# MESSAGE RECOVERY ATTACK TO NTRU USING A LATTICE INDEPENDENT FROM THE PUBLIC KEY

## MARIOS ADAMOUDIS AND K. A. DRAZIOTIS

ABSTRACT. In the present paper we introduce a new attack on NTRU-HPS cryptosystem using lattice theory and Babai's Nearest Plane Algorithm. This attack generalizes the classic CVP attack on NTRU. Finally, the attack is illustrated by many examples.

#### 1. Introduction

NTRU¹ cryptosystem was proposed in 1996 by the three mathematicians, Pipher, Hoffstein and Silverman, [21]. It is based on certain hard problems involving lattices and can be used as an encryption system, NTRUencrypt, as well as a digital signature, like the RSA system. Notably, NTRU seems immune to quantum attacks, whereas RSA/Diffie-Hellman are vulnerable to Shor's quantum attack [29]. It was among the seven finalists of NIST's competition for Public-Key Post-Quantum Cryptographic Algorithms², but NIST will not standardize it. However, this system remains a great choice, for instance it is implemented in openssh 9.0³.

For preliminaries in NTRU, see [21, 23, 32]. In the present work we shall study the original NTRU system, namely NTRU-HPS. There are also two other flavors of the original NTRU: NTRU-Prime and NTRU-HRSS [19]. The former avoids decryption failures and rings associated to cyclotomics. Furthermore, NTRU-Prime has two sub variants: the streamlined NTRU-Prime, which is similar to NTRU-HPS, and NTRU-LPRime [5, 6], but it is based on the non-cyclotomic NTRU problem. All the three flavors passed to the second round of NIST's competition. In the third round NTRU-HPS and NTRU-HRSS (which have been merged) is one of the seven finalists<sup>4</sup>, and NTRU-prime is an alternate candidate for round 3. For differences between the three flavors see [28].

Before we present our contribution in section 2, we provide some necessary definitions for NTRU cryptosystem and some classic attacks on it.

### 1.1. NTRU cryptosystem.

<sup>2010</sup> Mathematics Subject Classification. 94A60.

Key words and phrases. Public Key Cryptography; NTRU Cryptosystem; Lattices; LLL algorithm; Closest Vector Problem; Babai's Nearest Plane Algorithm.

<sup>&</sup>lt;sup>1</sup>N-th degree Truncated polynomial Ring Units

<sup>&</sup>lt;sup>2</sup>https://csrc.nist.gov/news/2020/pqc-third-round-candidate-announcement

 $<sup>^3</sup>$ https://www.openssh.com/txt/release-9.0. It is implemented in a hybrid scheme, Streamlined NTRU Prime + x25519 ECDH ey exchange method.

<sup>4</sup>https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions

#### 1.1.1. Convolution Rings.

**Definition 1.1.** Let N be a positive integer. We call the ring

$$R = \mathbb{Z}[x]/\langle x^N - 1\rangle$$

the ring of convolution polynomials. If we consider the ring  $\mod q$  i.e.

$$R_q = \mathbb{Z}_q[x]/\langle x^N - 1 \rangle$$

we call it the ring of convolution polynomials  $\mod q$ .

We set  $\mathbb{F}[x]$  to be either R or  $R_q$ . The degree of a polynomial  $\mathbf{a}(x)$  in  $\mathbb{F}[x]$  is < N. If there is a monomial of order  $\geq N$ , say  $x^j$ , then  $x^j = x^{Ni+\ell} = x^\ell$  inside the ring  $\mathbb{F}[x]$ . Also, instead of  $\mathbf{a}(x)$  we can consider the vector  $\mathbf{a}$  with coordinates the coefficients of  $\mathbf{a}(x)$ . Let

$$\mathbf{a}(x) = a_0 + a_1 x + \dots + a_{N-1} x^{N-1}$$

and

$$\mathbf{b}(x) = b_0 + b_1 x + \dots + b_{N-1} x^{N-1}.$$

We define the star multiplication  $\star$  in R as follows,

$$\mathbf{a}(x) \star \mathbf{b}(x) = \mathbf{c}(x) = c_0 + c_1 x + \dots + c_{N-1} x^{N-1},$$

where

$$c_k = \sum_{i+j \equiv k \pmod{N}} a_i b_{k-j}, \ 0 \le k \le N-1,$$

and  $0 \le i, j \le N - 1$ . If  $\mathbf{a}(x)$  and  $\mathbf{b}(x)$  are in  $R_q$ , then we compute the star product using the previous formula and then, we reduce  $\mod q$ .

1.1.2. Circulant matrices and star multiplication. Let  $\mathbf{h}(x) = h_0 + h_1 x + \cdots + h_{N-1} x^{N-1}$  be a polynomial in R. We represent polynomials of R as integer vectors. For instance we represent  $\mathbf{h}(x)$  as the vector

$$\mathbf{h} = (h_0, ..., h_{N-1}).$$

Consider the circulant matrix

(1.1) 
$$C(\mathbf{h}) = \begin{bmatrix} h_0 & h_1 & \dots & h_{N-1} \\ h_{N-1} & h_0 & \dots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \dots & h_0 \end{bmatrix}.$$

Then, we can easily see that, if  $\mathbf{a}(x) \in R$ , then we get

$$\mathbf{a} \star \mathbf{h} = [\mathbf{a}]C(\mathbf{h}),$$

where [a] is the row-matrix with entries the coordinates of a.

1.1.3. NTRU-HPS cryptosystem. Alice chooses public parameters (N, p, q, d) with N and p primes,  $\gcd(N,q)=\gcd(p,q)=1$ . To get an idea of the size and form of the parameters, say p is small (usually 3) and N,q are large (of the same order) and q is a power of 2. The degree parameter N is chosen prime due to the attack of Gentry [18] when N is composite. What is more, a prime integer N maximizes the probability an element of  $R_q$  be invertible in the case where q is a prime power. Also, we define the set of ternary polynomials  $\mathcal{T}(d_1,d_2)\subset R$ , be the polynomials of R with  $d_1$  entries equal to one,  $d_2$  entries equal to minus one, and the remaining entries are zero. When we write  $\mathcal{L}_F$  for some polynomial  $F \in R$ , we mean a subset of  $\mathcal{T}(d_1,d_2)$  for some  $d_1,d_2$ , relatively small with respect to q. Alice chooses her private key  $(\mathbf{f}(x),\mathbf{g}(x))$ , such that,  $\mathbf{f}(x)\in\mathcal{L}_f$  and  $\mathbf{g}(x)\in\mathcal{L}_g^{-5}$ , where  $\mathbf{f}(x)$  is invertible in  $R_q$  and  $R_p$ . If  $\mathbf{f}(x)$  is invertible, then the inverses are easily computed in  $R_p$  and  $R_q$  by using Euclidean algorithm and Hensel's Lemma. Let  $\mathbf{F}_q(x)$  and  $\mathbf{F}_p(x)$  be the inverses of  $\mathbf{f}(x)$  in  $R_q$  and  $R_p$ , respectively. Alice next computes

$$\mathbf{h}(x) = \mathbf{F}_q(x) \star \mathbf{g}(x) \mod q.$$

The polynomial  $\mathbf{h}(x)$  is Alice's public key. The problem of distinguishing  $\mathbf{h}(x)$  from uniform elements in  $R_q$  is called *decision NTRU problem*. Whilst, the problem of finding the private key  $(\mathbf{f}(x), \mathbf{g}(x))$  is called, *search NTRU problem*.

