Efficient Implementation of NTRU Cryptography using Residue Number System

Azin Zalekian
Faculty of Marine Engineering, Khorramshahr University of Marine Science and Technology, Iran

Mohammad Esmaeildoust
Faculty of Marine Engineering, Khorramshahr University of Marine Science and Technology, Iran

Amer Kaabi
Faculty of Marine Engineering, Khorramshahr University of Marine Science and Technology, Iran

ABSTRACT
The NTRU cryptography is a lattice-based public key cryptography. Encryption and decryption process in NTRU are based on polynomial multiplication. This property makes NTRU to be very fast compared to other public key cryptography algorithm such as elliptic curve cryptography and RSA. In order to fast implementation of NTRU, hardware implementation of NTRU by employing residue number system is presented. To achieve high speed implementation, balanced three moduli set $\{2^n, 2^{n+1}\text{-}1, 2^n\text{-}1\}$ is considered and the encryption and part of decryption process are implemented by considered RNS bases. The result shows the noticeable improvement compared to original NTRU cryptography.

General Terms
Public Key Cryptography, NTRU cryptography, Residue Arithmetic, Residue Number System

Keywords
NTRU cryptography, Residue number system, Reverse converter, Forward converter

1. INTRODUCTION
The NTRU Public-Key Cryptosystem [1] is a ring-based cryptosystem. NTRU is a relatively new cryptosystem that appears to be more efficient than the current and more widely used public-key cryptosystems, such as RSA [2] and elliptic curve cryptography [3-5]. Compared to RSA, NTRU requires less operation compared to RSA, approximately $O(N^2)$ operations and a key length of $O(N)$, whereas RSA requires $O(N^3)$ operations and a key length of $O(N^2)$ [1]. Therefore, lower complexity of NTRU and smaller key size make it as a good option for modern cryptography. NTRU Sign features high speed, low memory requirements, simple key generation, and like NTRU encryption scheme [6]. NTRU security is stand on the hard problem of solving the approximate shortest (or closest) vectors in a certain lattice, called NTRU lattice [1].

NTRU has higher speed and security level compared to other public key cryptography algorithm [1]. However, using a way to achieve higher speed in implementation is one of the challenging processes for the researchers. Residue Number System (RNS) is a non-weighted number system, which arithmetic operation like addition, multiplication and subtraction can be done over small moduli [7]. Therefore arithmetic operations on large numbers are divided into operation over small moduli. Employing RNS in implementation of NTRU encryption and decryption can lead to high speed implementation of these processes without losing security level. In this paper, balanced three moduli set $\{2^n, 2^{n+1}\text{-}1, 2^n\text{-}1\}$ [8] are considered for the implementation of NTRU cryptography. This moduli set is balanced and using well-formed modulus in the form of 2^n and 2^n-1 results in speed up the process of encryption and decryption [7]. According to IEEE P1363.1 [9], polynomials operation over modulus q = 2048 are required in the process of encryption and decryption of NTRU. By considering $n = 4$ in moduli set $\{2^n, 2^{n+1}\text{-}1, 2^n\text{-}1\}$, the operations in modulus q with 11 bit length are simplified to operation over moduli set $\{2^4, 2^5\text{-}1, 2^4\text{-}1\}$. This lead to reduction of operation from 11 to 5 bit length in critical path.

This paper is organized as follows: Related background of NTRU and RNS will be presented in section 2. Section 3 includes the implementation of NTRU with the considered RNS moduli set. Section 4 presents the performance comparison with the state-of-the-art works and finally section 5 concluded the paper.

2. RELATED BACKGROUND
This section presents a brief review of NTRU cryptography and RNS mathematics.

2.1 NTRU Cryptography
NTRU is defined in terms of operations on the set R of polynomials of degree less than N and having integer coefficients. The basic operations on these polynomials are addition and convolution multiplication [10]. Convolution multiplication $*$ of two polynomials $F = F_0 + F_1\,x + F_2x^2 + \cdots + F_ix^{N-1}$ *and* $G = G_0 + G_1x + G_2x^2 \ldots + G_kx^{N-1}$ is defined by the following formula [1]:

$$F * G = H$$
$$H_K = \sum_{i=0}^{k} F_iG_{k-i} + \sum_{i=k+1}^{N-1} F_i\ G_{N+k-i} = \sum_{i+j\equiv k\ (mod\ N)} F_iG_j \quad (1)$$

Where N is Dimension of the polynomial ring used (i.e. polynomials are up to degree N-1). According to IEEE P1363.1 [9], recommended value for N is based on table 1.

