\alpha \pm i\beta} & \cdots & \frac{a_{nn} \pm iB_{nn}}{\alpha \pm i\beta} \end{bmatrix}$

$$A^{-1}(\text{mod } p) = \begin{bmatrix} \frac{A_{11} \pm iB_{11}}{\alpha \pm i\beta} & \cdots & \frac{A_{1n} \pm iB_{1n}}{\alpha \pm i\beta} \\ \vdots & \ddots & \vdots \\ \frac{A_{n1} \pm iB_{n1}}{\alpha \pm i\beta} & \cdots & \frac{a_{nn} \pm iB_{nn}}{\alpha \pm i\beta} \end{bmatrix} (\text{mod } p)$$

Now we can multiply the complex conjugate of $\alpha \pm i\beta$ in the numerator and denominator of each $A_{ij} \pm iB_{ij}$.

$$A^{-1}(\text{mod } p) = \begin{bmatrix} \frac{A'_{11} \pm iB'_{11}}{\Delta} & \cdots & \frac{A'_{1n} \pm iB'_{1n}}{\Delta} \\ \vdots & \ddots & \vdots \\ \frac{A'_{n1} \pm iB'_{n1}}{\Delta} & \cdots & \frac{A'_{nn} \pm iB'_{nn}}{\Delta} \end{bmatrix} (\text{mod } p)$$

Where $A'_{ij} \pm iB'_{ij} = \text{Complex Conjugate}(\alpha \pm i\beta).(A_{ij} \pm iB_{ij})$ and $\Delta = \text{Complex Conjugate}(\alpha \pm i\beta).(\alpha \pm i\beta_{ij})$

$$A^{-1}(\text{mod } p) = \begin{bmatrix} X_{11} \pm iY_{11} & \cdots & X_{1n} \pm iY_{1n} \\ \vdots & \ddots & \vdots \\ X_{n1} \pm iY_{n1} & \cdots & X_{nn} \pm iY_{nn} \end{bmatrix}$$

Where Xij is defined as the smallest integer satisfying "p divides (Aij - XijΔ)", $0 \le Xij < p$. and Yij is defined as the smallest integer satisfying "p divides (Bij - YijΔ)", $0 \le Xij < p$.

II. PROPOSED GAUSSIAN INTEGER MATRIX USING NTRU

Bob creates a public/private key pair. He first randomly chooses two matrices X and Y, where matrix X should be an invertible matrix (modulus p). Bob keeps the matrices X and Y private, since anyone who knows either one of them will be able to decrypt messages sent to Bob. Bob's next step is to compute the inverse of X modulo q and the inverse of X modulo p. Thus he computes matrix Xq and Xp which satisfies

X*Xq = I (modulo q) and X*Xp = I (modulo p).

(Bob can ensure the existence of inverse of matrix X by checking X is non-singular and X is invertible mod p (i.e. [det[X]](mod p) ≠ 0) . Otherwise he needs to go back and choose another matrix X.). Now Bob computes the product H = p*Xq*Y (modulo q). Bob's private key is the pair of matrices X and Xp and his public key is the matrix H.

Example

Let 'p' and 'q' are small and large moduli numbers respectively. 117 is highest number used in the above square Integer Matrix. 127 is immediate next prime number after 117. So we have consideredp=127. Let us assume value of q=816 which is relatively prime to p=127. It is required that'p' and 'q' are relatively prime: gcd(p, q) = 1.

