International Journal of Innovative Computing 7(1) 33-38

UTM UNIVERSITI TEKNOLOGI MALAYSIA
RESEARCH UNIVERSITY

International Journal of Innovative Computing

Journal Homepage: http://se.fsksm.utm.my/ijic/index.php/ijic

ITRU: NTRU-Based Cryptosystem Using Ring of Integers

Juliet Nyokabi Gaithuru
Faculty of Computing
Universiti Teknologi Malaysia
Skudai 81310 Johor
Malaysia
julietgaithuru@yahoo.com

Mazleena Salleh
Faculty of Computing
Universiti Teknologi Malaysia
Skudai 81310 Johor
Malaysia
mazleena@utm.my

Ismail Mohamad
Faculty of Science
Universiti Teknologi Malaysia
Skudai 81310 Johor
Malaysia
ismailm@utm.my

***Abstract*— NTRU is a public key cryptosystem whose structure is based on the polynomial ring of integers. We present ITRU, an NTRU-like cryptosystem based on the ring of integers. We discuss the parameter selection procedure and provide an implementation of ITRU using an illustration. A comparison of the performance of ITRU and NTRU is provided which highlights the difference in parameter selection, invertibility and successful message decryption. We show that ITRU is an improvement of NTRU in that, it ensures successful message decryption upon implementation using the proposed parameter selection algorithm.**

Keywords — Cryptography, NTRU, integer ring.

I. INTRODUCTION

NTRU (N^{th} Degree Truncated Polynomial Ring) is a lattice-based public key cryptosystem which was presented by Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman in the rump session at CRYPTO '96 in 1996 and was published in 1998 [1]. It was later standardized as IEEE 803.11 (2008), ANSI X9.8 and EESS 1v2 (2003) and EESS 1v3(2015) [2-4]. NTRU operates in the polynomial ring with integer coefficients. Due to its lattice-based structure, the security of NTRU is based on the difficulty in solving the closest-vector problem, which is considered to be NP-hard. NTRU is a fast due its convolution multiplication operation involving small coefficients. However, NTRU has a problem of decryption failure whose probability can be reduced by selecting parameters which have a small probability of decryption failure. NTRU is resistant to quantum algorithm attacks, as opposed to other public key cryptosystems such as ECC and RSA.

In order to further enhance the security of NTRU, research has been conducted on other variants of NTRU. Some variants propose the use of polynomial rings with coefficients in other rings. In 2002, Gaborit suggested the use of the ring of polynomials instead of the ring of integers and presented CTRU [5]. Kouzmenko suggested the use of Gaussian Integers and presented GTRU in 2006 [6].

Other variants use alternative rings. In 2005, Coglianese & Goi suggested the use of matrices and presented MaTRU [7].

In 2011 and 2015 Jarvis and Nevins suggested the use of the ring of Eisenstein integers and presented ETRU [8, 9]. In 2009 Malekian *et al.* suggested the use of the ring of Quaternions and presented QTRU in 2015 [10, 11].

Other NTRU variants use varying commutative structures. In 2002, Banks presented a variant of NTRU which uses non-invertible polynomials [12]. In 2003, Rourke and Sunar presented a variant of NTRU which uses Montgomery multiplication [13]. In 2007, Truman presented the use of a non-commutative NTRU [14].

Furthermore, work by [15] presents a simplified version of NTRU referred to as mini-NTRU, which provides a generalized parameter selection criteria and reduced parameter sets which foster understanding of the NTRU public key cryptosystem.

The goal of this study is to present a variant of NTRU which is based on the ring of integers as opposed to using the polynomial ring with integer coefficients. We show that NTRU based on the ring of integers (ITRU), has a simple parameter selection algorithm, invertibility and successful message decryption. We describe a parameter selection algorithm and also provide an implementation of ITRU using an example. ITRU is shown to have successful

message decryption, which provides more assurance of security in comparison to NTRU.

The paper is organized as follows. We begin with a description of the mathematical background of the NTRU lattice structure in Section II. Furthermore, we described the structure of the classical NTRU public key cryptosystem along with the parameter selection criteria. In Section III, we present the theoretical model of ITRU, a variant of NTRU using the ring of integers. A parameter selection criteria for ITRU is also presented. This is followed by an example of ITRU's implementation in Section IV. In Section V, a performance comparison of ITRU and NTRU is provided. Finally, in Section VI, we present a conclusion and discuss avenues for future work.