Bob's plaintext is a polynomial  $\mathbf{m}(x) \in R$  whose coefficients are integers in the interval  $[-\frac{1}{2}(p-1), \frac{1}{2}(p-1)]$ . In other words, the plaintext  $\mathbf{m}(x)$  is the centerlift of a polynomial<sup>6</sup> in  $R_p$ . Thus, if p=3, then the message is the centerlift of a ternary polynomial. Bob chooses a random ephemeral key<sup>7</sup>  $\mathbf{r}(x) \in \mathcal{L}_r$  (in the original NTRU-HPS,  $\mathcal{L}_r = \mathcal{T}(d,d)$ ) and computes

(1.2) 
$$\mathbf{e}(x) \equiv p\mathbf{r}(x) \star \mathbf{h}(x) + \mathbf{m}(x) \mod q.$$

Finally, Bob sends to Alice the ciphertext  $\mathbf{e}(x) \in R_q$ .

To decrypt, Alice computes

$$\mathbf{a}(x) \equiv \mathbf{f}(x) \star \mathbf{e}(x) \mod q$$
.

Then, she center lifts  $\mathbf{a}(x)$  to an element of R say  $\mathbf{a}'(x)$ , and she finally computes,

$$\mathbf{b}(x) \equiv \mathbf{F}_p(x) \star \mathbf{a}'(x) \mod p$$
.

Then,  $\mathbf{b}(x)$  is equal to the plaintext  $\mathbf{m}(x)$  (this is true when a simple inequality between d, q and N is satisfied).

1.1.4. Background on lattices. We recall some well-known facts about lattices. Let  $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n$  be linearly independent vectors of  $\mathbb{R}^m$ . The set

$$\mathcal{L} = \left\{ \sum_{j=1}^{n} \alpha_{j} \mathbf{b}_{j} : \alpha_{j} \in \mathbb{Z}, 1 \leq j \leq n \right\}$$

is called a *lattice* and the finite vector set  $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$  is called a basis of the lattice  $\mathcal{L}$ . All the bases of  $\mathcal{L}$  have the same number of elements, say n, which is called *dimension* or rank of  $\mathcal{L}$ . If n = m, then the lattice  $\mathcal{L}$  is said to have full rank. We denote by M the  $n \times m$  matrix having as rows the vectors  $\mathbf{b}_1, \dots, \mathbf{b}_n$ .

<sup>&</sup>lt;sup>5</sup>in the original NTRU-HPS,  $\mathcal{L}_f = \mathcal{T}(d+1,d)$  (or  $\mathcal{L}_f = \{1+pG: G \in \mathcal{T}(d,d)\}$ ),  $\mathcal{L}_g = \mathcal{T}(d,d)$ . <sup>6</sup>that is, reduction of the coefficients into the interval (-p/2,p/2].

<sup>&</sup>lt;sup>7</sup>In NTRU-HRSS the public key is  $\mathbf{h}(x) = (x-1) \star \mathbf{F}_q(x) \star \mathbf{g}(x) \mod q$  and the choice of  $\mathbf{f}, \mathbf{g}, \mathbf{r}, \mathbf{m}$  is a little different from NTRU-HPS, but again all the polynomials are short ternary polynomials.

If  $\mathcal{L}$  has full rank, then the *volume* of the lattice  $\mathcal{L}$  is defined to be the positive number  $|\det M|$ . The volume, as well as the rank, are independent of the basis  $\mathcal{B}$ . It is denoted by  $vol(\mathcal{L})$  or  $\det \mathcal{L}$  (see also [16]). If  $\mathbf{v} \in \mathbb{R}^m$ , then  $||\mathbf{v}||$  denotes, as usually, the Euclidean norm of  $\mathbf{v}$ . Further, we denote by  $\lambda_1(\mathcal{L})$  the least of the lengths of vectors of  $\mathcal{L} - \{\mathbf{0}\}$ . Finally, if  $\mathbf{t} \in \mathbb{R}^m$ , then with  $dist(\mathcal{L}, \mathbf{t})$  we denote  $\min\{||\mathbf{v} - \mathbf{t}|| : \mathbf{v} \in \mathcal{L}\}$ .

There are two main problems in integer lattices. The Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP) and their approximation versions. SVP is defined as follows: Given a lattice  $\mathcal{L}$ , find a non zero vector  $\mathbf{u} \in \mathcal{L}$  such that, for every non zero  $\mathbf{u}' \in \mathcal{L}$  we have:

$$\|\mathbf{u}\| \le \|\mathbf{u}'\|$$
.

We define the approximate Closest Vector Problem  $CVP_{\gamma_n}(\mathcal{L})$  (for some  $\gamma_n \geq 1$ ) as follows: Given a lattice  $\mathcal{L}$  and a vector  $\mathbf{t} \in \mathbb{R}^m$ , find a lattice vector  $\mathbf{u}$  such that, for every  $\mathbf{u}' \in \mathcal{L}$  we have:

$$\|\mathbf{u} - \mathbf{t}\| < \gamma_n \|\mathbf{u}' - \mathbf{t}\|.$$

We say that we have a CVP oracle, if we have an efficient probabilistic algorithm that solves  $\text{CVP}_{\gamma_n}$ , for  $\gamma_n=1$ . To solve  $\text{CVP}_{\gamma_n}$ , we usually use Babai's algorithm [16, Chapter 18] (which has polynomial running time). In fact, combining this algorithm with the LLL algorithm, we solve  $\text{CVP}_{\gamma_n}(\mathcal{L})$  for some lattice  $\mathcal{L} \subset \mathbb{Z}^m$  having  $\gamma_n=2^{n/2}$  and  $n=rank(\mathcal{L})$ , in polynomial time. For more details on Babai algorithm see also [2, Section 2]