Bob randomly chooses two Gaussian Integer Matrices X, Y and R. He keeps the Matrices X, Y as private.

$$X = \begin{bmatrix} 0 & i & 1+i & 1 \\ i & 0 & 1 & i \\ 1 & i & 0 & 1+i \\ 1+i & 1 & i & 0 \end{bmatrix},$$

$$Y = \begin{bmatrix} i & 1 & 1 & 0 \\ 1-i & 1 & i & 0 \\ -1 & i & 1+i & 0 \\ i & 1+i & 1 & i \end{bmatrix}$$

and

$$R = \begin{bmatrix} -1+i & i & 1+i & 1 \\ 0 & 1+i & 0 & i \\ 1 & i & -1 & i \\ 1+i & 0 & 1-i & i \end{bmatrix}$$

As per NTRU algorithm, to compute the Matrix Xp Xq and we have considered two moduli p and q.. Next Bob computes two inverses Matrix Xp of X modulo with p' and Xq of X modulo with q'. He finds that

$$\text{Inverse of } X(\text{mod } p) = Xp = \begin{bmatrix} 57+i84 & 17+i83 & 44+i18 & 66+i26 \\ 17+i83 & 123+i75 & 53+i122 & 79+i57 \\ 66+i26 & 79+i57 & 70+i79 & 105+i118 \\ 44+i18 & 53+i122 & 4+i52 & 70+i79 \end{bmatrix}$$

$$\text{Inverse of } X(\text{mod } q) = Xq = \begin{bmatrix} 153+i61 & 183+i428 & 459+i184 & 245+i275 \\ 183+i428 & 398+i377 & 551+i397 & 826+i153 \\ 145+i275 & 826+i153 & 734+i826 & 214+i795 \\ 459+i184 & 551+i397 & 489+i550 & 734+i826 \end{bmatrix}$$

(a) Key generation

To generate the Public Key, Bob computes will apply the equation $H = pXq.Y(\text{mod } q)$

$$H = \begin{bmatrix} -275 - i389 & 415 + i166 & 70 + i332 & -332 + i70 \\ -280 + i319 & -262 - i358 & 236 - i83 & 83 + i236 \\ -415 - i166 & -96 - i140 & -319 - i153 & 153 - i319 \\ -319 + i101 & -275 - i389 & 210 + i236 & -236 + i83 \end{bmatrix}$$

(b) Encryption

Alice wants to send the message "I Like You" to Bob. Now this plain text message will converted into ASCII code. Each space between two words is considered as character. The equivalent ASCII codes of "I Like You" are {73, 32, 76, 105, 107, 101, 32, 89, 111, 117}. There are in total 10 characters including two space characters used in that message. The public key is a 4 X 4 matrix. So arrange these 10 ASCII codes into 4 X 4 matrix. We have to appended extra six space (ASCII code of space is 32) characters to fill completely 4 X 4 square Matrix.

$$= \begin{bmatrix} 73 & 32 & 76 & 105 \\ 107 & 101 & 32 & 89 \\ 111 & 117 & 32 & 32 \\ 32 & 32 & 32 & 32 \end{bmatrix}$$

Next we transform this Integer Matrix into Gaussian Integer Matrix. For this, we have purposed a concept that the Integer number of the corresponding ASCII code of a charterer into Gaussian Integer by taking LSB of the ASCII code as imaginary part of Gaussian Integer. The converted Gaussian Integer Matrix is called as Message Matrix and represented by 'm'. Now the message is converted into Gaussian Integer square Matrix form, which is shown below.

$$M = \begin{bmatrix} 7 + i3 & 3 + i2 & 7 + i6 & 10 + i5 \\ 10 + i7 & 10 + i & 3 + i2 & 8 + i9 \\ 11 + i & 11 + i7 & 3 + i2 & 3 + i2 \\ 3 + i2 & 3 + i2 & 3 + i2 & 3 + i2 \end{bmatrix}$$

Alice encrypt here message by using Bob's Public Key using **E=HR + M (modulo q)**

$$E = \begin{bmatrix} 339 + i187 & 309 + i378 & -211 + i299 & 54 - i231 \\ 54 - i356 & -130 + i224 & 374 + i277 & -67 + i385 \\ -142 + i320 & 374 - i76 & -93 - i11 & 200 - i426 \\ 112 - i335 & -220 + i116 & 107 - i133 & -246 - i198 \end{bmatrix}$$