II. RELATED BACKGROUND

A. Mathematical Background

NTRU is a lattice-based cryptosystem. A lattice is a regular arrangement of points in Euclidean space. Given the following set of linearly independent vectors $v_1, v_2, v_3, \ldots, v_k$ in R^N, the lattice $\mathcal{L}$ is the set of linear combinations of $v_1, v_2, v_3, \ldots, v_k$ with coefficients in $\mathbb{Z}$ (integers). Therefore, the lattice $\mathcal{L}$ is given by

$$\mathcal{L} = \{a_1 v_1 + a_2 v_2 + a_3 v_3 + \cdots + a_n v_n | a_1, a_2, \ldots, a_k \in \mathbb{Z}\} \quad (1)$$

where k is the rank and N is the dimension of lattice $\mathcal{L}$ [16, 17]. The basic computational problems in lattices are; the shortest vector problem, abbreviated as SVP, and the closest vector problem, abbreviated as CVP. The SVP is the problem of finding a shortest non-zero vector v in a lattice L that minimizes the Euclidean norm $\|v\|$ while the CVP refers to the problem whereby given a vector $w \in \mathbb{R}^m$ which is not in L, find a vector $v \in L$ that is closest to w that minimizes the Euclidean norm $\|w - v\|$ [16]. The CVP is known to be NP-hard while the SVP is NP-hard under certain "randomized reduction hypothesis". A solution to the SVP could be used to break various cryptosystems [16].

B. NTRU Cryptosystem

NTRU (N^{th} Degree Truncated Polynomial Ring), pronounced as *en-tru*, is a lattice-based public key cryptosystem whose structure is based on polynomial convolution rings. A ring is a set of elements which is closed under addition and multiplication operations which are associative and commutative. In a ring, an additive inverse and a multiplicative identity exists. The security of NTRU is based on the difficulty in solving the SVP or CVP.

The parameters used in NTRU include a parameter size N (where $N \geq 1$ and N is prime), two moduli p and q which are relatively prime. The polynomial convolution rings which are based on these three parameters are given by:

$$R = \frac{\mathbb{Z}[x]}{(x^N - 1)} \quad R_p = \frac{(\mathbb{Z}/p\mathbb{Z})[x]}{(x^N - 1)} \quad R_q = \frac{(\mathbb{Z}/q\mathbb{Z})[x]}{(x^N - 1)} \quad (2)$$

Polynomials in R_p and R_q have their coefficients reduced moduli p and q respectively. The parameter size N implies that there are at most N terms and that all polynomials have a maximum degree of $(N-1)$ thus all polynomial products are reduced modulo $(x^N - 1)$. Multiplication in the polynomial convolution ring is denoted by *. NTRU has various variants: binary, ternary (or trinary) and product form polynomials [1, 16].

TABLE 1: NTRU PARAMETERS, KEY GENERATION, ENCRYPTION AND DECRYPTION

	Public parameters	**Private parameters**	**Condition for selection**
	A sender/trusted party selects N, p, q, d	$f(x), g(x)$	$\gcd(N, q) = \gcd(p, q) = 1$. $q > (6d + 1)$. $f(x)$ is in $T(d+1, d)$. $g(x)$ is in $T(d, d)$.
Key creation		**Receiver:** Selects $f(x)$. Computes $F_p(x)$ and $F_q(x)$.	$f(x) * F_q(x) = 1\ (mod\ q)$ in R $f(x) * F_p(x) = 1\ (mod\ p)$ in R
	Receiver: Computes the public key $h(x) = (p * F_q(x) * g(x))\ mod\ q$ and makes it public.		$h(x)$ is in the ring R_q.
Encryption		**Sender:** Selects plaintext message $m(x)$. Selects random parameter $r(x)$. Computes the ciphertext $e(x) = (r(x) * h(x) + m(x))$ and sends it.	$m(x)$ is in R_p. $r(x)$is in $T(d, d)$ Coefficient range of $m(x)$ is $\frac{p}{2}$ to $-\frac{p}{2}$. $e(x)$ is in the ring R_q.
Decryption	**Receiver:** Computes $a(x) = f(x) * e(x)\ mod\ q$. Centers $a(x)$ in R. Computes $C(x) = F_p(x) * a(x)\ mod\ p$. Compares $C(x)$ to $m(x)$.		$a(x)$ is in the ring R_q. $C(x)$ is in the ring R_p.

Binary polynomials have 0's and 1's as coefficients whereas the set of coefficients for ternary polynomials are expressed as $T(d_1, d_2)$ where d_1 is the number of 1's, d_2 is the number of -1's and the rest are 0 coefficients [16, 18]. Product form polynomials provide for a private key which is always invertible [18].

