URECA: Security Evaluation of NTRU Public Key Cryptosystem

Charissa Irene Utomo
Nanyang Technological University
School of Physical and Mathematical Sciences

Assoc Prof Wu Hongjun
Nanyang Technological University
School of Physical and Mathematical Sciences

***Abstract* -** NTRU, formally known as N-th degree Truncated polynomial Ring Units, is a lattice-based public-key cryptosystem. The security of NTRU is due to the difficulty of mathematical computational problems has yet to be solved optimally. In the report, we visit one of the computational problems, namely the shortest vector problem, as we explore numerous algorithms and heuristics in finding and optimizing the shortest vector for NTRU. Furthermore, we will give an overview of how NTRU works especially for its well-known lattice structure and visit some existing attacks as we evaluate its advantages and disadvantages and give suggestive improvements. The purpose of this research is to learn the characteristics of the NTRU cryptosystem and examines the security of NTRU while formulating a possible new and improved attack on NTRU with the imbued knowledge of co-existing attacks and known algorithms such a Lenstra–Lenstra–Lovász lattice basis reduction and Blockwise Korkine-Zolotarev algorithm, for short LLL and BKZ respectively with integrations of integer programming.

Keywords - NTRU, cryptosystem, security, LLL, BKZ

1 INTRODUCTION

We would be investigating the security of N-th degree Truncated polynomial Ring Units which is also known as NTRU for short. NTRU is an open-sourced public-key cryptosystem allowing two remote parties to communicate in a secure way without sharing a secret key.

NTRU is required to be secure against the future quantum computer to minimise information intersections and attacks.

In this project, we will analyze the security of NTRU and delve on the characteristics of lattice-based cryptosystem. Furthermore, we will study and improve the current existing attacks on NTRU and attempt to develop a new attack against NTRU. In the corresponding chapters, we will discuss mainly on one of the many computational problems of NTRU, namely shortest vector problem, SVP in short.

The expected deliverables for the project is an overall analysis report of the existing attacks calibrating its advantages and suggestive improvement and an attempt on implementation of a new attack against NTRU.

2 MATERIAL AND METHODS

2.1 KEY DEFINITIONS

2.1.1 Cryptography

Cryptography is originated from Ancient Greek where “crypto” stands for secret or hidden and “graphy” stands for writing. Therefore, cryptography is the art of secret writing.

2.1.2 Lattice

Lattice is mathematically defined as given that with n linearly independent vectors in a basis of the lattice, $(b_1, b_2, \ldots, b_m) \in \mathbb{R}^m$.

$$L(b_1, b_2, \ldots, b_n) = \{ \sum x_i b_i \mid x_i \in \mathbb{Z} \}$$

2.2 PRINCIPLES AND THEOREMS

2.2.1 Kerckhoffs’s principle

In modern cryptography, reduced focus in security on obscurity unlike the past is due to Kerckhoff’s principle. Kerckhoff’s principle states that the security should not be breached even if majority of the detail about a general cryptosystem such as encryption and decryption except secret key is publicly known.

2.3 ENCRYPTION AND DECRYPTION

2.3.1 Detailed Description of Encryption of NTRU

Before delving into detail on how to tackle NTRU computational problem, one needs to understand how encryption of NTRU works. Suppose Alice is the one who encrypts the message and Bob is the other member who decrypts the message. Alice selects a message m in polynomial form in range

of [-π/2,π/2] and polynomial r with minimal coefficients. Alice will use public key *h*, to compute the encrypted message, $e = r \cdot h + m \ (mod \ q)$ and send to Bob. Figure 1 below presents a pictorial representation of the encryption scheme of NTRU.

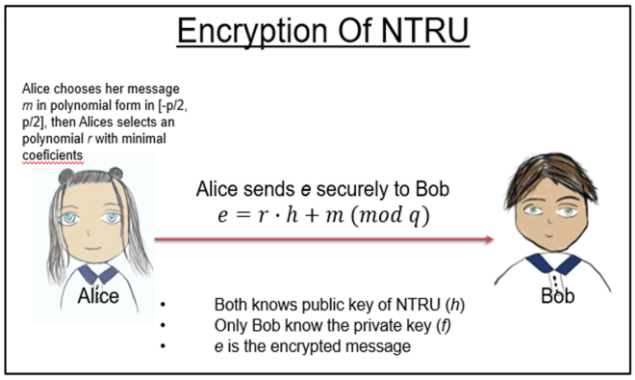

Figure 1: Alice and Bob Encryption Scheme for NTRU

2.3.2 Detailed Description of Decryption of NTRU

When Bob received encrypted message e, Bob can decrypt the message by first computing $a = f \cdot e \ (mod \ q)$, a integer coeficient polynomial, the recovered message cn be tabulated as such $c = f_p \cdot b = m \ (mod \ p)$. Figure 2 below illustrates the decryption scheme of NTRU.