2.1.1 Key Generation
For the public key, the user must:

1- Choose a secret key, a random secret polynomial $f \in R$, with coefficients reduced modulo p.

2- Choose a random polynomial $g \in R$, with coefficients reduced modulo p, and compute the inverse polynomial F_q of the secret key f modulo q.

Table 1. Recommended Value of *N* for different security level [9]

N	Q	Known strength	Recommended security level
401	2048	154.88	112
541	2048	141.766	112
659	2048	137.861	112
449	2048	179.889	128
613	2048	162.385	128
761	2048	157.191	128
653	2048	276.736	192
887	2048	245.126	192
1087	2048	236.586	192
853	2048	376.32	256
1171	2048	327.881	256
1499	2048	312.949	256

The procedures of key generation, encryption, and decryption of NTRU [1-11] are briefly outlined below.

Once the above has been completed, the public key, *h*, is found as

$$h = F_q * g \ mod \ q \quad (2)$$

2.1.2 Encryption

The encrypted message is computed as

$$e = p \, . r * \ h + m \ (mod \ q) \quad (3)$$

Where the message, m ϵ *R*, and the random polynomial, $r \in R$ has coefficients reduced modulo p.

2.1.3 Decryption

The decryption procedure requires three steps:

$$a = f * e \ \ mod \ q \quad (4)$$

Shift Coefficients of a to the range $\left(\frac{-q}{2}, \frac{q}{2}\right)$

$$d = F_p * a \ (mod \ p) \quad (5)$$

The last step of decryption requires the user to compute the inverse polynomial F_P of the secret key *f* modulo *p*. The decryption process outlined above will recover the original message $(d = m)$.

2.2 Residue Number System

The RNS is a non-weighted number system which is defined in terms of relatively-prime moduli set $\{m_1, m_2, \dots, m_n\}$ that is $\gcd(m_i, m_j) = 1$ for $i \neq j$. For a weighted number *X*, RNS representation is $(x_1, x_2, \dots, x_n)$ where:

$$x_i = X \ mod \ m_i = |X|_{m_i}, 0 \leq x_i \leq m_i \quad (6)$$

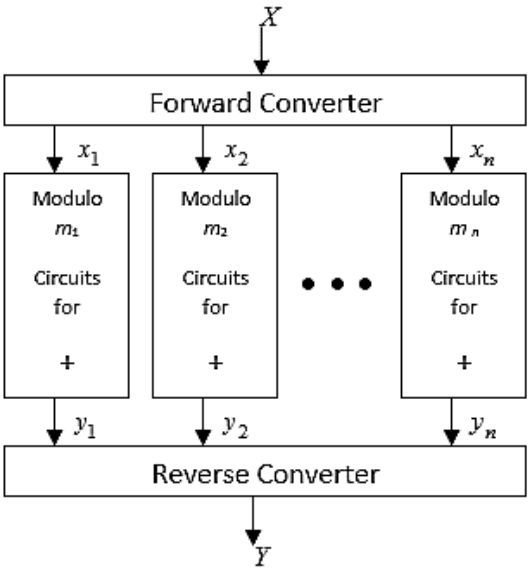

Fig. 1. Block diagram of a typical RNS system [7]

Such a representation is unique for any integer *X* in the range [0, *M*-1]. *M* is the dynamic range of the system equal to the product of m_i terms $M = m_1 m_2 \dots m_n$ [12]. RNS has three main parts which is shown in figure 1. As shown in figure 1, the weighted numbers are converted into residue representations by a binary-to-residue converter (forward converter). Then, arithmetic operations such as addition, subtraction and multiplication are performed on RNS numbers in parallel without carry propagation between residue digits. In order to use the result of arithmetic operations, residue numbers should be converted into its corresponding weighted binary number by using a reverse converter [7].