This encrypted message E will be sent to Bob

(c) Decryption

When Bob receives the encrypted message E, he uses his private Matrix X to compute the Matrix. **A=X.E(modulo q)**

$$A = \begin{bmatrix} 6 - i103 & 6 + i284 & -252 + i137 & -5 + i396 \\ 6 - i116 & -120 + i13 & -359 - i115 & -258 + i269 \\ 255 + i18 & -251 + i144 & -248 - i240 & -379 + i145 \\ -114 + i28 & -123 + i398 & -125 + i272 & -243 + i408 \end{bmatrix}$$

Bob next computes the Matrix **B = A (modulo p).**

$$B = \begin{bmatrix} 6 + i24 & 6 + i30 & 2 + i10 & -5 + i15 \\ 6 + i11 & 7 + i13 & -5 + i12 & -4 + i15 \\ 1 + i18 & 3 + i17 & 6 + i14 & 2 + i18 \\ 13 + i28 & 4 + i17 & 2 + i18 & 11 + i27 \end{bmatrix}$$

Finally Bob uses his private Matrix Xp to compute **C = Xp.B (modulo p).**

$$C = \begin{bmatrix} 7 + i3 & 3 + i2 & 7 + i6 & 10 + i5 \\ 10 + i7 & 10 + i & 3 + i2 & 8 + i9 \\ 11 + i & 11 + i7 & 3 + i2 & 3 + i2 \\ 3 + i2 & 3 + i2 & 3 + i2 & 3 + i2 \end{bmatrix}$$

Now, the next step is to combine the integer and imaginary part of each element of the matrix by eliminating imaginary notation ('i') from above decrypted message Matrix C. Bob will get the ASCII codes from which the plain text is found to be "I Like You", which is the message sent by Alice.

III. COMPUTATIONAL COMPLEXITY

We have compared the complexity of the two approaches such as Integer Matrix and Gaussian Integer Matrix using NTRU crypto algorithm. Table 1 is representing the complexity costs of Key Generation, Encryption and Decryption with respect to number of additions, number of multiplications and number of modular operations are performed in both Integer based NTRU Cryptosystem and Gaussian Integer based NTRU Cryptosystem.

Assume, 'Ca' is cost of performing addition of two Matrices. 'Cm' is cost of performing multiplication of two Matrices. 'Cr' is cost of performing modulus operation of two Matrices. 'Cac' is cost of performing addition of two complex Matrix. 'Cmc' is cost of performing multiplication of two complex Matrix. 'Crc'cost of performing modulus operation of two complex Matrix. We assumed as Ca= 1, Cm= 2 and Cr=2 for calculating Key Generation Cost, Encryption Cost and Decryption Cost based on the formulas derived stated in Table1.

We inferred from Tables 1 and Tables 2. and Figure 1 to Figure 4 that value 'N' increases, all the complexity costs are increasing in nature of both Gaussian Integer Matrix based NTRU Cryptosystem and Integer based NTRU Cryptosystem. All complexity costs of Gaussian Integer based NTRU Cryptosystem is 3 to 4 times higher than the complexity costs of Integer based NTRU Cryptosystem because the former uses complex numbers instead of Integers. It needs more time for mathematical operations for encryption/ decryption of any message. The level of confusedness will increase twice due to the combination of real and imaginary numbers for hackers and need more to compute for hacking messages. So it is robust due to complexity provides more security.

Table 1: Comparison of Total Complexity Cost.

	Integer Matrix	**Gaussian Integer Matrix**
No. of Additions	2N2(N - 1)	8N3- 4N2
No. of Multiplications	2N3	8N3
No. of Modulo operations	N2	2N2
Cost of Key Generation	2N2(N - 1)*Ca + 2N3*Cm + N2*Cr	4N2 (2N - 1)*Ca + 8N3 * Cm + 2N2 * Cr
No. of Additions	N3	4N3
No. of Multiplications	N3	4N3
No. of Modulo operations	2N2	4N2
Cost of Encryption	N3*Ca + N3*Cm + 2N2*Cr	4N3 * Ca + 4N3 * Cm + 4N2 * Cr
No. of Additions	2 N2 (N - 1)	4N2(2N - 1)
No. of Multiplications	2 N3	8N3
No. of Modulo operations	3 N2	6N2
Cost of Decryption	2N2(N - 1)*Ca + 2N3*Cm + 3N2*Cr	4N2 (2N - 1)*Ca + 8N3 * Cm + 6N2 * Cr