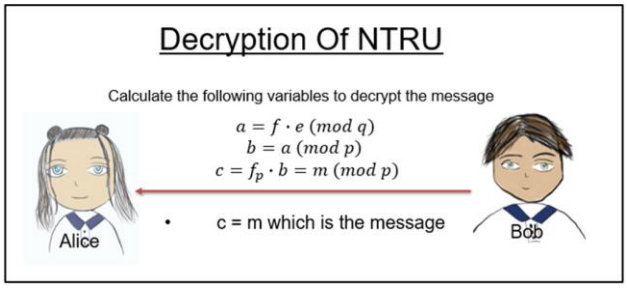

Figure 2: Alice and Bob Decryption Scheme for NTRU

2.4 BEHIND THE MATH OF NTRU

In this section, we evaluate the Mathematical aspect of NTRU particularly lattice-based cryptography and its characteristics. This is crucial in understanding the crux of how NTRU works and what can be the potential improvements be made.

2.4.1 Lattice-based Cryptography

With the definition of Lattices explained under 2.1.2, we can further delve on how it works with an applicable example. We will further discuss the application of lattices under 2.5 & 2.6 Encryption and Decryption.

2.4.2 Characteristics of NTRU

NTRU cryptosystem remain secure due to difficulty of lattice reduction as the following problem that will be discussed namely SVP, CVP, BDP and CRP.

2.4.3. Shortest Vector Problem

Shortest Vector Problem also known as SVP requires one to calculate a non-zero shortest vector in the lattice with the basis [1].

2.4.4 Closest Vector Problem

Closest Vector Problem, in short CVP, is to discover the closest vector in the lattice to a given vector, v not in the lattice [1].

2.4.5 Bounded Distance Decoding

BDD is Bounded Distance Decoding which is a specific case of CVP which needs to find the closest lattice point, s and that v is marginally close to s [1].

2.4.6. Covering Radius Problem

Covering Radius Problem, CRP, is given the basis of lattice, evaluate the sphere with the smallest volume s.t. for every lattice point, it involves 2 lattices points [1].

We acknowledge there are many computational problems of NTRU, however amongst all of them, this report will mainly focus on one which is the SVP of Lattice.

2.5 ALGORITHMS

The algorithm that we mentioned below would be useful in understanding how to tackle the computational problems of NTRU lattice problems.

2.5.1 Euclidean Algorithm

The Euclidean Algorithm determines the greatest common divisor between two given integers as follows $(\text{i.e } a,b \in \mathbb{Z}_+)$:

$$gcd(a,b) = \begin{cases} b & \text{if } a \equiv 0 \pmod{b} \\ gcd(a, a \pmod{b}) & \text{otherwise} \end{cases}$$

2.5.2 Gram Schmidt's Orthogonalisation Procedure

Gram Schmidt's Orthogonalisation Procedure outputs an orthonormal basis, $S = \{w_1, w_2, \ldots, w_n\}$,

when given a linearly independent set, $S' = \{v_1, v_2, \ldots, v_n\}$.The following will show how $S' = \{v_1, v_2, \ldots, v_n\}$ is set up and determined:

$$\begin{cases} v_1 = w_1 \\ v_k = w_k - \sum_{j=1}^{k-1} \frac{\langle w_k, v_j \rangle}{\|v_j\|^2} \text{ for } 2 \le k \le n \end{cases}$$

2.5.3 LLL Algorithm

LLL algorithm, is for short for Lenstra–Lenstra–Lovász lattice basis reduction algorithm, given a basis of the lattice, the algorithm would output an approximation of the shortest vector. The first step is to undergo Gram Schmidts Orthogonalisation refered in Section 2.3.2. Following with checking if the following conditions hold:

- For $1 \le j < i \le n$: $|\mu_{i,j}| \le 0.5$, to ensure the length is reduced
- Lovasz condition denoted as for $2 \le k \le n$, $\delta \|\mathbf{b}_{k-1}^*\|^2 \le \|\mathbf{b}_k^*\|^2 + \mu_{k,k-1}^2 \|\mathbf{b}_{k-1}^*\|^2$

The pseudocode can be refered below:

Algorithm 1: LLL Algorithm

Input: $\{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n\}$

Repeat two steps until find the LLL reduced basis

Step 1: Gram-Schmidt orthogonalization

for $i = 1$ **to** n **do**
  for $k = i - 1$ **to** 1 **do**
    $m \leftarrow$ nearest integer of $u_{k,i}$
    $\mathbf{b}_i \leftarrow \mathbf{b}_i - m\mathbf{b}_k$
  end
end

Step 2: Check Condition 2, and swap

for $i = 1$ **to** $n - 1$ **do**
  if $\|\mathbf{b}_{i+1}^* + u_{i,i+1}\mathbf{b}_i^*\|^2 < \frac{3}{4}\|\mathbf{b}_i^*\|^2$ **then**
    swap $\mathbf{b}_{i+1}$ and $\mathbf{b}_i$
    go to step 1
  end
end

Figure 3: Pseudocode of LLL Algorithm [2]

2.5.4 BKZ Algorithm

BKZ is a natural generalisation of LLL with block sizes ranges of $\beta \ge 2$. BKZ utilises SVP oracle of ranks of lesser than β as a subroutine while undergoing LLL algorithm to ensure bases are independent in order to compute (δ, β)-BKZ-reduced bases in higher dimension matrices. [3]

The following shows the pseudocode of BKZ algorithm:

Algorithm 1 BKZ: Schnorr-Euchner's BKZ algorithm [SE91]

Input: A blocksize $\beta \in (2, n)$, a relaxation factor $\delta \in (1, 2)$, and a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of a lattice L in $\mathbb{Z}^m$.
Output: A (δ, β)-BKZ-reduced basis of L.

$z \leftarrow 0$; $j \leftarrow 0$; $\frac{1}{2}$-LLL-reduce B
while z < n-1 **do**
  $j \leftarrow (j \bmod (n-1)) + 1$; $n_j \leftarrow \min\{j + \beta - 1, n\}$; $h \leftarrow \min\{j + \beta, n\}$
  Run an enumeration for $L(B_{[j,n_j]})$ to find $(\alpha_j, \ldots, \alpha_{n_j}) \in \mathbb{Z}^{n_j - j + 1}$ and compute $\mathbf{b} = \sum_{i=j}^{n_j} \alpha_i \mathbf{b}_i$ such that $\|\pi_j(\mathbf{b})\| = \lambda_1(L(B_{[j,n_j]}))$
  if $\|\mathbf{b}_j^*\|^2 > \delta \times \|\pi_j(\mathbf{b})\|^2$ **then**
    $z \leftarrow 0$; $\frac{1}{\delta}$-LLL-reduce $(\mathbf{b}_1, \ldots, \mathbf{b}_{j-1}, \mathbf{b}, \mathbf{b}_j, \ldots, \mathbf{b}_h)$ at stage j
  else
    $z \leftarrow z + 1$; 0.99-LLL-reduce $(\mathbf{b}_1, \ldots, \mathbf{b}_h)$ at stage $h - 1$
  end if //Due to LLL calls, $B_{[j,n_j]}$ may no longer be δ-SVP-reduced right after this step.
end while
return B. //It is folklore in practice to allow $\delta = 1$ and run (say,) 0.99-LLL-reductions at Steps 1 and 6.

Figure 4: Pseudocode of BKZ Algorithm [2]

2.3.4 Simplex Algorithm

Simplex algorithm is a method of solving integer programming models with slack variables, tableaus, and pivot variables to find an optimal solution. The objective is to either maximise or minimise $\mathbf{c}^T\mathbf{x}$ with subjected to constraints of $\mathbf{Ax} \le \mathbf{b}$ and $\mathbf{x} \ge \mathbf{0}$. Below shows the tableau shows in matrix form:

$$\begin{bmatrix} 1 & -\mathbf{c}^T & 0 \\ 0 & \mathbf{A} & \mathbf{b} \end{bmatrix}$$

2.3.4 Linear programming (Branch and Bound)

A method of linear programming is branch and bound, a heuristic approach searching information to finding optimal paths for linear programming, each path has a constraint and eliminates those that does not satisfy the conditions of the original problem.