The parameters used in NTRU are the public parameters (N, p, q, d) where d is the number of 1 coefficients. The rest of the NTRU parameters are polynomials.

The private parameters $f(x)$ and $g(x)$ are used for private and public key generation. The parameter $f(x)$ is chosen in $T(d, d)$ and has multiplicative inverses modulo p and q. In the event that the selected parameter f is not invertible, it is discarded and a new parameter selected. The parameter $r(x)$ is a random one-time parameter which is different for each encryption operation. The message is expressed as the parameter $m(x)$. The computed parameters are the public key parameter $h(x)$, the ciphertext $e(x)$, the parameter $a(x)$ and the decrypted message $C(x)$ which is then compared to the plaintext $m(x)$. Table 1 shows the NTRU parameters, the conditions for their selection, the key generation, encryption and decryption processes.

As illustrated in the Table 1, the operation of the NTRU public key cryptosystem begins when the trusted authority/sender chooses the parameters N, p, q, d. The receiver selects the private parameters $f(x), g(x)$. After which the receiver generates the private key by computing the multiplicative inverse of f in R_p given by $F_p(x)$ resulting in the private key pair $(f(x), F_p(x))$.

Afterwards, the multiplicative inverse of f in R_q given by $F_q(x)$ is computed. Furthermore, the public key is generated by computing $h(x) = p * F_q(x) * g(x) \ mod \ q$ in R_q and the polynomial $h(x)$ is published.

What follows is the initialization of the plaintext message polynomial $m(x)$ in R whose coefficients range between $\frac{p}{2}$ and $-\frac{p}{2}$. The sender then goes on to select a random polynomial $r(x)$ which will be used for encryption. The message is then encrypted by generating the ciphertext $e(x) = ((r(x) * h(x)) + m(x)) \ mod \ q$. The ciphertext $e(x)$ is then sent to its intended receiver.

Upon receiving the ciphertext, the receiver decrypts it using the private key by computing $a(x) = (f(x) * e(x)) mod \ q$. The coefficients of $a(x)$ are then adjusted to an element of R. Finally, the plain text message is recovered by computing $C(x) = (F_p(x) * a(x)) \ mod \ p$ which is eventually verified to confirm that $C(x) = m(x)$.

III. THE PROPOSED ITRU ALGORITHM- NTRU USING RING OF INTEGERS

A. *Parameters and Notation*

The proposed ITRU cryptosystem operates in the ring of integers Z modulo n, denoted as Z/nZ. The ring Z/nZ of integers modulo n is the set of equivalence classes of integers modulo n. A ring is a set of values which are commutative, that is closed under addition (+) and multiplication (×) [19]. Addition and multiplication is carried out using regular integer operations and the result is then reduced modulo n.

The ITRU consists of integer parameters whose descriptions and compositions are shown in Table 2. The parameters are computed in the rings $Z/p'Z$ and $Z/q'Z$ where p' is set to 1000 while q' is a large prime integer.

TABLE 2: ITRU PARAMETERS

Parameter	Description
p'	Small modulus
q'	Large modulus
f'	Private integer for private key generation
g'	Private random integer for public key generation
r'	Private random integer for cipher-text generation
m'	Decimal representation of the message
K_{pr}	Private key pair $(f', F_{p'})$
K_{pb}	Public key parameter h
a'	Intermediate parameter
C'	Decrypted message

In ITRU, the parameters p', q', f', g', r' are selected using the Algorithm 1.

Algorithm 1: ITRU parameter generation

1. Initialize p'=1000
2. Randomly select an odd integer. Assign value to f'
3. Randomly select two integers. Assign values to g' and r'
4. Set q'=prime integer $>(p' \times r' \times g' + f' \times m')$
5. Compute modular multiplicative inverse ${f'}^{-1} \ mod \ p' \rightarrow F_{p'}$
6. Compute modular multiplicative inverse of ${f'}^{-1} \ mod \ q' \rightarrow F_{q'}$

As shown in Algorithm 1, the ITRU parameter selection is carried out as follows:

1) p' is set as 1000.

This is because implementation of ITRU is in the ring of integers which are decimal integers. In this study, we assuming that the message is

comprised of alphabetical characters of upper case (A-Z) or lower case (a-z), which implies that the range is 65-90 and 97-122 respectively.