Table 2: Comparison of Total Key Generation Complexity Cost Varying 'N'

N	**Total Key Generation Complexity Cost**		**Total Encryption Complexity Cost**		**Total Decryption Cost**	
	Gaussian Integer Matrix	**Integer Matrix**	**Gaussian Integer Matrix**	**Integer Matrix**	**Gaussian Integer Matrix**	**Integer Matrix**
1	130	35	120	45	190	65
2	1000	260	720	240	1720	380
3	3330	855	2160	675	3870	1125
4	7840	2000	4800	1440	8800	2070
5	14250	3875	9000	2625	16750	4625

Though the SMS security system using Gaussian Integer Matrix will be more secure, but it will incur higher cost also. So, caution may be taken in considering applications of proposed method with respect to the level of security that is required for a specific application.

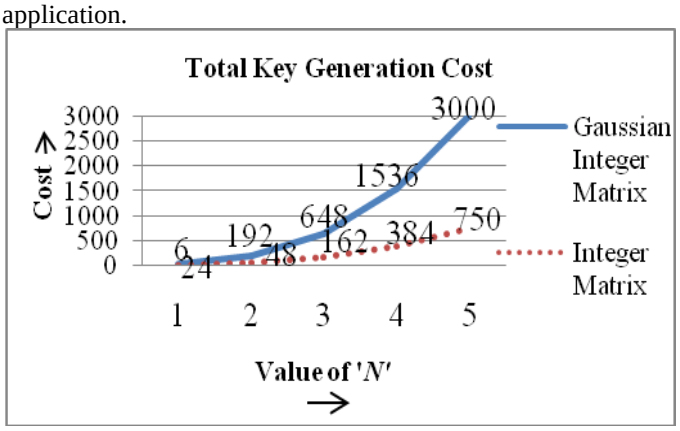

Figure 1:Comparisonof TotalKey Generation Complexity Cost between Integer Matrix and Gaussian Integer Matrix Using NTRU.

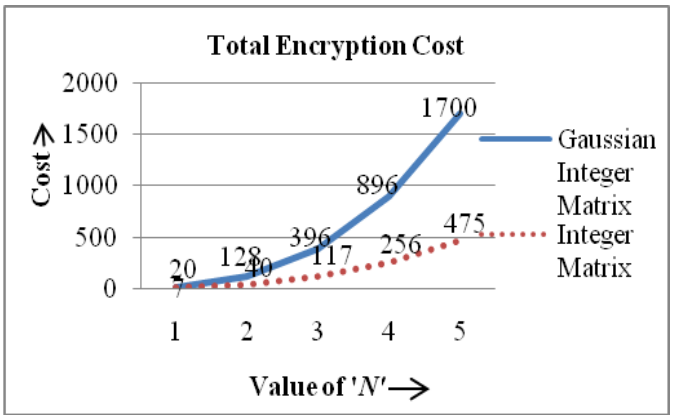

Figure 2:Comparisonof TotalEncryption Complexity Cost between Integer Matrix and Gaussian Integer Matrix Using NTRU.