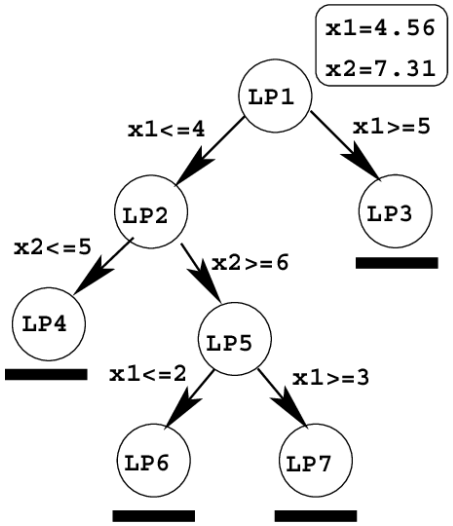

Figure 5: Branch and bound diagram [4]

3 CASE STUDIES

In this section, we will evaluate the pre-existing attacks used on NTRU, namely the LLL attack,

BKZ attack, BKZ 2.0, meet-in-the middle attack, and brute force attack to understand on what to focus and target on when attacking specifically NTRU system.

3.1 PRE-EXISTING ATTACK

3.1.1 LLL Attack

LLL basis reduction algorithm utilises Gram-Schmidt's Orthogonalisation to compute a set of orthogonal set of vectors. Furthermore, the algorithm checks the vectors under Lovász condition whether to swap the vectors. This algorithm tackles one of the root computational problems of NTRU, SVP explained under Section 2.4.3.

3.1.2 BKZ Attack

From section 2.5.4, the BKZ attack in NTRU utilises the BKZ algorithm to discover the shortest norm from the NTRU lattice.

3.1.3 BKZ 2.0 Attack

BKZ 2.0 integrates new refinements to BKZ, for instance, Gama-Nguyen-Regev pruning [8]. When blocksize is large, $\beta \geq 30$, BKZ 2.0 would an improved version of BKZ.

3.1.4 BRUTE-FORCE ATTACK

Brute-force is one of the most common attacks for cryptosystems. However, it usually can take a longer period to be able to fully attack the system. Due to large runtime complexity, it may take many permutations to be carried out before determining the actual plaintext.

3.1.5 MEET-IN-THE-MIDDLE ATTACK

Meet-in-the-middle attack, also known as MITM, is a form of cryptanalysis to observe the patterns from both ends until the middle part of the ciphertext. This attack can break the system in faster and more elegant manner as compared to brute-force as it reduces the number of permutations to discover the secret keys of NTRU.

3.2 LIMITATIONS

3.2.1 LIMITATIONS OF LLL

There is a trade-off between the time and the quality of the vector output as it might not provide the shortest vector in the shorter time limit[5].

3.2.2 LIMITATIONS OF BKZ

Similarly to LLL, there is a trade-off between the time and the quality of the vector output and may not provide the shortest vector in the shorter time limit [5]. There is also another limitation to assess the quality of BKZ output

3.2.3 LIMITATIONS OF BKZ 2.0

If blocksize is rather small, it might not reduce the time complexity that much to the LLL attack as we might consider most SVP oracles when going through the recursive step, so it might not be a faster alternative to LLL.

4 RESULTS AND FINDINGS

4.1 TRIALS

We performed LLL attack on a few lattices to evaluate and understand how it worked to form the shortest vector.

An example of a lattice produced from NTRU is

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 50 & 5 & 32 & 36 & 31 & 53 & 28 & 46 & 25 & 49 & 11 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 & 50 & 5 & 32 & 36 & 31 & 53 & 28 & 46 & 25 & 49 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 49 & 11 & 50 & 5 & 32 & 36 & 31 & 53 & 28 & 46 & 25 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 25 & 49 & 11 & 50 & 5 & 32 & 36 & 31 & 53 & 28 & 46 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 46 & 25 & 49 & 11 & 50 & 5 & 32 & 36 & 31 & 53 & 28 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 28 & 46 & 25 & 49 & 11 & 50 & 5 & 32 & 36 & 31 & 53 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 53 & 28 & 46 & 25 & 49 & 11 & 50 & 5 & 32 & 36 & 31 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 31 & 53 & 28 & 46 & 25 & 49 & 11 & 50 & 5 & 32 & 36 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 36 & 31 & 53 & 28 & 46 & 25 & 49 & 11 & 50 & 5 & 32 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 32 & 36 & 31 & 53 & 28 & 46 & 25 & 49 & 11 & 50 & 5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61
\end{bmatrix}$$

The results give a short norm of 327.17 with basis [0, -1,-2,-2, -1, 2, 2, -1, 1, 1, 1, -62, -62, -125, 61,123, 0, 0, -184, 124,125,0].