1.1.5. SVP and NTRU. Let the lattice  $L_{\rm h}$  generated by the rows of the matrix

$$M_{\mathbf{h}} = \begin{bmatrix} I_N & C(\mathbf{h}) \\ \mathbf{0}_N & qI_N \end{bmatrix},$$

where  $C(\mathbf{h})$  is the circulant matrix generated by the vector  $\mathbf{h}$ , see the definition in (1.1). This matrix is public, since it contains the public key of the NTRU cryptosystem. From  $\mathbf{f}(x) \star \mathbf{h}(x) \equiv \mathbf{g}(x) \pmod{q}$ , there is a polynomial  $\mathbf{b}(x) \in R$  such that  $\mathbf{f}(x) \star \mathbf{h}(x) - q\mathbf{b}(x) = \mathbf{g}(x)$ , so considering polynomials as vectors we get  $\mathbf{f}C(\mathbf{h}) - q\mathbf{b} = \mathbf{g}$ , thus  $(\mathbf{f}, -\mathbf{b})M_{\mathbf{h}} = (\mathbf{f}, \mathbf{g})$ . That is,  $(\mathbf{f}, \mathbf{g}) \in L_{\mathbf{h}}$ . We can see that

(1.3) 
$$L_{\mathbf{h}} = \{ (\mathbf{u}(x), \mathbf{v}(x)) \in R^2 : \mathbf{u}(x) \star \mathbf{h}(x) \equiv \mathbf{v}(x) \pmod{q} \}$$

or

$$L_{\mathbf{h}} = \{(\mathbf{u}, \mathbf{v}) \in \mathbb{Z}^{2N} : \mathbf{u}C(\mathbf{h}) = \mathbf{v}\}.$$

Thus, the problem of finding the private key  $(\mathbf{f}, \mathbf{g}) \in \mathbb{Z}_q^{2N}$ , comes down to find a short vector in the integer lattice  $L_{\mathbf{h}}$ . Note that,  $L_{\mathbf{h}}$  is a q-ary lattice, i.e.  $q\mathbb{Z}^{2N} \subset L_{\mathbf{h}}$ .

1.1.6. *CVP* and *NTRU*. Closest Vector Problem (CVP) can be used to recover the message and not the private key as previous in subsection 1.1.5. To see how we apply CVP to recover the message  $\mathbf{m}$ , we first note that  $(p\mathbf{r}, \mathbf{e} - \mathbf{m}) \in L_{\mathbf{h}}$ . Indeed, from the form of lattice  $L_{\mathbf{h}}$ , see (1.3), and from the encryption of  $\mathbf{m}$ , see (1.2), we get  $p\mathbf{r} \star \mathbf{h} = \mathbf{e} - \mathbf{m} \pmod{q}$ , therefore  $(p\mathbf{r}, \mathbf{e} - \mathbf{m}) \in L_{\mathbf{h}}$ . Now, we write

$$(\mathbf{0}_N, \mathbf{e}) = (p\mathbf{r} - p\mathbf{r}, (p\mathbf{r} \star \mathbf{h} + \mathbf{m}) \mod q) =$$
  
 $(p\mathbf{r}, p\mathbf{r} \star \mathbf{h} \mod q) + (-p\mathbf{r}, \mathbf{m}).$ 

Finally, the vector  $(p\mathbf{r}, p\mathbf{r} \star \mathbf{h} \bmod q) = (p\mathbf{r}, \mathbf{e} - \mathbf{m}) \in L_{\mathbf{h}}$  and the vector  $(-p\mathbf{r}, m)$  is quite short. To see this and assuming that p = 3 i.e.  $\mathbf{r}, \mathbf{m} \in \{-1, 0, 1\}^N$ , we get

$$dist((p\mathbf{r}, \mathbf{e} - \mathbf{m}), (\mathbf{0}, \mathbf{e})) = ||(p\mathbf{r}, \mathbf{e} - \mathbf{m}) - (\mathbf{0}, \mathbf{e})|| = ||(p\mathbf{r}, -\mathbf{m})|| \le \sqrt{10N}.$$

Therefore, in order to find  $\mathbf{m}$ , we apply CVP in  $L_{\mathbf{h}}$  with target vector  $(\mathbf{0}_N, \mathbf{e})$ . We generalize this method by using a modification of the target vector which is closer to the previous lattice, see Remark 3.1.

Roadmap. In section 2 we provide the bibliography and results that concerns attacks to NTRU and also we present our contribution. In section 3 we present some auxiliary results which we shall use in our attack in the next section 4. In subsection 4.1 we provide the algorithm of the attack and we depict in some examples, using Sagemath and Fpylll, the success of our attack. In the final section, we end up by presenting some concluding remarks and commenting possible future work.

#### 2. Previous work-Our Contribution

In 1997 Coppersmith and Shamir [13] proposed a lattice based attack on the NTRU cryptosystem. Their attack uses lattice reduction to find the private key<sup>8</sup>. In [18] the author proposed lattice attacks that are efficient, when N is composite, by reducing lattices of small dimension to find partial information about the secret key. Furthermore, in [26] May used a different class of lattices, which he called run-lattices. This class of lattices is constructed from the classic NTRU-lattice by multiplying columns N+1 till N+r by a constant  $\theta$ .

In [33], Silverman generalized May's idea and he proposed a method where he selects r coefficients and then force them equal to zero by reducing the dimension of the lattice. In [17] the authors present some new chosen-ciphertext attacks. The attacks exploit the decryption failures.

In [24], the authors presented the meet-in-the middle attack, which was first observed by Andrew Odlyzko. The idea of this attack is to split the search space for the secret key into two parts and use a collision search algorithm. Thus, one reduces the steps by a square root. The drawback is that it needs a significant amount of memory.

In [22] Howgrave-Graham proposed the improved Hybrid Attack. This is a combination of lattice reduction and meet-in-the-middle techniques. He compared his attack with the meet-in-the-middle proposed by Odlyzko and he deduced that the algorithm of Hybrid Attack requires about  $2^{60.3}$  loops in order to recover the private key of ee251ep6 parameter set. On the other hand Odlyzko's attack requires about  $2^{84.3}$  loops. What is more, it requires a factor of  $2^{24.6}$  less storage. This is the most practical attack, but it also needs exponential time. The Hybrid Attack has been used to estimate the security of many lattice-based cryptographic schemes. In [8, 36] the authors support that the analysis in [22] as well as those in other schemes are not entirely satisfactory, leading to unreliable estimates. Specifically the authors in [8] improved the analysis of the runtime of the attack by proving that in some cases, meet-in-the-middle attack is better than Hybrid Attack. Furthermore, they proposed a generalized version of the hybrid attack for solving SVP

 $<sup>^8</sup>$ This attack does not have any relation with the Coppersmith attack as used, for instance in RSA ([12]) or (EC)DSA (for instance see [7, 14])

and BDD problems in q-ary lattices. In 2016, Albrecht, Bai and Ducas [3] and independently Cheon, Jeong and Lee [9] proposed much the same methods to attack the NTRU cryptosystem with larger modulus than in the NTRUEncrypt standard. The main idea is to decrease the dimension of the NTRU lattice using the multiplication matrix by the norm (resp. trace) of the public key in some subfield. In [20] the authors presented a new variant of the subfield attacks which is better than both of the two previous attacks in practice. They proved that in  $\mathbb{Q}(\zeta_{2^n})$ , the time complexity is polynomial for  $q = 2^{\Omega(\sqrt{n\log\log n})}$ . Furthermore, they made a comparison between this attack and the hybrid attack, concluding that hybrid attack is better in practice.

