A Logical XOR Operation for NTRU Cryptosystem

Bhanu Pratap Tripathi
Department of Mathematics
Govt.N.P.G.College of Science
Raipur (C.G.) 492010 India

Khushboo Thakur
School of Studies in Mathematics
Pt.Ravishankar Shukla University
Raipur-492010 (C.G.)

ABSTRACT
The NTRU public key cryptosystem was first presented by J. Hoffstein, J. H. Silverman and J. Pipher in 1996. This system is based on shortest and closest vector problem in a lattice and operations based on objects in a truncated polynomial ring. In this paper we propose new variant of NTRU cryptosystem which is based on logical exclusive OR operator. This system works under the same general principles as that of the NTRU cryptosystem except the logical operators "exclusive OR" with the different bit size for encryption and decryption which are used in place of truncated polynomial in NTRU cryptosystem. We also calculate the time complexity which shows that this system is faster than NTRU cryptosystem.

Keywords
NTRU, logical operator, Boolean function, Encryption, Decryption.

1. INTRODUCTION
NTRU is a public key cryptosystem presented by J. Hoffstein, J. pipher and J. Silverman [1]. The first version of the NTRU encryption system was presented at the crypto 96 conference; [1]. The computational basis of the NTRU lies in polynomial algebra. Two different modulo are used for reduction of polynomials as fundamental tools which is significantly speeds up its main competitors RAS and ECC. Polynomial algebra is the basic building block of the NTRU Encryption system. The truncated polynomials[2,3], in the ring $R = Z[x] / (X^n - 1)$ are basic objects and the reduction of polynomials with respect to relatively prime modulo i.e., p and q are the basic tool. NTRU polynomials a(x) are frequently reduced modulo p and q, the small and large modulo. The large modulus q is an integer, so reduction of $a(x) = a_0 + a_1 x + a_2 x^2 + \ldots\ldots\ldots\ldots a_{N-1} x^{N-1}$ 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 i.e. gcd (p, q) = 1. 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 * fp = 1 (mod p) and f * fq= 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. Firstly Bob 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) [fp] and inverse of f (mod q) [fq], where p and q are relatively prime to each other. He then computes h = p * fq * g (mod q), which becomes the public key for Alice the pair of polynomials f and fp forms his private key pair. The message is also represented in the form of a truncated polynomial. Let it be m. The sender Alice encrypts using the public key 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 = fp * b (mod p) and c is the original message:
c = fp * [f * (p * fq * g * r + m) (mod q)](mod p)
c = m using the identities f * fp = 1 and f * fq = 1

2. PROPOSED ALGORITHM
We extend the NTRU approach using XOR operation we give a brief introduction to logical operation [4, 5] on binary number.

2.1. General Description:
We first present the logical operation on binary code. Let a and b $\in \{0,1\}$.Then their bit is defined as:

a=1010 and b=1111.
By the binary operation property therefore we get,

1) $Y = \bar{a}b + a\bar{b} = 0101$ [variation on ExOR]

2) $Y = ab + \bar{a}\bar{b} = 1010$ [variation on ExOR]

(3) Y = 1 =1111 [constant]
(4) Y = 0 =0000 [constant]
(5) Y = $\bar{a}$ = 0101 [Inversion]
(6) Y = a = 1010 [No Inversion]
(7) Y = a $\oplus$ b = 0101 [XOR]

Normally the identities of boolean algebra are as follows:

- X $\oplus$ X=0 [Inverse Law]
- X $\oplus$ $\bar{X}$ =1 [Inverse Law]
- X $\oplus$ 0 = X [Identity law]
- X $\oplus$ 1 = $\bar{X}$ [Complement law]
- X $\oplus$ Y = Y $\oplus$ X [Commutative law]

Next , the algorithms for key generation, encryption and decryption following the above logical operation we give as follows:

Key Generation
Step1: Bob randomly chooses two binary code f and g where the binary code f and g is private. He consider,
f = X $\oplus$ X and g = X $\oplus$ $\bar{X}$, Since X= A $\oplus$ B
Where A and B is binary code of decimal.

Step2: Bob next step is to compute the inverse of f mod q which is fq and the inverse of f mod p which is fp .Thus,

$f \oplus fq = 1 \pmod q$

$f \oplus fp = 1 \pmod p$

and

$f * fq = 0 \pmod q$

$f * fp = 0 \pmod p$

Step 3: Now Bob computes the product $H = p \oplus fq \oplus g \pmod q$. Where fq and g is private key and public key is H.

Encryption:

Step 1: Alice wants to send a message to Bob using Bob's public key H. She first put her message in the form of binary code M and its size is same as private key f and g.

Step 2: To create the encrypted message, Alice chooses a Random binary code of decimal $R = A \cdot 0$, where $A .0 = 0$ [Law of Intersection]

Step 3: Next Alice computes the encrypted message using R and Bobs public key as follows.

$$E = R \oplus H \oplus M \pmod q$$

The binary code E is the encrypted message which Alice sends to Bob.

Decryption

Step 1: Now Bob where receives the Alice's encrypted message E and decrypt it. He begins by using his private binary code f to compute the binary code. $A = f \oplus E \pmod q$.

Step 2: Next, Bob next computes the binary code $B = A \pmod p$.