3. HARDWARE IMPLEMENTATION OF NTRU USING RNS

Encryption and decryption in NTRU cryptography will be presented in this section. For simplicity encryption process is rewritten in Eq. (7).

$$e = p.r * h + m \ (mod \ q) \quad (7)$$

Where, *m* is plaintext, *r* is a random polynomial, *h* is public key, and *p* and *q* are equal to 3 and 2048 according to IEEE P1363.1™, respectively [9]. According to Eq. (7), the encryption process consists of multiplication of two polynomials and also multiplication of constant *p* in a polynomial. *n* modular multiplications are needed for multiplication of a polynomial of degree *N* in a constant *p*.

Numbers of multiplication of two polynomials $A = a_0 + a_1 x + a_2 x^2 + \dots + a_j x^{N-1}$ and $B = b_0 + b_1 x + b_2 x^2 \dots + b_k x^{N-1}$ in modulo *q* can be calculated according to Eq. (8) [10].

$$r_i = \left(c + \sum_{j+k=i} a_j \, b_k\right) \bmod N \quad j \in \{0, 1, \dots, N-1\}, \ k \in \{0, 1, \dots, N-1\} \quad (8)$$

Where *N* is equal to 2048 [9]. Eq. (8) needs O (N^2) multiplication in modulo *q*.

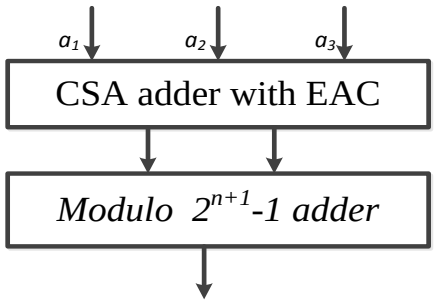

Figure 2. Forward converter for modulus 2^{n+1}-1

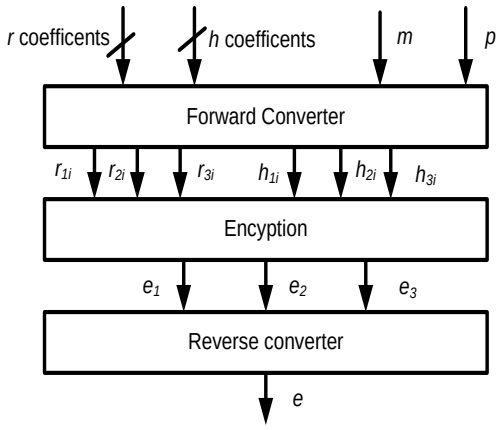

Fig. 3. Required steps for encryption using RNS moduli set

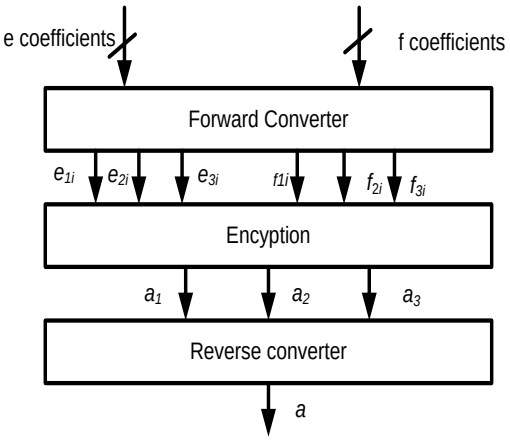

Fig. 4. Required steps for decryption using RNS moduli set

In order to reduce the modular operations, moduli set {2^n, 2^{n+1}-1, 2^n-1} is considered to implement Eq. (7). Figure 2 shows the three steps required in implementation of Eq. (7) using considered RNS moduli set. Forward converter for a number with $3n$ bit in modulus 2^n and 2^n-1 are presented in [7]. In critical moduli 2^{n+1}-1 it can be calculated as:

$$|(X_{3n-1}, \dots, X_{2n+3})2^{2n+2} + (X_{2n+2}, \dots, X_{n+1})2^{n+1} + (X_n, \dots, X_0)|_{2^{n+1}-1} \quad (9)$$

Eq. (9) can be rewritten as:

$$|a_1 + a_2 + a_3|_{2^{n+1}-1} \quad (10)$$

Where

$$a_3 = X_{3n-1}, \dots, X_{2n+3}$$

$$a_2 = X_{2n+2}, \dots, X_{n+1}$$

$$a_1 = X_n, \dots, X_0$$

The forward conversion in modulo 2^n, 2^{n+1}-1 and 2^n-1 are done in parallel channels. By considering modulus 2^{n+1}-1 as critical moduli, in worse case for forward conversion according to figure 2, delay of carry save adder (CSA) adder with end around carry (EAC) followed by modulo 2^{n+1}-1 adder is resulted.