Finally, in 2021 Nguyen [27] analyses the meet in the middle and the hybrid attack by making some simplifications and further improvements. What is more, he deduced that the security estimates of the NTRU finalist in NIST's post-quantum standardization need to be revised.

Another very interesting line of research presented in 2005 by Silverman, Smart and Vercaturen in [34] and extended in 2009 by Bourgeois and Faugère in [10], where they used Witt vectors to reduce the NTRU problem to a multivariate quadratic system over the Galois field with two elements.

2.1. Our contribution. There is the classic message recovery attack using closest vector problem, see subsection 1.1.6. In the present paper we generalize the classic CVP attack. In fact we apply a CVP attack to recover the message **m** without using the NTRU lattice based on the public key. We remark here that there not many results in this direction, i.e. attacks based on CVP. On the other hand, there are plenty of such results, i.e. based on CVP, for other cryptosystems, for instance (EC)DSA (Digital Signatur Algorithm). To our knowledge, in the bibliography, there are not results, concerning attacks to NTRU that are based on CVP on some lattice, except the classic that we have already presented in subsection 1.1.6.

In fact we apply CVP to a lattice  $\mathcal{L}_{\mathbf{a}}$  (instead of  $L_{\mathbf{h}}$ ) for some fixed and suitably chosen vector  $\mathbf{a} \in \mathbb{Z}^{2N}$  and target vector  $(\mathbf{0}_N, \mathbf{a} \star \mathbf{e} + \mathbf{E})$  (instead of  $(\mathbf{0}_N, \mathbf{e})$  in the classic CVP attack), for some suitable vector  $\mathbf{E}$ . The new lattice does not depend on the public polynomial  $\mathbf{h}(x)$  and depends only on N, q, and a (real) parameter y. The new idea here is that, this new lattice  $\mathcal{L}_{\mathbf{a}}$  allows us to provide a deterministic attack to NTRU under the assumption that we know an approximation of the unknown vector  $\mathbf{E}$  and assuming that we have a CVP oracle.

Furthermore, in practice we implement the attack using Babai's nearest plane algorithm on the lattice  $\mathcal{L}_{\mathbf{a}}$ , where the vector  $\mathbf{a}$  depends on N,q, and a fixed real parameter y. The target vector, as we shall see, is a sum of two vectors. The first vector is known, but the second vector needs some guesses on the part of the attacker. The unknown part of target vector is a multiple of the nonce  $\mathbf{r}$ . If, for instance, we have a weak generator for  $\mathbf{r}$ , we might predict some digits of ternary polynomial  $\mathbf{r}$  and so we get a better estimate for the target vector. In our examples, the previous approximation is provided by a suitable oracle. It may seem unrealistic to have such an oracle in practice, but a weak random generator or an implementation of a side channel attack, it could provide us such an oracle.

Finally, since the lattice does not depend on the public key  $\mathbf{h}$ , we can use a LLL/BKZ reduction only one time, as far as (N,q,y) remain the same. So, if the keys changed, we do not have to repeat the reduction step of the closest vector problem.

As far as we know this attack is new. We shall present some experiments, where we show that our attack is successful. In fact, in subsection  $4.1.1^9$ , in examples 5 and 6, we consider the state of the art parameters: ntruhps2048509 and ntruhps2048677 i.e. (N,q)=(509,2048) and (677,2048), respectively. These parameters were recommended in the 3rd NTRU submission in NIST [11].

## 3. Auxiliary Results

Let N > 2 be a prime number and  $\mathbf{a} = (a_0, \dots, a_{N-1})$  be a random vector of  $R_q$ . With  $\mathcal{L}_{\mathbf{a}}$  we denote the lattice generated by the rows of the matrix,

(3.1) 
$$M_{\mathbf{a}} = \begin{bmatrix} 1 & 0 & \dots & 0 & a_0 & a_1 & \dots & a_{N-1} \\ 0 & 1 & \dots & 0 & a_{N-1} & a_0 & \dots & a_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & a_1 & a_2 & \dots & a_0 \\ \hline 0 & 0 & \dots & 0 & q & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & q \end{bmatrix}$$

For  $\mathcal{L}_{\mathbf{a}}$  we make the following assumption:

**Assumption**. Let q be a positive integer and  $y \ge 1$  be a real number. We assume that

(3.2) 
$$\lambda_1(\mathcal{L}_{\mathbf{a}}) > q^{1/y}.$$

Since it is not possible to compute  $\lambda_1$  for large values of N, we can not really check the validity of this inequality. We shall use Gaussian heuristic to get a more practical form of this inequality. The Gaussian heuristic  $GH(\mathcal{L}_{\mathbf{a}})$  for  $L_{\mathbf{a}}$  suggests that the length of a shortest vector is approximately,

$$\sqrt{\frac{2N}{2\pi e}} \det(M_{\mathbf{a}})^{1/2N} = \sqrt{\frac{qN}{\pi e}} \approx 0.35 \times \sqrt{qN}.$$

However, according to [21, Proposition 6.61] the private key is about  $\frac{1}{\sqrt{N}}$  the bound suggested by Gaussian heuristic. So, we update the previous bound :

$$\lambda_1 \approx \frac{1}{\sqrt{N}} 0.35 \times \sqrt{qN} = 0.35 \sqrt{q}.$$

Now, under the previous heuristic, the assumption is written,

$$(3.3) 0.35\sqrt{q} > q^{1/y}.$$

It is easy to check that there are plenty of (N, q, y) that satisfy the two previous inequalities. For instance, see Appendix B, for some examples where the heuristic inequality (3.3) and the inequality (3.2) are both valid.

The following Proposition will allows us to implement an attack on NTRU.

## Proposition 3.1. Let

$$\mathbf{a}(x) = a_0 + a_1 x + \dots + a_{N-1} x^{N-1}$$

<sup>&</sup>lt;sup>9</sup>For the code see https://github.com/drazioti/ntru

be a polynomial in  $R_q$ . Consider the equation in  $R_q$ ,

$$\mathbf{a}(x) \star \mathbf{m}(x) = \mathbf{b}(x) + \mathbf{c}(x),$$

where the unknowns are the polynomials

$$\mathbf{m}(x) = \sum_{j=0}^{N-1} m_j x^j, \quad \mathbf{c}(x) = \sum_{j=0}^{N-1} c_j x^j,$$

and  $\mathbf{b}(x) = b_0 + b_1 x + \dots + b_{N-1} x^{N-1}$  with  $b_0, \dots, b_{N-1} \in \{0, \dots, q-1\}$ . Let  $\mathbf{V} = (\mathbf{m}, \mathbf{c})$  be a vector with coordinates the coefficients of a solution  $(\mathbf{m}(x), \mathbf{c}(x))$ . If we can find a vector  $\mathbf{E} = (E_0, E_1, \dots, E_{N-1}, E_N, \dots, E_{2N-1}) \in \mathbb{Z}^{2N}$  satisfying

$$\|\mathbf{V} - \mathbf{E}\| < \frac{1}{2}q^{\frac{1}{y}},$$

then we can determine the solution vector  $\mathbf{V}$  using a CVP oracle.