2) m' is a representation of the message in decimal form.
3) f' is a private integer, which is randomly selected as an odd integer so as to have a multiplicative inverse modulo p'.
4) g' and r' are private integers which are selected as any random integer. g' is used to generate the public key while r' is used for generating the cipher-text.
5) Select a large prime q'.
q' should be set as the next large prime integer greater than $(p' \times r' * g' + f' \times m')$.

B. Key Creation

To generate the private and public key pairs, the sender takes the selected private integer f' and then computes the modular multiplicative inverse of $f' \ mod \ p'$ and $f' \ mod \ q'$ which are labelled as $F_{p'}$ and $F_{q'}$ respectively. The inverses obtained satisfy the conditions

$$f' \times F_{p'} \ mod \ p' \equiv 1 \quad (3)$$

$$f' \times F_{q'} \ mod \ q' \equiv 1 \quad (4)$$

The parameter f' selected must have an inverse modulo p' and q'. The modular multiplicative inverse is computed using the Extended Euclidean algorithm. The modular inverse algorithm utilized in this implementation of ITRU is as shown in Algorithm 2.

Algorithm 2: Finding Modular Inverse using Extended Euclidean Algorithm

1. procedure EXTENDED EUCLIDEAN(a, b)
2. Input: $t_1, \ t_2$
3. Output: r_1, d
4. Initialize
   $q = 0, \ r_1 = b, \ r_2 = a, \ r = r_1 - (q \times r_2), \ t_1 = 0, \ t_2 = 1, \ t = 0$
5. while $r \neq 0$ do
   5.1. $q = r_1 / r_2, r = r_1 - (q \times r_2)$
   5.2. $t = t_1 - (q \times t_2)$
   5.3. $r_1 \leftarrow r_2, r_2 \leftarrow r, t_1 \leftarrow t_2, t_2 \leftarrow t$
6. Return $GCD(r_1, r_2) = r_1$
7. Return $a^{-1} mod \ b = t_1$
8. If $t_1 < 0$ then
   8.1. $d = t_1 + b$
9. Else
   9.1. $d = t_1$
10. Return r_1, d

As indicated in Algorithm 2, the modular multiplicative inverse is computed using a function where the inputs into the function are (f', p') and (f', q') so as to obtain the inverse $F_{p'}$ and $F_{q'}$ respectively. The algorithm works using a series of successive divisions of r_1 by r_2 and then the product of the resulting quotient with r_2 is computed and then subtracted from r_1. The next row of values is obtained by a series of cyclic shifts to the right. In the event that obtained inverse is a negative integer, the positive modular equivalent is computed.

For instance, the function call EXTENDED EUCLIDEAN$(7, 960)$ finds the modular multiplicative inverse r_1 such that $7r_1$ mod 960=1. It should be noted that GCD$(f', p') =$ GCD$(f', q') =$ d $= 1$, that is a and b must be relatively prime in order to find a multiplicative inverse.

The sender then obtains his private key pair

$$K_{pr} = (f', F_{p'}) \quad (5)$$

The public key K_{pb} is obtained by computing

$$K_{pb} = h' = (p' \times F_{q'} \times g') \ mod \ q' \quad (6)$$

C. Encryption

To encrypt a message, the sender takes the randomly selected integer r' and the message which is converted to its decimal representative. The conversion can be done using ASCII conversion tables, in the case of alphabetical character messages.

The sender then encrypts the message m' by computing

$$e' = \left((r' \times h') + m'\right) mod \ q' \quad (7)$$

The cipher-text e' is then sent to its intended recipient.

D. Decryption

To decrypt the received cipher-text, the recipient first computes

$$a' = (f' \times e') \ mod \ q' \quad (8)$$

Finally, recovering the message by computing

$$C' = (F_{p'} \times a') \ mod \ p' \quad (9)$$

The recovered C' should be equal to m'.

E. Why Decryption Works

In the decryption step, using Eq. 7 the receiver has

$$a' = (f' \times (r' \times h' + m')) \ mod \ q'$$

$$= (f' \times r' \times h' + m') \ mod \ q' \quad (10)$$

Given that $h' = (p' \times F_{q'} \times g') \ mod \ q'$, then $a' = ((f' \times F_{q'} \times p' \times r' \times g') + (f' \times m')) \ mod \ q'$.