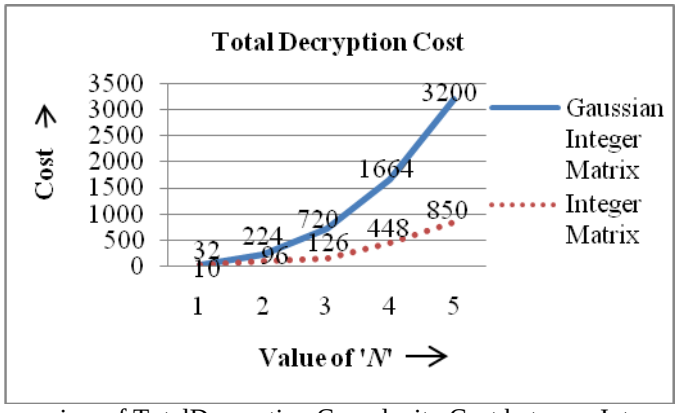

Figure 3:Comparisonof TotalDecryption Complexity Cost between Integer Matrix and Gaussian Integer Matrix Using NTRU.

IV. SECURITY

4.1 BRUTE FORCE ATTACK

In cryptography, a brute-force attack, or exhaustive key search, is a cryptanalytic attack that can, in theory, be used against any encrypted data (except for data encrypted in an information-theoretically secure manner). Such an attack might be used when it is not possible to take advantage of other weaknesses in an encryption system (if any exist) that would make the task easier. It consists of systematically checking all possible keys or passwords until the correct one is found. In the worst case, this would involve traversing the entire search space.

In NTRU the polynomial f is chosen of degree N. The coefficient of f can be {1, 0, -1}. So we have an exhaustive search, we have to search from the key space of size N3. Where as in Gaussian Integer matrix approach of NTRU each element of the matrix can be either {1+i, 1-i, -1+i, -1-i, 0,1,-1,-i and i}. So with exhaustive search, we have to search from the key space of size N9

Assume the polynomial NTRU with degree N=9. So the total search space will be 93=729 possible keys. The search space with Gaussian Integer Matrix will be 99=387420489. This is 531441 times more than that of Polynomial approach.

We can conclude easily that as Gaussian Integer Matrix have sufficiently larger key space so it is difficult to break. When the matrix size increases, the key space also increases exponentially.

4.2 REACTION ATTACK

The idea underlying a Reaction Attack is for the attacker to produce a sequence of encrypted messages e, each of which has a significant probability of decrypting into a valid message and also a significant probability of decrypting into an invalid message. It is assumed that the attacker is able to distinguish which messages have valid decryptions and which ones do not. This may be possible because an error message is generated when an invalid message is received, or it may be possible via a timing attack in which the attacker measures how long it takes a message to be decrypted.
The assumption is that an attacker is able to detect when a wrapping failure occurs during the decryption process.

An encrypted NTRU message has the form $e \equiv hr + m \pmod q$. The smallest modification that an attacker can make to e and still have it sometimes decrypt correctly is to replace e with $e' = e + npX_{ij}$ for some $0 \leq i,j < N$ and some $n \geq 1$. This will cause wrap or gap failure. if some coefficient of the intermediate decryption polynomial $a = prg + mf$ is within np of q/2 and if X_{ij} f has a corresponding +1 coefficient.

The important point to observe is that for the correct choice of n, the [i,j]'s which cause decryption failure for $e + npX_{ij}$ will reflect (with some shifting and possible duplication) the [i,j]'s for which the private key f has a term of the form $+X_{ij}$. Thus the attacker potentially gains information about the +1 bits in the private key f.

Similarly, using negative values for n may give information about the −1 bits of f.

An attack can begin to yield useful information, it must deform the encrypted message sufficiently that the attack can be detected by the NTRU user. We also note that such an attack can only hope to succeed if a single private key is used for the decryption of a large number of messages.

Countermeasures to Reaction Attacks on the NTRU PKCS There is no chance for a Reaction Attack to succeed if any given public/private key pair is used for only one message, or at most a few messages.

This solution would not be useful for public key cryptosystems such as RSA or ECC Because NTRU key creation is so fast that use of disposable key pairs is a very reasonable option. The Private/public key creation in RSA or ECC is quite time consuming.

If the owner of private key keeps track of the number of wrap failure messages received, she can simply discard the current key and replace it with a new public/private key. For even greater security, one might change keys after every wrap/gap failure.