*Proof.* Let  $\mathbf{u} = (m_0, m_1, \dots, m_{N-1}, b_0 + c_0, \dots, b_{N-1} + c_{N-1})$ . We construct the lattice  $\mathcal{L}_{\mathbf{a}}$  spanned by the rows of the  $2N \times 2N$  matrix:

$$M_{\mathbf{a}} = \left[ \begin{array}{c|c} I_N & C(\mathbf{a}) \\ \hline \mathbf{0}_N & qI_N \end{array} \right]$$

It is easy to see that **u** is a lattice point. Indeed, there is a polynomial  $\mathbf{v}(x) \in R$  such that,

$$\mathbf{a}(x) \star \mathbf{m}(x) = \mathbf{b}(x) + \mathbf{c}(x) + q\mathbf{v}(x).$$

Then,

$$\begin{aligned} (\mathbf{m}, -\mathbf{v}) M_{\mathbf{a}} &= (\mathbf{m}, -\mathbf{v}) \left[ \frac{I_N \mid C(\mathbf{a})}{\mathbf{0}_N \mid q I_N} \right] = (\mathbf{m}, \mathbf{m} C(\mathbf{a}) - q \mathbf{v}) = \\ (\mathbf{m}, \mathbf{m} \star \mathbf{a} - q \mathbf{v}) &= (\mathbf{m}, \mathbf{a} \star \mathbf{m} - q \mathbf{v}) = (\mathbf{m}, \mathbf{b} + \mathbf{c}) = \mathbf{u}. \end{aligned}$$

Let

$$\mathbf{b}_{target} = (E_0, E_1, \dots, E_{N-1}, b_0 + E_N, \dots, b_{N-1} + E_{2N-1}).$$

Remark that,

$$\mathbf{b}_{target} = \mathbf{E} + (\mathbf{0}_N, \mathbf{b}).$$

So,

$$\mathbf{b}_{target} + \mathbf{V} = \mathbf{E} + (\mathbf{0}_N, \mathbf{b}) + \mathbf{V} = \mathbf{E} + (\mathbf{m}, \mathbf{b} + \mathbf{c}) = \mathbf{E} + \mathbf{u}.$$

Therefore,

$$\|\mathbf{u} - \mathbf{b}_{target}\| = \|\mathbf{V} - \mathbf{E}\| < \frac{1}{2} q^{\frac{1}{y}}.$$

We call a CVP Oracle with input the lattice  $\mathcal{L}_{\mathbf{a}}$  and target vector  $\mathbf{b}_{target}$  and we get a vector  $\mathbf{w} \in \mathcal{L}_{\mathbf{a}}$  such that,

$$\|\mathbf{w} - \mathbf{b}_{target}\| \le \|\mathbf{u} - \mathbf{b}_{target}\| = \|\mathbf{V} - \mathbf{E}\| < \frac{1}{2} q^{\frac{1}{y}}.$$

So we get,

$$\|\mathbf{w} - \mathbf{u}\| \le \|\mathbf{w} - \mathbf{b}_{target}\| + \|\mathbf{u} - \mathbf{b}_{target}\| < \frac{1}{2} q^{\frac{1}{y}} + \frac{1}{2} q^{\frac{1}{y}} = q^{\frac{1}{y}}.$$

Our assumption (3.2) implies that every lattice vector of  $\mathcal{L}_{\mathbf{a}}$  has length at least  $q^{1/y}$ . Since  $\mathbf{u}$  and  $\mathbf{v}$  are lattice vectors, then also  $\mathbf{u} - \mathbf{v}$  is a lattice vector, so  $||\mathbf{u} - \mathbf{v}|| > q^{1/y}$ . Therefore,  $\mathbf{w} - \mathbf{u} = \mathbf{0}$  or  $\mathbf{w} = \mathbf{u}$ . The result follows.

**Remark 3.1.** We note that, if we take the original lattice  $L_h$ , we can have a similar Proposition for the classic CVP (see 1.1.5), taking as target vector  $(\mathbf{0}_N, \mathbf{e}) + \mathbf{E}$ . So, the Proposition generalizes the classic CVP attack.

## 4. The Attack to NTRU

In this section we show how we can find the plaintext  $\mathbf{m}(x)$  using the previous Proposition 3.1. In this Proposition we provide only a sufficient condition. It may happen to find the message, even if the inequality is not being satisfied. The encrypted message  $\mathbf{e}(x)$  is given by the following relation,

$$\mathbf{e}(x) \equiv p\mathbf{r}(x) \star \mathbf{h}(x) + \mathbf{m}(x) \mod q$$
.

By multiplying both sides with  $\mathbf{a}(x)$  (where  $\mathbf{a}(x)$  is a random polynomial of  $\mathcal{R}_q$ ) we get,

$$\mathbf{a}(x) * \mathbf{m}(x) \equiv \mathbf{a}(x) * \mathbf{e}(x) - p\mathbf{a}(x) * \mathbf{r}(x) * \mathbf{h}(x) \mod q$$

or

(4.1) 
$$\mathbf{a}(x) * \mathbf{m}(x) \equiv \mathbf{b}(x) + \mathbf{c}(x) \mod q.$$

We have set

$$\mathbf{b}(x) = \mathbf{a}(x) * \mathbf{e}(x) \mod q$$

and

$$\mathbf{c}(x) = -p\mathbf{a}(x) * \mathbf{r}(x) * \mathbf{h}(x) \mod q.$$

In equation (4.1),  $\mathbf{a}(x)$  and  $\mathbf{b}(x)$  are known. We assume that, the following vector

(4.2) 
$$\mathbf{V} = (m_0, m_1, \dots, m_{N-1}, c_0, c_1, \dots, c_{N-1})$$

is a solution of equivalence (4.1). We construct the lattice  $\mathcal{L}_{\mathbf{a}}$  spanned by the rows of the  $2N \times 2N$  matrix  $M_{\mathbf{a}}$  as in (3.1). Let  $\mathbf{E} = (E_0, E_1, \dots, E_{N-1}, E_N, \dots, E_{2N-1}) \in \mathbb{Z}^{2N}$  such that,

$$\|\mathbf{V} - \mathbf{E}\| < \frac{1}{2}q^{\frac{1}{y}}.$$

We call our CVP Oracle with input the lattice  $\mathcal{L}_{\mathbf{a}}$  and target vector

$$\mathbf{b}_{target} = (E_0, E_1, \dots, E_{N-1}, b_0 + E_N, \dots, b_{N-1} + E_{2N-1}).$$

Say that we get a vector  $\mathbf{w} \in \mathcal{L}_{\mathbf{a}}$ . Then, Proposition 3.1 yields

$$w_0 = m_0, \dots, w_{N-1} = m_{N-1}.$$

In this way we compute the message  $\mathbf{m}(x)$ .