4.2 LATTICE-BASED PROPOSED ATTACK

With the application of linear programming and LLL, we aim to tackle the shortest vector problem. Figure 6 is a visualisation of how the branch and bound would work in the proposed attack.

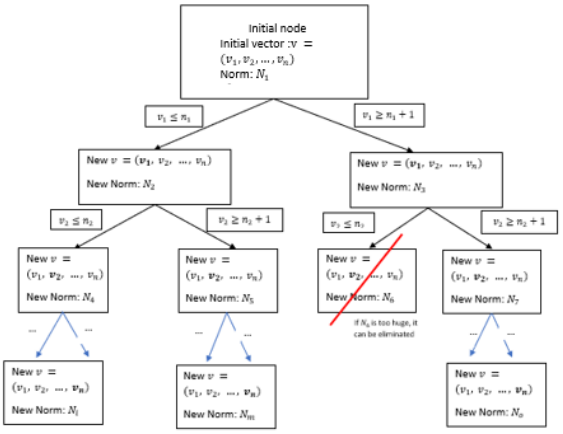

Figure 6: Tree diagram of Branch and Bound on NTRU

There is a split in branches in each step, comparison of the norms to observe which branch to traverse.

This method is a type of brute force but reduce and eliminate some unnecessary iterations that cause the norm to inflate that would not satisfy the shortest vector problem.

After iterating for every index of the vector, the branch and bound algorithm would output the shortest vector problem.

5 CONCLUSION

In a nutshell, we have gain knowledge on the characteristics of lattice-based cryptography specifically NTRU. Moreover, we examined its attacks and flaws to understand the security of NTRU. Conversely, I strongly believe that overall NTRU remains as secure and can be used universally. This concludes our project (readily available at https://github.com/charutomo/Security-Evaluation-of-NTRU-Public-Key-Cryptosystem) and feel free to contact us via email for further enquires. Thank you for your upmost support.

ACKNOWLEDGMENT

I would like to acknowledge the funding support from Nanyang Technological University – URECA Undergraduate Research Programme for this research project.

REFERENCES

[1] The LLL Algorithm. Survey and Applications. Editors: Nguyen P., Vallée B. 2022., Springer Berlin, Heidelberg

[2] Deng, X., n.d. An Introduction to Lenstra-Lenstra-Lovasz Lattice Basis Reduction Algorithm. Retrieved June 6, 2022, from https://math.mit.edu/~apost/ courses/18.204-2016/18.204_Xinyue_Deng _final_paper.pdf

[3] Li, J., & Nguyen, P. Q. 2020. A Complete Analysis of the BKZ Lattice Reduction Algorithm.from https://eprint.iacr.org/2020 /1237

[4] Fig. 1. A sketch of the working principle of the branch-and-bound... (n.d.). ResearchGate. Retrieved June 6, 2022, from https://www.researchgate.net/figure/A-sketch-of-the-working-principle-of-the-branch-and-bound-method-for-a-twovariable_fig1_220742933

[5] Laarhoven T. 2015. Sieving for Shortest Vectors in Lattices Using Angular Locality-Sensitive Hashing, Advances in Cryptology—CRYPTO 2015.

NONMENCLATURE

BDD	Bounded Distance Decoding
BKZ	Blockwise Korkine-Zolotarev (BKZ) Algorithm
CRP	Covering Radius Problem
CVP	Closest Vector Problem
LLL	Lenstra–Lenstra–Lovász lattice basis reduction algorithm
MITM	Meet-in-the-middle attack
NTRU	N-th degree Truncated polynomial Ring Units
s.t.	Such That
SPMS	School of Physical and Mathematical Sciences
SVP	Shortest Vector Problem
URECA	Undergraduate Research Experience on CAmpus