V. CONCLUSION

The original NTRU crypto algorithm is based on the concept of 'N' degree truncated polynomial rings[4]. We have modified this crypto algorithm using the concept of Gaussian Integer Matrix, which is a maiden attempt among all present cryptosystem techniques. This novel scheme uses combination of real and imaginary values in Matrix format.

In this proposed work it can be seen that the encryption / decryption time in Gaussian Integer matrix is far more than that of integer matrix. It is hard to improve the security without slowing down the speed, and vice versa [15]. In our proposed paper the same time time for encryption/ decryptionis more but at the same time the security is also increased many fold. The most important novelty of our scheme is that it provides additional security due to use of complex numbers for NTRU Cryptosystem

REFERENCES

[1] William Stallings: "Lecture Notes for Use with Cryptography and Network Security".

[2] Fei Hu, Kyle Wilhelm, Michael Schab, Marcin Lukowiak, Stanislaw Radziszowski and Yang Xiao: "NTRU – based sensor network security: a low-power hardware implementation perspective", pp. 11, Security and Communication Networks, Wiley InterScience, 2008.

[3] SulaimanAlMuhteband PadmavathammaMokkala: "Performance analysis of Jordan Totient RSA (Jk – RSA) and NTRU", International Journal of Scientific & Engineering Research, Volume. 5, Issue. 3, March – 2014.

[4] Ron Steinfeld: "NTRU Cryptosystem: Recent Developments and Emerging Mathematical Problems in Finite Polynomial Rings", http://users.monash.edu.au/~rste/NTRU_survey.pdf

[5] A. Naresh Reddy, Rakesh Nayak and S. Baboo: "Analysis and Performance Characteristics of Cryptosystem using Image Files", International Journal of Computer Applications, Volume. 51, No. 22, August 2012.

[6] Jeffrey Hoffstein, J. Pipher and J. H. Silver man: "NTRU: A Ring-Based Public Key Cryptosystem", Algorithmic Number Theory (ANTS III), Portland, June 1998, J.P. Buhler (ed.), Lecture Notes in Computer Science 1423, Springer – Verlag, Berlin, pp. 267 – 288, 1998.

[7] Michael Coglianese and Bok-Min Goi: "MaTRU: A New NTRU - Based Cryptosystem", Lecture Notes in Computer Science, Springer –Verlag, Volume. 3797, pp. 232 – 243, 2005.

[8] Rakesh Nayak, C.V.Sastry and Jayaram Pradhan: "A matrix formulation for NTRU cryptosystem" in Proceedings of 16th IEEE, International Conference on Networks (ICON – 2008), pp. 1 – 5 at New Delhi, 12 – 14 December 2008.

[9] Rakesh Nayak, C.V Sastri and Jayaram Pradhan: "Algorithmic Comparison between Polynomial Base and Matrix Base NTRU Cryptosystem", International Journal of Computer and Network Security, Volume. 2, Issue. 7, 2010.

[10] Rakesh Nayak, Jayaram Pradhan and C V Sastry: "Evaluation of Performance Characteristics of Polynomial based and lattice Based NTRU Cryptosystem", ACEEE International Journal of Network Security, pp. 1 – 4, Volume. 03, Issue. 1, January 2012.

[11] Xiaoyu Shen, Zhenjun Du and Rong Chen: "Research on NTRU Algorithm for Mobile Java Security", in Proceedings International Conference on Scalable Computing and Communications; 8th International Conference on Embedded Computing (SCALCOM - EMBEDDEDCOM'09), pp. 366 – 369 at School of Information Science and Technology, Dalian Maritime University, Dalian, China during 25 – 27 September 2009. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5341656

[12] Zhenfei Zhang, Thomas Plantard, and Willy Susilo: "Reaction Attack on Outsourced Computing with FullyHomomorphic Encryption Schemes", http://www.uow.edu.au/~thomaspl/pdf/ZhaPlaSus11.pdf

[13] Ramanjaneyulu.S and Rakesh Nayak: "Secure Mobile System Using NTRU Encrypt": International Journal of Computer Trends and Technology, Volume. 4, Issue. 2, 2013.