The delay and area of reverse converter for moduli set {2^n, 2^{n+1}-1, 2^n-1} [8] are included in table 2.

Table2. Delay and area of revere converter for moduli set {2^n, 2^{n+1}-1, 2^n-1} [8]

Converter	Delay	Area
{2^n, 2^{n+1}-1, 2^n-1}-converter II [8]	$(2n+7)$ D_{FA}	$(14n+21)A_{FA}$ + $(2n+3)A_{HA}$ +$(2n+1)A_{MUX}$

Since q is equal to 2048 (2^{11}) [9], by considering n = 4 in moduli set {2^n, 2^{n+1}-1, 2^n-1}, the required dynamic range 2048 can be covered by moduli set {2^4, 2^5-1, 2^4-1}. Therefore in order to implement NTRU cryptography polynomials r and h are calculated in moduli {2^4, 2^5-1, 2^4-1} as shown in figure 3.

In order to calculate the coefficients $r = r_0 + r_1x + \cdots + r_{n-1}x^{N-1}$ and $h = h_0 + h_1x + \cdots + h_{n-1}x^{N-1}$ into residues, a pipeline manner is considered and the coefficients are inserted as an input of forward converter sequentially. Once a coefficient is converted into residues, the process of encryption will be started. Therefore forward conversion process does not reduce the efficiency of the system.

Figure 3 shows the required steps for Encryption. As mentioned before encryption in NTRU required 11-bit modular operations. As shown in figure 3, it can be seen that by reducing the 11-bit modular operations into 5-bit noticeable improvement in speed of encryption is achieved. Using moduli set {2^n, 2^{n+1}-1, 2^n-1} modular operations are simplified in to 5-bit operation in critical moduli.

In the decryption process, calculation of $a = f.e \bmod q$ is required. Calculation of $a = f.e \bmod q$ can be done with same way employed for encryption. Figure 4 shows the required step decryption using RNS moduli set.

4. PERFORMANCE COMPARISON

In this section the performance comparison with original NTRU implementation will be presented by using 11-bit modular multiplication. In order to multiply two polynomial $a = a_0 + a_1x + \cdots + a_{n-1}x^{N-1}$ and $b = b_0 + b_1x + \cdots + b_{n-1}x^{N-1}$ in modulo q = 2048, N^2 modular multiplications are required in worse case. In this report moduli set {2^n, 2^{n+1}-1, 2^n-1} is employed with n=4. Therefore, 11-bit operations

are reduced to 5-bit in critical moduli channel. Table 3 shows the required operations for polynomial multiplication in modulo q with and without using RNS. It can be seen that the operation of two polynomial multiplications are reduced from 11 to 5-bit modular multiplication. In table 3, the delays of forward and reverse conversion are calculated by assumption considered in [8].

Table 3: Comparison of polynomial multiplication

References	Required operation	Delay (critical path)
[1]	N^2 modulo q Multiplication	$N^2(D^*_{\text{modulo } 2^{11}\text{ multiplication}})$
proposed	N^2 moduli $\{2^4, 2^5\text{-}1, 2^4\text{-}1\}$ multiplication, forward and reverse conversion	$N^2\,(D_{\text{modulo } 2^5\text{-}1 \text{ multiplication}})+(30)D_{FA}$ **

*D denotes the delay of unit
**D_{FA} denotes the delay of full adder gate

Table 4: Multiplication of a constant in polynomial

	Required operation	Delay (critical path)
[1]	N modulo q Multiplication	$N\,(D_{\text{modulo } 2^{11}\text{ multiplication}})$
proposed	N moduli $\{2^4, 2^5\text{-}1, 2^4\text{-}1\}$ multiplication, forward conversion, Reverse conversion	$(D_{\text{modulo } 2^5\text{-}1 \text{ multiplication}})+(30)D_{FA}$ *

Table 4 shows the required operations for multiplication of a constant in polynomials. As mentioned before, encryption process needs polynomial multiplication and multiplication of a constant in a polynomial in modulo q. Decryption process also needs multiplication of two polynomials in modulo q. Table 5 shows the operations required for encryption and decryption and also the delay of critical path. As it can be seen, operations modulo 2048 are reduced to operations over modulo 2^5-1 in critical path. As reported in table 1, large values for N are considered in NTRU cryptography. Therefore, reduction of modulo 2^{11} operations to 2^5-1 will results in noticeable improvement in speed of implementation of algorithm.

