Polynomial equation in algebraic attack on NTRU-HPS and NTRU-HRSS

Fadila Paradise and *Kiki Ariyanti* Sugeng*

Department of Mathematics, Faculty of Mathematics and Natural Sciences (FMIPA), Universitas Indonesia, Depok 16424, Indonesia

Abstract. NTRU is a lattice-based public-key cryptosystem designed by Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman in 1996. NTRU published on Algorithmic Number Theory Symposium (ANTS) in 1998. The ANTS'98 NTRU became the IEEE standard for public key cryptographic techniques based on hard problems over lattices in 2008. NTRU was later redeveloped by NTRU Inc. since 2018 and became one of the finalists in round 3 of the PQC (Post-Quantum Cryptography) standardization process organized by NIST in 2020. There are two types of NTRU algorithms proposed by NTRU Inc., which are classified based on parameter determination, NTRU-HPS (Hoffstein, Pipher, Silverman) and NTRU-HRSS (Hulsing, Rijnveld, Schanck, Schwabe). Algebraic attacks on ANTS'98 NTRU had previously been carried out in 2009 and 2012. In this paper, algebraic attack is performed on the renewed NTRU submitted to the third round of NIST PQC standardization in 2020. This study aims to obtain system of polynomial equations representing plaintext bits of NTRU-HPS and NTRU-HRSS which can be used to find the private key in the next study. As a result, some polynomial equations with unknown variables were obtained.

Keywords. Algebraic attack, NTRU-HPS, NTRU-HRSS

1 Introduction

Concept of quantum computing has changed many scientific fields, one of which is cryptography. Quantum computers can perform some code breaking methods much faster than classical computers [1]. For example, Shor's Algorithm is a quantum computer algorithm for finding the prime factors of an integer. The existence of Shor's Algorithm makes the 'large integer factorization problem' in RSA can be solved [2]. Therefore, it is necessary to develop cryptosystems to provide a replacement for classical cryptosystems that are vulnerable to quantum computer-based attacks. This replacement cryptosystem is called post quantum cryptography [3].

In order to develop a standard for post quantum cryptography, National Institute of Standards and Technology (NIST) is selecting one or more public key cryptography algorithms through a public, competition-like process. The NIST Post-Quantum

* Corresponding author: kiki@sci.ui.ac.id

Cryptography (PQC) Standardization Process began in 2017 with a total of 69 candidates. In July 2020, NIST published candidates who were finalists in round 3, one of which was NTRU [4]. NTRU is a lattice-based public key cryptosystem that provides digital signature and encryption algorithm solutions [3]. NTRU published on Algorithmic Number Theory Symposium (ANTS) in 1998 and became the IEEE standard for public key cryptographic techniques based on hard problems over lattices in 2008 [5]. The ANTS'98 NTRU was later redeveloped by NTRU Inc. since 2018 and updated for several times during the PQC standardization process.

Currently, a lot of research is occurring on the implementation of NTRU whether on networks, hardware, or Internet of Things (IoT) [5]. Many security tests were also carried out on ANTS'98 NTRU, such as algebraic attack using Witt vectors and Grobner bases carried out by Bourgeois and Faugere in 2009 [6], algebraic attack via solving equations over real numbers by Ding and Schmidt in 2012 [7], also lattice attack experiment conducted by Bi and Han in 2021 [8]. In this paper, algebraic attack is performed on the renewed NTRU algorithm submitted to the third round of NIST PQC Standardization until polynomial equations generation process. This process aims to obtain system of polynomial equations representing plaintext bits of NTRU-HPS and NTRU-HRSS which can be used to find the private key in the next study. The types of NTRU algorithms that chosen as the object of algebraic attack are ntruhps2048509 ($q = 2048, n = 509$) and ntruhrss701 ($n = 701$). This paper also shows the stages in performing algebraic attacks on NTRU and the polynomial equations obtained.