[14] Elaine Barker, William Barker, William Burr, William Polk and Miles Smid: "Recommendation for Key Management – Part 1: General (Revised)" http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf

[15] A. Joseph Raphael and V. Sundaram: "Secured Communication through Fibonacci Numbers and Unicode Symbols", International Journal of Scientific & Engineering Research, Volume. 3, Issue. 4, April – 2012.

ABOUT AUTHOR

Mr. Ashok Kumar Nanda passed M.Tech in Computer Science & Engineering (CSE) from Guru Jambheswar University (GJU), Hissar, Haryana, India. He has around 8yrs teaching experience for both UG & PG courses. He was Associate Professor during Dec'06 to Jun'09. After that, he joined as full time research scholar in Cosmputer Science & Engineering (CSE) Department at National Institute of Technology (NIT), Hamirpur, Himachal Pradesh, India. His research area is light weight cryptography on Mobile devices. He has published 2 International Journal papers and 8 International Conferences papers. He is life member of Cryptography Research Society of India (CRSI), Computer Society of India (CSI), Indian Society for Technical Education (ISTE), graduate student member of IEEE and associate member of The Institution of Engineers (India) and member of International Association of Computer Science and Information Technology (IACSIT), International Association of Engineers (IAENG), Internet Society.

Dr. Rakesh Nayak working as a Associate Professor in the Department of Computer Science and Engineering at Sri Vasavi Engineering College,Tadepalligudem since 2010. He received his Ph.D. Degree in Computer Science from Behrampur University in the year 2013.He received his Master in Computer Applications Degree from Indira Gandhi National Open University in the year 2007 and M.Tech(CSE) from Acharya Nagarjuna University in 2010. He is also having M.Sc. and M.Phil. degrees in Mathematics from Sambalpur University. He completed M.Phil in Mathematics in the year 1997. He is having more than 14 years of teaching experience. He worked as a Lecturer in Mathematics in National Institute of Computer Science, Rourkela from 1999 for 4 years and worked as Asst. Professor(Sr. Grade) in the Department of Computer Application at Regency Institute of Technology, Yanam for 6 years. He is a life member of Indian Society for Technical Education (ISTE)), International Association of Engineers (IAENG), Internet Society and Senior member of the International Association of Computer Science and Information Technology (IACSIT)

Prof. Lalit Kumar Awasthi was born on 19 May 1966. He is a senior most Professor in Computer Science & Engineering (CSE) Department at National Institute of Technology, Hamirpur, Himachal Pradesh, India. He has completed M. Tech. CSE from Indian Institute of Technology (IIT), Delhi in 1993 and Ph. D. from Institute of Technology (IIT), Roorkee, India in 2003. He holds First position in Merit List of Shivaji University, India for B. Tech. In Computer Science & Engineering. He joined CSE Department at NIT, Hamirpur, Himachal Pradesh, India in August 1988 as lecturer. He has more than 23yrs teaching experience. Recently he has joined as Director of Atal Bihari Govt. Engineering College Pragati Nagar, Himachal Pradesh, India. He is member of many National bodies as expert, such as National Board of Accreditation (NBA), All India Council of Technical Education (AICTE), India. His research area includes Checkpointing, Grid Computing, Mobile Computing and cloud computing. Under his guidance, five students have completed their PhD degree, and other five are pursuing their Ph. D. He has guided many M. Tech and B. Tech students for their dissertation/projects. He has published and presented more than 154 papers in National /International Journals/Conferences and book chapters. He is reviewer of many reputed International journals like IEEE, Taylor and Frances, Inderscience etc. He is Life Member of Computer Society of India (CSI), Indian Society for Technical Education (ISTE), Fellow Member of The Institution of Engineers (India) and Senior Member of IEEE and member of International Association of Computer Science and Information Technology (IACSIT), International Association of Engineers (IAENG), Internet Society.