4.1. **The Attack.** In this subsection, we present our message recovery attack based on the previous analysis.

#### Algorithm 1

INPUT: N: prime,  $y \in \mathbb{R}_{>1}$ ,  $\mathbf{e}$  the encryption of a message  $\mathbf{m}$ ,  $\mathbf{E} \in \mathbb{Z}^{2N}$ , and q which is a prime or a prime power.

OUTPUT: a message m'

- 1.  $\mathbf{a} \stackrel{\$}{\leftarrow} \{0,1\}^{N-1} \times \{|Nq^{1/y}\}|$
- 2. Construct the matrix  $M_{\mathbf{a}}$  and the lattice  $\mathcal{L}_{\mathbf{a}}$  generated by the rows of  $M_{\mathbf{a}}$
- 3.  $\mathbf{b} \leftarrow \mathbf{a} \star \mathbf{e} \text{ in } R_q$
- 4.  $\mathbf{b}_{target} \leftarrow (\mathbf{0}_N; \mathbf{b}) + \mathbf{E}$
- 5.  $\mathbf{w} \leftarrow \mathtt{babai}(\mathcal{L}_{\mathbf{a}}, \mathbf{b}_{target})$

6.  $\mathbf{m}' \leftarrow \mathbf{w}[0:N-1] \# \text{ the } N\text{-first coordinates of } \mathbf{w}$ 7. return m'

Since NTRUEncrypt is used in Key Encapsulation Mechanisms, with the previous attack we expect to find the shared key. Indeed, if the message is right, then we can compute the nonce  $\mathbf{r}$  and then taking a suitable hash we get the shared key. The validity of the key can only be checked indirectly, by starting some encrypted conversation with the owner of the secret key, by using a symmetric cryptosystem.

In line 1, we choose small entries for the vector a except one which is near to  $Nq^{1/y}$ . Someone would expect to pick **a** randomly from  $R_q$ . There are some theoretic arguments for this choice. We explain this in Appendix A. In line 6, we return the first N coordinates of w, i.e.  $w_0, ..., w_{N-1}$ . In line 5 before applying Babai's algorithm, we reduce the basis by using LLL. The LLL reduction is the same for all (N,q,y), so we can make this step independently from the others, i.e. we can apply LLL in line 2.

We tried the previous attack for various parameters. In all the examples we assume that  $\mathbf{E} = (\mathbf{r}_N, \mathbf{E}')$ , where  $\mathbf{r}_N \in \{-1, 0, 1\}^N$ ,  $\mathbf{E}' = (E'_1, ..., E'_N)$  with  $|E'_i - c_i| \leq R$ , and  $c_i$  are the coefficients of  $\mathbf{c}(x)$ . Note that,  $\mathbf{c}(x)$  is the product in  $R_q$  of  $-p \star \mathbf{h}(x) \star \mathbf{a}(x)$  and the unknown nonce  $\mathbf{r}(x)$ , which is a short ternary polynomial. Furthermore, whenever the attack succeeded, we recovered the message  $\mathbf{m}$ , in less than a minute. For the implementation we used Sagemath [31] and for the LLL reduction and Babai algorithm, we used fpylll library [15]. Before we present our experiments, we describe our attack by using an oracle.

Say we have an oracle  $\mathcal{O}$  that on input the public key of the system, the parameter R, and a seed, provides us with some integers  $E'_i$ , i = 1, 2, ..., N, such that  $|E'_i - c_i| \le 1$ R. Each call to the oracle with a different seed provides a new random set of  $E'_i$ with the same property. We assume that the distribution of  $E'_i$  with this property is uniform.

4.1.1. Examples. In our examples,  $\mathcal{L}_f = \mathcal{T}(d+1,d)$  and  $\mathcal{L}_g = \mathcal{L}_r = \mathcal{T}(d,d)$ . In the document submitted to the third phase of NIST [11], the authors recommend for the case of NTRU-HPS,  $\mathcal{L}_f$  be the set of ternary polynomials (i.e. with coefficients in  $\{-1,0,1\}$ ) of degree at most N-2. In examples 1-4, we follow Algorithm 1 for the choice of vector a. All the examples use LLL and Babai's algorithm, thus are very fast.

**Example 1.**  $N = 239, d = 71, p = 3, q = 2^8 = 256, y = 2.3, \mathbf{r}_N = \mathbf{0}.$ 

In this experiment for R=9 we got the message after some calls (< 100). For R = 10 we did not manage to find the message.

Example 2.  $N = 257, d = 91, p = 3, q = 2^8 = 256, y = 2.3, \mathbf{r}_N = \mathbf{0}.$ 

In this experiment for R=9 after some calls (< 100) we got the message. For R = 10 we did not manage to find the message after 100 calls to the oracle.

**Example 3.**  $N = 283, d = 99, p = 3, q = 2^{10} = 1024, y = 2.3, \mathbf{r}_N = \mathbf{0}.$ 

In this experiment for R=16 after some calls (< 100) we got the message. For R = 17 we did not manage to find the message.

**Example 4.**  $N = 307, d = 15, p = 3, q = 2^{10} = 1024, y = 2.5, \mathbf{r}_N = \mathbf{0}.$ 

In this experiment for R = 18 we got the message after at most 100 calls to the oracle. For R = 19 we did not manage to find the message.

**Example 5.**  $N = 509, d = 10, p = 3, q = 2^{11}, y = 2.5, \mathbf{r}_N = \mathbf{0}.$  For this experiment we picked **a** from  $\{-2, 2\}^{N-1} \times \{\lfloor Nq^{1/y} \rfloor\}$  and we randomly

shuffle it. In this experiment for R=26 after at most 100 calls we got the message. For R=27 we did not manage to find the message after 100 calls. The specific parameters were suggested in [11] for ntruhps2048509.

For the next examples we used the following,

$$\mathbf{a} = (-k, -k+1, ..., -1, 1, 2, ..., k, \lfloor Nq^{1/y} \rfloor + 1),$$

where  $k = \frac{N-1}{2}$ .

**Example 6.**  $N = 677, d = 20, p = 3, q = 2^{11}, y = 2.5, \mathbf{r}_N = \mathbf{0}.$ 

The authors of [11] for better security suggested ntruhps2048677. For R=17 we got almost immediately the message, assuming that we have the LLL-reduced basis. The LLL reduction took about 15 minutes in Fpylll [15]. For R=18 after 100 calls to the oracle we did not find the message.