5. CONCLUSION

In this paper, hardware implementation of NTRU cryptography using residue number system is presented. To achieve high speed implementation, balanced three moduli set $\{2^n, 2^{n+1}\text{-}1, 2^n\text{-}1\}$ is considered and the encryption and decryption process of NTRU cryptography are implemented by using this RNS system. By employing this moduli set, operation over modulo 2^{11} are reduced to operation over 2^5-1 which results in noticeable improvement in speed of NTRU implementation.

Table 5: Encryption and decryption operations and delay in modulo q

References	Required operation	Delay (critical path)
[1]	$(2N^2+N)$ modulo q Multiplication + N modulo q addition	$(2N^2+N)\,(D_{\text{modulo } 2^{11}\text{ multiplication}}) + N\,(D_{\text{modulo } 2^{11}\text{ addition}})$
proposed	$(2N^2+N)$ moduli $\{2^4, 2^5\text{-}1, 2^4\text{-}1\}$ multiplication, N moduli $\{2^4, 2^5\text{-}1, 2^4\text{-}1\}$ addition, forward conversion, Reverse conversion	$(2N^2+N)\,(D_{\text{modulo } 2^5\text{-}1 \text{ Multiplication}}) + N\,(D_{\text{modulo } 2^5\text{-}1 \text{ addition}})+(30)D_{FA}$

6. REFERENCES

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

[2] R. L. Rivest, A. Shamir, and L. Adelman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" Communications of the ACM, vol. 21, pp. 120-126, February 1978

[3] N. Koblitz,"Elliptic curve cryptosystems", Mathematics of Computation. vol. 48, no. 177, pp.203 -209, January 1987.

[4] V. S. Miller, "Use of elliptic curves in cryptography", Proceeding Advanced in Cryptology – CRYPTO '85 Proceedings, Lecture Notes in Computer Science Vol.218 , pp.417 -426 1986.

[5] M. Esmaeildoust, D. Schinianakis, H. Javashi, T. Stouraitis, K. Navi, "Efficient RNS Implementation of Elliptic Curve Point Multiplication Over GF (p)," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol.21, no.8, pp.1545, 1549, Aug. 2013.

[6] Efficient NTRU Implementations: A Thesis Submitted to the Faculty of the Worcester Polytechnic Institute In partial fulfillment of the requirements for the Degree of Master of Science in Electrical & Computer Engineering by Colleen Marie O'Rourke - April 2002

[7] K. Navi, A. S. Molahosseini, and M. Esmaeildoust, "How to teach residue number system to computer scientists and engineers?" IEEETrans. Edu, vol.54, no. 1, pp. 156–163, Feb. 2011.

[8] P. V. A. Mohan, "RNS-To-Binary Converter for a New Three-Moduli Set $\{2^{n+1}-1, 2^n, 2^n-1\}$," *IEEE Transactions on Circuits and Systems-II*, vol. 54, no. 9, pp. 775-779, 2007.

[9] IEEE P1363.1/D10 - Draft Standard for Public-Key, Cryptographic Techniques Based on Hard Problems over Lattices.

[10] Performance Improvements and a Baseline Parameter Generation Algorithm for NTRUSign J. Hoffstein, Nicholas Howgrave-Graham, Jill Pipher, Joseph H. Silverman, William Whyte NTRU Cryptosystems.

[11] J. Hoffstein and J. H. Silverman, Optimizations for NTRU, in Proceedings of Public Key Cryptography and Computational Number Theory, de Gruyter, Warsaw, September 2000. F.J, Taylor, "Residue arithmetic: a tutorial with examples," *IEEE Computer*, vol. 17, pp. 50–62, 1984.

[12] F.J, Taylor, "Residue arithmetic: a tutorial with examples," *IEEE Computer*, vol. 17, pp. 50–62, 1984.