2 Theoretical framework

2.1 Algebraic attack

Algebraic attack was first introduced by Kipnis and Shamir (1999) in a paper entitled "Cryptanalysis of The HFE Public Key Cryptosystem by Relinearization" [9]. Kipnis and Shamir use an algorithm called "Relinearization" to solve a system of nonlinear equations on a finite field [9]. The main principle of an algebraic attack is very simple, which is to change the problem of attacking a cryptographic system (such as finding a secret key) into solving a system of polynomial equations. Basically, there are two stages in performing algebraic attack, the first is finding a system of equations, the second is finding solution of the system of equations [6]. There are several methods commonly used to find solutions to polynomial equations, including Grobner's Basis, F4, F5, and XL algorithm [10]. The complexity of solving polynomial equations grows exponentially according to the polynomial formed, therefore cryptanalysis is needed by considering the use of small degree polynomial equations [11].

2.2 NTRU

NTRU is a lattice-based public key cryptosystem designed by Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman in 1996 [11]. NTRU became one of the finalists in round 3 of the NIST standardization process in 2020. In general, there are two types of algorithms proposed by NTRU in the round 3 standardization selection process if classified based on the determination of the parameters, namely NTRU-HPS (Hoffstein, Pipher, Silverman) and NTRU-HRSS (Hulsing, Rijnveld, Schanck, Schwabe). Specifically, there are four types of both NTRU-HPS and NTRU-HRSS submitted to NIST PQC Standardization that is named by length of n and q, ntruhps2048509, ntruhps2048677, ntruhps4096821, and

ntruhrss701 [12]. This study chooses ntruhps2048509 and ntruhrss701 to become the object of algebraic attack.

In general, NTRU has several parameters, constants, and functions. To perform algebraic attack on NTRU, it is important to know first the parameters in NTRU. Below are parameter sets in NTRU that used in performing algebraic attack on NTRU [12]:

1. $(\mathbb{Z}/n)^{\mathrm{x}}$ is the multiplicative group of integers modulo n.
2. Φ_1 is the polynomial $(x-1)$.
3. Φ_n is the polynomial $(x^n-1)/(x-1) = x^{n-1} + x^{n-2} \ldots + 1$.
4. R is the quotient ring $\mathbb{Z}[x]/\Phi_1\Phi_n$.
5. S is the quotient ring $\mathbb{Z}[x]/\Phi_n$
6. $S/3$ is the quotient ring $\mathbb{Z}[x]/(3, \Phi_n)$
7. R/q is the quotient ring $\mathbb{Z}[x]/(q, \Phi_1\Phi_n)$.
8. $\underline{Rq}(a)$ is canonical R/q representative of polynomial a with coefficients in $\{-\frac{q}{2}, -\frac{q}{2}+1, \ldots, \frac{q}{2}-1\}$.
9. $Rq(a)$ is non-normative R/q representative.
10. S/q is the quotient ring $\mathbb{Z}[x]/(q, \Phi_n)$.
11. $\underline{Sq}(a)$ is canonical S/q representative of polynomial a with coefficients in $\{-\frac{q}{2}, -\frac{q}{2}+1, \ldots, \frac{q}{2}-1\}$.
12. $Sq(a)$ is non-normative S/q representative.
13. $\underline{S3}(a)$ is canonical $S/3$ representative of polynomial a with coefficients in $\{-1,0,1\}$.
14. $S3(a)$ is non-normative $S/3$ representative.
15. A polynomial is ternary if its coefficients are in $\{-1,0,1\}$.
16. T is the set of non-zero ternary polynomials of degree at most $n-2$.
17. $T(d)$, for an even positive integer d, $T(d)$ consisting of ternary polynomials that have exactly $d/2$ coefficients equal to 1 and $d/2$ coefficients equal to -1.

Functions in NTRU are needed to encode and identify keys and ciphertext in performing algebraic attack on NTRU. NTRU functions can be seen in Table 1.

Table 1. NTRU functions

Encoding ***(Functions used in encoding polynomial to binary and vice versa)***
• bit strings and byte arrays *(encode bit to bytes and vice versa)*
• pack_Rq0 *(encode polynomial $a \equiv 0\ mod\ (q, \Phi_1\Phi_n)$ to binary)*
• unpack_Rq0 *(encode binary to polynomial $a \equiv 0\ mod\ (q, \Phi_1\Phi_n)$)*
• pack_Sq *(encode polynomial $Sq(a)$ to binary)*
• unpack_Sq *(encode binary to polynomial $Sq(a)$)*
• pack_S3 *(encode polynomial $S3(a)$ to binary)*
• unpack_S3 *(encode binary to polynomial $S3(a)$)*
Deterministic Public Key Encryption (DPKE)
• DPKE_Key_Pair *(generate private key and public key pairs, consist of DPKE_Public_Key function)*
• DPKE_Public_Key *(compute public key using input from DPKE_Key_Pair)*
• DPKE_Encrypt *(encrypt plaintext)*
• DPKE_Decrypt *(decrypt ciphertext)*
Key Encapsulation Mechanism (KEM)
• Key_Pair *(encapsulate private key and public key pairs)*
• Encapsulate *(encapsulate ciphertext)*
• Decapsulate *(decapsulate ciphertext)*