Example 7.  $N = 557, d = 40, p = 3, q = 2^{13}, y = 2.5, \mathbf{r}_N = \mathbf{0}.$ 

In this experiment for R=38 after at most 100 calls we got the message. For R=39 we did not manage to find the message after 100 calls. The parameters  $(N,q)=(557,2^{13})$  were recommended in [19].

**Remark 4.1.** In the previous examples the heuristic inequality (3.3) is not satisfied (but the two sides of the inequality are very close). Proposition 3.1 is not if and only if statement, i.e. if assumption (3.2) is satisfied then we get the result of the Proposition. However, it may occur, the assumption be false and the attack may still works.

**Remark 4.2.** The experiments suggest that if we increase  $q = 2^e$ , greater values for the range R are allowed in order to get a successful attack.

**Remark 4.3.** Assume that  $\mathbf{r}_N = \mathbf{0}$ , p = 3 and for some  $E_i'$  we have  $|E_i' - c_i| < R$   $(1 \le i \le N)$ . We consider the notation of section 4. Then,

$$||\mathbf{V} - \mathbf{E}||^2 = ||\mathbf{m}||^2 + \sum_{i=1}^{N} (E_i' - c_i)^2 \le N + R^2 N = N(1 + R^2).$$

Also if  $N(1+R^2) < \frac{1}{4}q^{2/y}$  or

$$y < \frac{2\log_2 q}{2 + \log_2(N(1+R^2))},$$

then from Proposition 3.1 we get that a CVP oracle will provide us the solution vector  $\mathbf{V}$ .

### 5. Conclusion

In this work we used lattice theory to attack the NTRU-HPS cryptosystem. Multiplying the encryption equation with a (random) polynomial, we create an equivalent equation, where an unknown value belongs to a lattice, for which we can find a lower bound of the first successive minima of the lattice. Then, we choose a suitable target vector and apply Babai's nearest plane algorithm with the hope that the output is the unknown value. The difficult part is the choice of vector  $\mathbf{E}$ , thus we need the help of a suitable oracle. This is a drawback of the attack, since it is difficult in practice to have such an oracle. To address this problem someone has to apply a side channel attack, hoping to get some information about the vector  $\mathbf{c}$  of Proposition 3.1. Then, the attacker may have a good guess for the vector  $\mathbf{E}$ .

For instance, side channel attacks were studied in [4, 25, 35]. Finally, we provided several examples showing the success of the attack with the use of the oracles.

Acknowledgment. Marios Adamoudis is co-financed by Greece and the Eu-







ropean Union (European Social Fund-ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning" in the context of the Act "Enhancing Human Resources Research Potential by undertaking a Doctoral Research" Sub-action 2: IKY Scholarship Programme for PhD candidates in the Greek Universities.

Finally, the authors sincerely thank professor Poulakis for his helpful suggestions.

#### References

- [1] Marios Adamoudis, Konstantinos A. Draziotis, and Dimitrios Poulakis, Enhancing an attack to DSA schemes, CAI 2019, p.13–25, LNCS 11545, Springer 2019.
- [2] Marios Adamoudis, Konstantinos A. Draziotis, and Dimitrios Poulakis, Attacking (EC)DSA With Partially Known Multiples of Nonces, Cryptology ePrint Archive, Report 2021/347, 2021, https://eprint.iacr.org/2021/347.
- [3] M. Albrecht, S. Bai, and L. Ducas, A subfield lattice attack on overstretched NTRU assumptions. CRYPTO 2016. LNCS 9814, Springer 2016.
- [4] F. Aydin, Aydin Aysu, M. Tiwari, A. Gerstlauer, and M. Orshansky, Horizontal Side-Channel Vulnerabilities of Post-Quantum Key Exchange and Encapsulation Protocols, ACM Transactions on Embedded Computing Systems Vol. 20 Issue 6, 2021, https://doi.org/10.1145/ 3476799.
- [5] Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang. NTRU Prime. Round 3 submission to NIST post-quantum call for proposals, 2020, https://ntruprime.cr.yp.to/warnings.html.
- [6] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal, NTRU prime: Reducing attack surface at low cost. SAC 2017: 24th Annual International Workshop on Selected Areas in Cryptography, LNCS 10719, Springer, Heidelberg, 2017.
- [7] I. F. Blake and T. Garefalakis, On the security of the digital signature algorithm, Des. Codes Cryptogr., 26 (2002), p.87–96.
- [8] J. A. Buchmach, F. Göpfert, R. Player, and T. Wunderer. On the hardness of LWE with binary error: Revisiting the hybrid lattice-reduction and meet-in-the-middle attack. AFRICACRYPT 2016, LNCS 9646, Springer 2016.
- [9] J. H. Cheon, J. Jeong, and C. Lee, An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without an encoding of zero. Cryptology ePrint Archive, Report 2016/139, 2016.
- [10] Gerald Bourgeois and Jean-Charles Faugère, Algebraic attack on NTRU using Witt vectors and Gröbner bases, Journal of Mathematical Cryptology 3(3) p. 205–214, 2009.
- [11] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hülsing, Joost Rijneveld, John M. Schanck, Tsunekazu Saito, Peter Schwabe, William Whyte, Keita Xagawa, Takashi Yamakawa, and Zhenfei Zhang, Algorithm Specifications And Supporting Documentation, The round 3 NIST submission package, https://ntru.org.
- [12] Don Coppersmith, Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptology 10 (1997), p.233–260.
- [13] D. Coppersmith and A. Shamir, Lattice Attacks on NTRU. In Proc. Eurocrypt 1997, LNCS 1223.
- [14] K. A. Draziotis, DSA lattice attacks based on Coppersmith's method, Information Processing Letters 116 (8), (2016).
- [15] Fpylll, The FPLLL development team, fpylll, a Python wraper for the fplll lattice reduction library, Version: 0.5.6, 2021, available at https://github.com/fplll/fpylll.