Since $f' \times F_{q'} \ mod \ q' \equiv 1$ then

$$a' = ((p' \times r' \times g') + (f' \times m')) \ mod \ q' \quad (11)$$

Putting into consideration that the parameter $|q'| > (p' \times r' \times g' + f' \times m')$, the value a' will evaluate to

$$a' = p' \times r' \times g' + f' \times m' \quad (12)$$

The original message is then recovered by multiplying a' with $F_{p'}$ in the ring $Z/p'Z$ resulting in

$$C' = (f' \times F_{p'} \times m') + (p \times F_{p'} \times r' \times g') \ mod \ p'$$

Considering that $f' \times F_{p'} \ mod \ p' \equiv 1$ and $|p \times F_{p'} \times r' \times g'| \ mod \ p' = 0$, the decrypted message evaluates to

$$C' = m' \quad (13)$$

Thereby recovering the original encrypted message.

IV. ITRU ENCRYPTION EXAMPLE

An implementation of the proposed ITRU is illustrated using the following example. Suppose a sender wants to use ITRU to encrypt and send the message "hi".

A. Parameter Selection

He starts off by establishing the ITRU encryption parameters. Selecting the parameters $(p', f', g', r') = (10^3, 7, 76, 19)$. The parameter q' is then set as the next large prime greater than $(p' \times r' \times g' + f' \times m')$. In this case $q'(next\ prime) > 1444742$, resulting in $q' = 1444747$.

The message m' is initialized by representing the plaintext message in decimal form using ASCII conversion tables.

$$m' = (104, 105) \tag{14}$$

B. Key creation

The private key pair is created by finding the modular multiplicative inverse $F_{p'}$ and $F_{q'}$ using Algorithm 1 resulting in

$$F_{p'} = 7^{-1}\ mod\ 1000 = 143 \tag{15}$$

$$F_{q'} = 7^{-1}\ mod\ 1444747 = 412785 \tag{16}$$

Thus, the private key pair is

$$K_{pr} = (7, 143) \tag{17}$$

While the public key is obtained by computing

$$K_{pb} = h' = (1000 \times 412785 \times 76) mod\ 1444747 = 423642 \tag{18}$$

C. Encryption

The message is encrypted and sent to its intended recipient by computing

$$e' = \big(19 \times 423642 + (104, 105)\big) mod\ 1444747 = (825567, 825568) \tag{19}$$

D. Decryption

Upon receiving the cipher text, the recipient decrypts it by computing

$$a' = \big(7 \times (825567, 825568)\big) mod\ 1444747 = (1444728, 1444735) \tag{20}$$

Then retrieves the original message by computing

$$C' = \big(143 \times (1444728, 1444735)\big) mod\ 1000 = (104, 105) \equiv$$

"hi". (21)

V. PERFORMANCE COMPARISON

The comparison of the classical NTRU [1] and our proposed ITRU cryptosystem is as follows:

- The generation of ITRU parameters requires less pre-computation, as shown in Algorithm 1. ITRU requires the random selection of 2 integers g' and r', and one odd integer f' whose inverse is then computed. Provided f' is prime and all parameters are selected in accordance with the prescribed criteria, f' is always invertible. This differs from the classical NTRU which has a more elaborate parameter selection criteria and f' is not always invertible.
- The computational complexity for key generation in the ITRU algorithm is $O(N^2)$, since it only involves the application of the Extended Euclidean algorithm. In classical NTRU, the key generation algorithm which is an adaptation of the adaptation of the "Almost Inverse Algorithm" [20] has computational complexity of $O(N^2(log^2 p + log^2 q))$ [21].
- The proposed ITRU algorithm results in successful message decryption, provided that the parameter selection criteria is adhered to. On the other hand, the classical NTRU public key cryptosystem has a probability of decryption failure of 2^{-145} [18] for the most recent version of NTRU which uses product-form polynomials.
- ITRU is based on integer rings as opposed to the lattice structure of the classical NTRU. The security of ITRU is based on the integer factorization problem.
- Evaluations conducted by running NTRU and ITRU on the Magma Computational Algebra System V2.22-7 running on a 3.60GHz i7 Windows 10 operating system showed that the key generation process in ITRU is less complex and faster. In addition, the cipher text in ITRU had a message expansion factor of 3.

VI. CONCLUSION

In this study, an NTRU-based cryptosystem using the ring of integers, referred to as ITRU is presented. The proposed algorithm uses integers as opposed to polynomials, as is the structure of the classical NTRU. The paper describes a description of the theoretical model of ITRU, the parameter selection conditions as well as an illustration of its implementation using an example. The performance of ITRU is then contrasted with that of NTRU.

This study shows that the selection of parameters in accordance with the described parameter selection algorithm ensures a successful message decryption, thereby eliminating the risk of decryption failure. However, owing to ITRU's integer structure, it is susceptible to similar attacks as those posed on encryption schemes which have integer structures such as RSA. Such attacks include the number field sieve algorithm and Pollard's Rho Method.