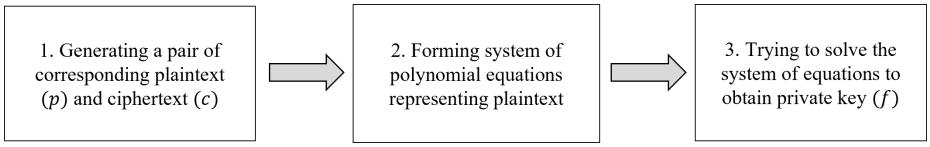

Fig. 1. Attack scenario.

In this study, one sample for each ntruhps2048509 and ntruhrss701 are taken randomly from NTRU submission documentation. Samples are packed in strings of hexadecimal. Samples need to be decapsulated using KEM Decapsulate to separate the keys and ciphertext. After being decapsulated, the keys and ciphertext strings need to be encoded from hexadecimal to polynomial using encoding functions. The ciphertext in polynomial representative is decrypted using private key through DPKE_Decrypt so that the plaintext can be obtained. DPKE_Decrypt becomes the main function that used to generate system of polynomial equations in occurring algebraic attack on NTRU.

Polynomials in NTRU are multiplied by the cyclic convolution product. Let $H = F \cdot G$. Then for each $0 \leq k < N$, the k^{th}-coefficient H_k of H is given by,

$$H_k = \sum_{i=0}^{k} F_i G_{k-i} + \sum_{i=k+1}^{n-1} F_i G_{n+k-i} = \sum_{i+j \equiv k (\text{mod } n)} F_i G_j. \quad (1)$$

3 Algebraic attack on NTRU

3.1 Attack scenario

In real implementation, public key is shared with the public whether private key is known only to the owner. Attacker can generate a pair of corresponding plaintext and ciphertext using public key and perform an algebraic attack scenario with the following process.

As previously explained, plaintext and ciphertext from chosen sample in NTRU submission documentation have been obtained through several functions. In this study, the plaintext and ciphertext from sample are considered as a pair of corresponding plaintext and ciphertext generated by attacker in the stage 1 of Fig. 1.

The attack scenario carried out in this paper is up to stage 2, forming system of polynomial equations representing plaintext. In forming system of polynomial equations, attacker perform decryption through decryption function (DPKE_Decrypt) using unknown variable of private key.

3.2 Modification of Decryption Function

The encryption and decryption functions in NTRU use two modulus operations. Decryption function of NTRU in simple form is as follows,

$$Sq(m) = \left(c \cdot f \ mod \ (q, \Phi_1 \Phi_n)\right) \cdot f_p \ mod \ (p, \Phi_n) \quad (2)$$

with $c \ (mod \ q) \neq c \ (mod \ p)$. The use of two modulus operations make it difficult for attackers because as long as c is operated with unknown f and f_p, the operation must still be

in the modulus q and cause the coefficients on the polynomial formed to be in the range $\{-q/2, \dots, q/2\}$.

On the other hand, there is an opportunity that NTRU-HPS and NTRU-HRSS can be attacked using algebraic attack. It can be seen that the polynomial f and f_p on NTRU-HPS and NTRU-HRSS are in $S/3$ or quotient ring $\mathbb{Z}[x]/(3, \Phi_n)$ and can be operated with c, so that the decryption function can be changed into,

$$Sq(m) = \Big(\big(c \cdot f \ mod\ (q, \Phi_1\Phi_n)\big) \cdot f_p \ mod\ (p, \Phi_1\Phi_n)\Big) mod\ (p, \Phi_n). \tag{3}$$