Step 3: Finally B is decrypted cipher text which should be equal to original message M.

3. RESULT

To Verify the proof of correctness we have to show that B = M so that its correctness can be prove as the same message received by the receiver send by the sender. For the steps for verification are as below:

$B = A \pmod p$ [since $A = f \quad E \pmod q$]

$B = f \oplus E \pmod q \pmod p$

$B = [f \oplus R \oplus H \oplus M \pmod q] \pmod p$ [since $E = R \oplus H \oplus M \pmod q$]

$B = [(f \oplus R \oplus p \oplus fq \oplus g \oplus M) \pmod q] \pmod p$

[Since $H = p \oplus fq \oplus g \pmod q$]

$B = [(p \oplus f \oplus fq \oplus R \oplus 1 \oplus M \pmod q)] \pmod p$ [since g=1]

$B = [(p \oplus 1 \oplus R \oplus 1 \oplus M] \pmod p$

$B = [(p \oplus 1 \oplus 1 \oplus R \oplus M] \pmod p$

$B = [(p \oplus 0 \oplus R \oplus M] \pmod p$ [Since $(1 \oplus 1 = 0)$]

$B = [(p \oplus R \oplus M] \pmod p$

$B = [p \pmod p \oplus (R \oplus M) \bmod p]$

$B = [0 \oplus R \oplus M] \pmod p$ [Since $p \pmod{p=0}$]

$B = (R \oplus M) \bmod p$

$B = (A.0 \oplus M) \bmod p$ [Since R=A.0]

$B = (0 \oplus M) \bmod p$ [Since A.0=0]

$B = M \bmod p$

4. EXAMPLE

Example of key generation, encryption and decryption for proposed design with randomly choose value is as follows.

Key generation:

Let A=10010011, B=10010100. Let parameters p = 91, q = 127

$X = [10010011 \oplus 10010100]$

$X = 00100111$

$f = X \oplus X$

$f = 00000000$

$g = X \oplus \overline{X}$

$g = 00100111 \oplus 11011000$

$g = 11111111$

$fq = \overline{f} \pmod{127}$

$fq = 11111111 \pmod{127}$

$fq = 00000001$

So,

$f \oplus fq = 000000000 \oplus 00000001$

$f \oplus fq = 00000001$

Now Bob generates the public key H as

$H = 01011011 \oplus 00000001 \oplus 00000001$

$H = 01011011 \oplus 00000000$

$H = 01011011$

Bob's private key is the pair of binary bit f and fq and his public key is H.

Encryption:

Now, suppose Alice want to send the message M to bob by using bob's public key.

$M = 00000001$

Let, $R = A .0$

$R = 00000000$

Therefore encrypted message $E = (R \oplus H \oplus M) \pmod q$ is computed as

$E = [00000000 \oplus 01011011 \oplus 00000001] \pmod{127}$

$E = [00000000 \oplus 01011100] \pmod{127}$

$E = 01011100 \pmod{127}$

$E = 01011100$ Thus, Alice sends this encrypted message E to Bob.

Decryption:

Bob has received the encrypted message from Alice. He uses his private key f to compute

$A = [f \oplus E] \pmod q$

$A = [00000000 \oplus 01011100] \pmod{127}$

$A = 01011100 \pmod{127}$

$A = 01011100$

Since Bob has computing A (mod q). So, Bob decrease the coefficients of A (mod p), we get

$B = A \pmod p$

$B = [01011100] \pmod{91}$

$B = 00000001$

$B = M$

5. TIME COMPLEXITY CALCULATION

The time complexity of an algorithm calculates the amount of time taken by an algorithm to run as a function of the length of the string representing the input. It is commonly expressed by big O notation, which excludes coefficients and lower order terms. In our paper the time complexity of binary to decimal, decimal to binary and binary to string conversion is $O(\log_2 N)$. Time complexity of length of binary string is O(N) because i varies from 1 to N. Also the time complexity of inverse of any binary number is O(1). The time complexity of addition of two binary number is O(N), Multiplication of two binary number is $O(N^2)$ Therefore the total time complexity of our scheme is O(log2(N)) + O(N) + O(1) + $O(N^2) = O(\log_2(N))$

6. CONCLUSION

This paper propose a method, which is suitable to send large messages in the form of binary number and this method is more secure since binary number are only 0 and 1. As there is no method to know whether a coefficient of polynomial is only 0 and 1 we can use this method. Because a logical operator is bit value when its binary number is found.

7. REFERENCES

[1] J. Hoffstein, J. Pipher and J. H. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem". Algorithmic Number Theory (ANTS III), Springer- Verlag, 1998, pp. 267-288.

[2] J. Hoffstein, D. Lieman, J. Silverman "Polynomial Rings and Efficient Public Key Authentication", Proceeding of the International Workshop on Cryptographic Techniques and E-Commerce, 1999.

[3] P. Prapoorna Roja, P.S. Avadhani. and E .V .Prasad, "An Efficient Method of Shared Key Generation Based on Truncated Polynomials". IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.(8B), 2006, pp. 156-161.

[4] Steven G., and Paul H.,"Introduction to Boolean Algebras". Springer-Verlag, 2009.

[5] Whitesitt Eldon J. "Boolean Algebra and its Application". Springer, 2010.