Further research may be conducted on the corresponding ITRU parameters for comparative security levels with the classical NTRU. This will provide opportunities to compare public key sizes and running times at equivalent security levels.

REFERENCES

[1] Hoffstein, J., J. Pipher, and J.H. Silverman, NTRU: A Ring-Based Public Key Cryptosystem, in Algorithmic Number Theory. 1998, Springer. p. 267-288.

[2] IEEE Draft Standard Specification for Public- Key Cryptographic Techniques Based on Hard Problems over Lattices. IEEE Unapproved Draft Std P1363.1/D12, Oct 2008, 2008: p. 1.

[3] IEEE, Efficient Embedded Security Standards (EESS), in EESS #1: Implementation Aspects of NTRUEncrypt and NTRUSign. 2003, Consortium for Efficient Embedded Security. p. 78.

[4] Security, C.F.E.E., Efficient Embedded Security Standard (EESS) #1 in Version 3.0, 2015,. 2015.

[5] Gaborit, P., J. Ohler, and P. Solé, CTRU, a Polynomial Analogue of Ntru. 2002, INRIA.

[6] Kouzmenko, R., Generalizations of the NTRU Cryptosystem. Diploma Project, École Polytechnique Fédérale de Lausanne,(2005–2006), 2006.

[7] Coglianese, M. and B.-M. Goi, MaTRU: A New NTRU-Based Cryptosystem, in Progress in Cryptology - Indocrypt 2005: 6th International Conference on Cryptology in India, Bangalore, India, December 10-12, 2005. Proceedings, S. Maitra, C.E. Veni Madhavan, and R. Venkatesan, Editors. 2005, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 232-243.

[8] Jarvis, K., NTRU over the Eisenstein Integers. 2011, Université d'Ottawa/University of Ottawa.

[9] Jarvis, K. and M. Nevins, ETRU: NTRU over the Eisenstein Integers. Designs, Codes and Cryptography, 2015. 74(1): p. 219-242.

[10] Malekian, E., A. Zakerolhosseini, and A. Mashatan, QTRU: A Lattice Attack Resistant Version of NTRU PKCS Based on Quaternion Algebra. preprint, Available from the Cryptology ePrint Archive: http://eprint. iacr. org/2009/386. pdf, 2009.

[11] Malekian, E., A. Zakerolhosseini, and A. Mashatan, QTRU: Quaternionic Version of the NTRU Public-Key Cryptosystems. The ISC International Journal of Information Security, 2015. 3(1).

[12] Banks, W.D. and I.E. Shparlinski. A Variant of NTRU with Non-Invertible Polynomials. in International Conference on Cryptology in India. 2002. Springer.

[13] Rourke, C.O. and B. Sunar, Achieving NTRU with Montgomery Multiplication. IEEE Transactions on Computers, 2003. 52(4): p. 440-448.

[14] Truman, K., Analysis and Extension of Non-Commutative NTRU. 2007, University of Maryland.

[15] Gaithuru, J.N., M. Salleh, and I. Mohamad. Mini N-Th Degree Truncated Polynomial Ring (Mini-NTRU): A Simplified Implementation Using Binary Polynomials. in 2016 IEEE 8th International Conference on Engineering Education (ICEED). 2016.

[16] Nguyen, H.B., An Overview on the NTRU Cryptographic System. 2015, San Diego State University.

[17] Hoffstein, J., J. Pipher, J.H. Silverman, and J.H. Silverman, An Introduction to Mathematical Cryptography. Vol. 1. 2008: Springer.

[18] Hoffstein, J., J. Pipher, J.M. Schanck, J.H. Silverman, W. Whyte, and Z. Zhang, Choosing Parameters for NTRUEncrypt, in Report 2015/708. 2015, IACR: Cryptology ePrint Archive.

[19] Stein, W., The Ring of Integers Modulo N, in Elementary Number Theory: Primes, Congruences, and Secrets: A Computational Approach. 2009, Springer New York: New York, NY. p. 1-27.

[20] Silverman, J.H., Almost Inverses and Fast NTRU Key Creation. NTRU Cryptosystems,(Technical Note#014):http://www.ntru.com/cryptolab/pdf/NTRU Tech014. pdf, 1999.

[21] Brand, K., NTRU: A Lattice-Based Cryptosystem and Attacks Against It, in Institute of Mathematics. 2013, University of Zurich: Switzerland.