With unknown value of f dan f_p, operation $c.f.f_p$ must be in $(mod\ q)$ until value of f and f_p are found, so decryption function is turned into this form to produce polynomial equations that represent plaintext in NTRU-HPS and NTRU-HRSS,

$$Sq(m) = \Big(\mathrm{c} \cdot f \cdot f_p \ mod\ (q, \Phi_1\Phi_n)\Big) \ mod\ (p, \Phi_n).$$

3.3 Polynomial equations generation

Polynomials in NTRU are multiplied by the cyclic convolution product which shown in Equation (1). As example, by using cyclic convolution product, multiplication of Equation (2) in ntruhps2048509 produce,

$$v_0 = c_0f_0 + c_1f_{508} + c_2f_{507} \dots + c_{507}f_2 + c_{508}f_1 \tag{4}$$

$$v_1 = c_0f_1 + c_1f_0 + c_2f_{508} \dots + c_{507}f_3 + c_{508}f_2 \tag{5}$$

$$\vdots$$

$$v_{508} = c_0f_{508} + c_1f_{507} + c_2f_{506} \dots + c_{507}f_1 + c_{508}f_0. \tag{6}$$

After that the multiplication in Equation (3) produce,

$$m_0 = v_0f_{p_0} + v_1f_{p_{508}} + v_2f_{p_{507}} \dots + v_{507}f_{p_2} + v_{508}f_{p_1} \tag{7}$$

which can be expand into,

$$\begin{aligned} m_0 = {} & (c_0f_0 + c_1f_{508} + c_2f_{507} \dots + c_{507}f_2 + c_{508}f_1)f_{p_0} + \\ & (c_0f_1 + c_1f_0 + c_2f_{508} \dots + c_{507}f_3 + c_{508}f_2)f_{p_{508}} + \dots + \\ & (c_0f_{508} + c_1f_{507} + c_2f_{506} \dots + c_{507}f_1 + c_{508}f_0)f_{p_1} \end{aligned} \tag{8}$$

and result,

$$m_0 = \left(c_0 f_0 f_{p_0} + c_1 f_{508} f_{p_0} + c_2 f_{507} f_{p_0} \ldots + c_{507} f_2 f_{p_0} + c_{508} f_1 f_{p_0}\right) + \left(c_0 f_1 f_{p_{508}} + c_1 f_0 f_{p_{508}} + c_2 f_{508} f_{p_{508}} \ldots + c_{507} f_3 f_{p_{508}} + c_{508} f_2 f_{p_{508}}\right) + \cdots + \left(c_0 f_{508} f_{p_1} + c_1 f_{507} f_{p_1} + c_2 f_{506} f_{p_1} + \cdots + c_{507} f_1 f_{p_1} + c_{508} f_0 f_{p_1}\right). \quad (9)$$

Variables in the equations then sort into,

$$m_0 = \left(c_0 f_0 f_{p_0} + c_{508} f_1 f_{p_0} + c_{507} f_2 f_{p_0} + \cdots + c_2 f_{507} f_{p_0} + c_1 f_{508} f_{p_0}\right) + \left(c_{508} f_0 f_{p_1} + c_{507} f_1 f_{p_1} + \cdots + c_2 f_{506} f_{p_1} + c_1 f_{507} f_{p_1} + c_0 f_{508} f_{p_1}\right) + \cdots + \left(c_1 f_0 f_{p_{508}} + c_0 f_1 f_{p_{508}} + c_{508} f_2 f_{p_{508}} + c_{507} f_3 f_{p_{508}} + \cdots + c_2 f_{508} f_{p_{508}}\right) \quad (10)$$

By using several steps shown in Equation (4) until Equation (10), $m_0, m_1, \ldots m_{n-1}$ in ntruhps2048509 and ntruhrss701 are obtained. Below is the system of polynomial equations that represent plaintext bits of ntruhps2048509.

1. $1209 f_0 f_{pd_0} + 713 f_1 f_{p_0} + 269 f_2 f_{p_0} + 1865 f_3 f_{p_0} + 1907 f_4 f_{p_0} + \cdots + 1154 f_{507} f_{p_{508}} + 174 f_{508} f_{p_{508}} = 0$
2. $230 f_0 f_{p_0} + 1209 f_1 f_{p_0} + 713 f_2 f_{p_0} + 269 f_3 f_{p_0} + 1865 f_4 f_{p_0} + \cdots + 838 f_{507} f_{p_{508}} + 1154 f_{508} f_{p_{508}} = 1$
3. $174 f_0 f_{p_0} + 230 f_1 f_{p_0} + 1209 f_2 f_{p_0} + 713 f_3 f_{p_0} + 269 f_4 f_{p_0} + \cdots + 1848 f_{507} f_{p_{508}} + 838 f_{508} f_{p_{508}} = -1$
4. $1154 f_0 f_{p_0} + 174 f_1 f_{p_0} + 230 f_2 f_{p_0} + 1209 f_3 f_{p_0} + 713 f_4 f_{p_0} + \cdots + 560 f_{507} f_{p_{508}} + 1848 f_{508} f_{p_{508}} = -1$
5. $838 f_0 f_{p_0} + 1154 f_1 f_{p_0} + 174 f_2 f_{p_0} + 230 f_3 f_{p_0} + 1209 f_4 f_{p_0} + \cdots + 1110 f_{507} f_{p_{508}} + 560 f_{508} f_{p_{508}} = 0$
6. $1848 f_0 f_{p_0} + 838 f_1 f_{p_0} + 1154 f_2 f_{p_0} + 174 f_3 f_{p_0} + 230 f_4 f_{p_0} + \cdots + 741 f_{507} f_{p_{508}} + 1110 f_{508} f_{p_{508}} = 0$
7. $560 f_0 f_{p_0} + 1848 f_1 f_{p_0} + 838 f_2 f_{p_0} + 1154 f_3 f_{p_0} + 174 f_4 f_{p_0} + \cdots + 1839 f_{507} f_{p_{508}} + 741 f_{508} f_{p_{508}} = 0$
8. $1110 f_0 f_{p_0} + 560 f_1 f_{p_0} + 1848 f_2 f_{p_0} + 838 f_3 f_{p_0} + 1154 f_4 f_{p_0} + \cdots + 567 f_{507} f_{p_{508}} + 1839 f_{508} f_{p_{508}} = -1$

   ⋮

508. $269 f_0 f_{p_0} + 1865 f_1 f_{p_0} + 1907 f_2 f_{p_0} + 382 f_3 f_{p_0} + 1123 f_4 f_{p_0} + \cdots + 230 f_{507} f_{p_{508}} + 1209 f_{508} f_{p_{508}} = -1$
509. $713 f_0 f_{p_0} + 269 f_1 f_{p_0} + 1865 f_2 f_{p_0} + 1907 f_3 f_{p_0} + 382 f_4 f_{p_0} + \cdots + 174 f_{507} f_{p_{508}} + 230 f_{508} f_{p_{508}} = 0$

Below is the system of polynomial equations that represent plaintext bits of ntruhrss701.

1. $4687 f_0 f_{p_0} - 3502 f_1 f_{p_0} + 3349 f_2 f_{p_0} + 1843 f_3 f_{p_0} + 6515 f_4 f_{p_0} + \ldots + 5049 f_{700} f_{p_{700}} = -1$
2. $3301 f_0 f_{p_0} + 4687 f_1 f_{p_0} - 3502 f_2 f_{p_0} + 3349 f_3 f_{p_0} + 1843 f_4 f_{p_0} + \ldots + 2283 f_{700} f_{p_{700}} = -1$
3. $5049 f_0 f_{p_0} + 3301 f_1 f_{p_0} + 4687 f_2 f_{p_0} - 3502 f_3 f_{p_0} + 3349 f_4 f_{p_0} + \ldots + 1091 f_{700} f_{p_{700}} = 1$

4. $2283f_0f_{p_0} + 5049f_1f_{p_0} + 3301f_2f_{p_0} + 4687f_3f_{p_0} - 3502f_4f_{p_0} + \ldots + 7376f_{700}f_{p_{700}} = 1$
5. $1091f_0f_{p_0} + 2283f_1f_{p_0} + 5049f_2f_{p_0} + 3301f_3f_{p_0} + 4687f_4f_{p_0} + \ldots + 262f_{700}f_{p_{700}} = 0$
6. $7376f_0f_{p_0} + 1091f_1f_{p_0} + 2283f_2f_{p_0} + 5049f_3f_{p_0} + 3301f_4f_{p_0} + \ldots + 3835f_{700}f_{p_{700}} = 1$
7. $262f_0f_{p_0} + 7376f_1f_{p_0} + 1091f_2f_{p_0} + 2283f_3f_{p_0} + 5049f_4f_{p_0} + \ldots + 7984f_{700}f_{p_{700}} = 1$
8. $3835f_0f_{p_0} + 262f_1f_{p_0} + 7376f_2f_{p_0} + 1091f_3f_{p_0} + 2283f_4f_{p_0} + \ldots + 1178f_{700}f_{p_{700}} = -1$

   ⋮

700. $3349f_0f_{p_0} + 1843f_1f_{p_0} + 6515f_2f_{p_0} + 4838f_3f_{p_0} + 5250f_4f_{p_0} + \ldots + 4687f_{700}f_{p_{700}} = 0$
701. $-3502f_0f_{p_0} + 3349f_1f_{p_0} + 1843f_2f_{p_0} + 6515f_3f_{p_0} + 4838f_4f_{p_0} + \ldots + 3301f_{700}f_{p_{700}} = 0$

The complete system of polynomial equations can be seen in the appendix. The number of polynomial equations formed is equal to n for each algorithm. Every polynomial equation formed contain n^2 nonzero coefficients. If these systems of equations are converted into matrices, the matrices will have a very large size of $n \times n^2$.

4 Conclusion

Two modulus operations on the encryption and decryption function become the strength of NTRU algorithm. In NTRU submission documentation, it is recommended to use $p = 3$ for all types of NTRU algorithm, $q = 2048$ for ntruhps2048509, and $q = 8192$ for ntruhrss701. The use of modulus $p = 3$ produces plaintext and ciphertext with small value so the calculation in encrypting and decrypting message can be light, while the use of large q makes the process in encryption and decryption function be complex so these processes are difficult to be attacked. Unfortunately, in performing algebraic attack on NTRU, the calculation must be in modulus q and can only be modulus p after the values of f and f_p are found. Polynomial c used in generating polynomial must in normative representation. This causes the resulting polynomial equations have maximum coefficient of $q - 1$.

In this study, 509 polynomial equations representing ntruhps2048509 plaintext bits and 701 polynomial equations representing ntruhrss701 plaintext bits were obtained as the result. Every polynomial equation formed contain n^2 non-zero coefficients. If these systems of equations are converted into matrices, the matrices will have a very large size of $n \times n^2$ which means 509×509^2 for ntruhps2048508 and 701×701^2 for ntruhrss701. For further research, it is necessary to find a suitable method that can reduce the size of the system of polynomial equations to make it easier to find solutions.

Acknowledgements

This research is funded by Dikti Research Grant No NKB-988/UN2.RST/HKP.05.00/2022.

References

1. M. Haart and C. Hoffs, "Quantum computing: What it is, how we got here, and who's working on it", (2019) available at https://www.researchgate.net/publication/331844245_Quantum_Computing_What_it_is_how_we_got_here_and_who%27s_working_on_it.
2. K. Li, P. Yan, and Q. Cai, Fundam. Res. **1**, 85–87 (2021).
3. K. Roy, Int. J. Appl. Eng. Res. **14**, 2608–2615 (2019).
4. G. Alagic et al., *Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process* (NIST.IR.8309, Gaithersburg, 2020).
5. IEEE, *IEEE Standard Specification for Public Key Cryptographic Techniques Based on Hard Problems over Lattices*, IEEE Std 1363.1-2008. 1–81. (2009).
6. G. Bourgeois and J. Faugère, J. Math. Cryptol. **3**, 205–214 (2009).
7. J. Ding and D. Schmidt, Cryptol. ePrint Arch. (2012).
8. J. Bi and L. Han, IEEE Access **9**, 66218–66222 (2021).
9. A. Kipnis and Shamir, *Advances in Cryptology-CRYPTO'99* (Springer, 1999), pp. 19–30.
10. A. A. Hafez, R. A. Elbarkouky and W. Hafez, Int. J. Adv. Res. Sci. Eng. Technol. **3**, 85–90 (2016).
11. D. Micciancio and O. Regev, Lattice-based cryptography, in *Post-quantum Cryptography* (Springer, Berlin, 2009), pp. 147–191.
12. C. Chen et al., *NTRU: Algorithm Specifications and Supporting Documentation* (NIST, 2019).