- [16] S. Galbraith, Mathematics of Public key Cryptography, Cambridge university press, 2012.
- [17] N. Gama and Phong Q. Nguyen, New Chosen-Ciphertext Attacks on NTRU. Public Key Cryptography – PKC 2007, LNCS 4450, Springer 2007.
- [18] C. Gentry, Key recovery and message attacks on NTRU-composite, EUROCRYPT 2001, LNCS 2045, Springer 2001.
- [19] Andreas Hülsing, Joost Rijneveld, John Schanck, and Peter Schwabe, High-speed key encapsulation from NTRU. Cryptographic Hardware and Embedded Systems CHES 2017, LNCS 10529, Springer-Verlag 2017.
- [20] P. Kirchner and P. A. Fouque, Revisiting Lattice Attacks on Overstretched NTRU Parameters. EUROCRYPT 2017, LNCS 10210, Springer 2017, https://link.springer.com/chapter/10.1007/978-3-319-66787-4\_12
- [21] J. Hoffstein, J. Pipher, and J. H. Silverman, NTRU: A ring-based public key cryptosystem, in Proceedings of ANTS '98 (ed. J. Buhler), LNCS 1423, p. 267–288, 1998.
- [22] N. Howgrave-Graham. A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU. CRYPTO 2007, LNCS 4622, Springer 2007.
- [23] N. Howgrave-Graham, J. H. Silverman, and W. Whyte, Choosing parameter sets for NTRU-Encrypt with NAEP and SVES-3, CT-RSA 2005, LNCS 3376, Springer 2005.
- [24] N. Howgrave-Graham, J. H. Silverman, and W. Whyte, Meet-in-the-middle Attack on an NTRU private key, Technical report, NTRU Cryptosystems, July 2006. Report 04, available at http://www.ntru.com.
- [25] A. A. Kamal and A. M. Youssef, A Scan-Based Side Channel Attack on the NTRUEncrypt Cryptosystem, ARES '12: Proceedings of the 2012 Seventh International Conference on Availability, Reliability and Security, 2012, p. 402–409, https://doi.org/10.1109/ARES.2012.14
- [26] A. May, Cryptanalysis of NTRU (preprint), 1999, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.3484.
- [27] Phong Q. Nguyen, Boosting the Hybrid Attack on NTRU: Torus LSH, Permuted HNF and Boxed Sphere, Computer Science, Mathematics, 2021.
- [28] NTRU Prime FAQ team, FAQ, https://ntruprime.cr.yp.to/faq.html, Accessed 1 January 2022.
- [29] Peter W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring. In 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994, p. 124–134. IEEE Computer Society, 1994.
- [30] D. Poulakis, New lattice attacks on DSA schemes, J. Math. Cryptol. 10 (2), p. 135–144, 2016.
- [31] Sage Mathematics Software, The Sage Development Team (version 8.1). http://www.sagemath.org.
- [32] Andreas Salvanos, The NTRU cryptosystem and attacks on the private key, Master Thesis, 2018, Math. Department, Aristotle University of Thessaloniki, Greece, http://ikee.lib.auth.gr/record/303247/files/GRI-2019-23739.pdf.
- [33] J. H. Silverman, Dimension-Reduced Lattices, Zero-Forced Lattices, and the NTRU Public Key Cryptosystem. Technical Report 13, Version 1, NTRU Cryptosystems, 1999.
- [34] H. Silverman, N. P. Smart, and F. Vercauteren, An algebraic approach to NTRU (q=2n) via Witt vectors and overdetermined systems of non linear equations. Security in Communication Networks SCN 2004, LNCS **3352**, p. 278–298. Springer, 2005.
- [35] Nikolay Vasilev Vizev, Side Channel Attacks on NTRUEncrypt, Bachelor Thesis, University of Technology Darmstadt, Department of Computer Science, 2007.
- [36] T. Wunderer, A detailed analysis of the hybrid lattice-reduction and meet-in-the-middle attack, Journal of Mathematical Cryptology, vol. 13, no. 1, 2019, p. 1–26. https://doi.org/ 10.1515/jmc-2016-0044.

## APPENDIX A. HOW TO PICK a?

**Proposition A.1.** Let q and N be positive integers with N = 2k + 1, and y a real number > 1. We set,

$$a_j(N,q,y) = a_j = \begin{cases} j-k, & j = 0, 1, ..., k-1 \\ j+1-k, & j = k, k+1, ..., 2k-1 = N-2 \\ \lfloor Nq^{1/y} \rfloor + 1, & j = 2k = N-1 \end{cases}$$

For instance if  $N = 11, q = 2^5, y = 1$  we get

$$\mathbf{a} = (-5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 353).$$

We assume that.

(A.1) 
$$a_0^2 + a_1^2 + \dots + a_{N-2}^2 + a_{N-1}^2 = \frac{(N^2 - 1)N}{12} + a_{N-1}^2 < \frac{1}{N} \frac{(q - q^{1/y})^2}{q^{2/y}}$$

Let  $\mathbf{a} = (a_j)_j$  and  $\mathcal{L}_{\mathbf{a}} = \mathcal{L}(M_{\mathbf{a}})$  be the lattice spanned by the rows of the  $2N \times 2N$  matrix

$$M_{\mathbf{a}} = \begin{bmatrix} I_N & C(\mathbf{a}) \\ \mathbf{0}_N & qI_N \end{bmatrix}$$

Then, for every  $v \in \mathcal{L}_a - \{0\}$  not belonging to the lattice generated by the rows of the matrix

$$[I_N \mid C(\mathbf{a})],$$

we have:

$$\|\mathbf{v}\| > q^{\frac{1}{y}}.$$

*Proof.* Assume that there is a non-zero vector  $\mathbf{v} \in \mathcal{L}_{\mathbf{a}}$  such that,

$$\|\mathbf{v}\| \le q^{\frac{1}{y}}.$$

Let  $\mathbf{b}_1, \dots, \mathbf{b}_{2N}$  be the rows of the matrix  $M_{\mathbf{a}}$ . We define the following symbol, for a integer and n positive integer,

$$[a]_n = \begin{cases} a+n, & a < 0 < n \\ a, & 0 \le a < n \end{cases}$$

Since  $\mathbf{v} \in \mathcal{L}_{\mathbf{a}}$ , there are integers  $l_1, \ldots, l_{2N}$  such that

$$\mathbf{v} = l_1 \mathbf{b}_1 + \dots + l_{2N} \mathbf{b}_{2N} =$$

$$\left(l_1, l_2, \dots, l_N, \sum_{j=1}^N l_j a_{[1-j]_N} + q l_{N+1}, \sum_{j=1}^N l_j a_{[2-j]_N} + q l_{N+2}, \dots, \sum_{j=1}^N l_j a_{[N-j]_N} + q l_{2N}\right).$$

Then we get

(A.2) 
$$\begin{cases} |l_{1}|, |l_{2}|, \dots, |l_{N}| \leq q^{\frac{1}{y}} \\ |l_{1}a_{0} + l_{2}a_{N-1} + \dots + l_{N}a_{1} + ql_{N+1}| \leq q^{\frac{1}{y}} \\ |l_{1}a_{1} + l_{2}a_{0} + \dots + l_{N}a_{2} + ql_{N+2}| \leq q^{\frac{1}{y}} \\ \dots \\ |l_{1}a_{N-1} + l_{2}a_{N-2} + \dots + l_{N}a_{0} + ql_{2N}| \leq q^{\frac{1}{y}} \end{cases}$$

Let  $(i_1, ..., i_N)$  be a random right rotation of (0, 1, ..., N-1). By the Cauchy-Schwarz inequality and relations (A.1) and (A.2) we obtain:

$$|l_1 a_{i_1} + l_2 a_{i_2} + \dots + l_N a_{i_n}|^2 \le (l_1^2 + l_2^2 + \dots + l_N^2)(a_0^2 + a_1^2 + \dots + a_{N-1}^2) \le (q^{\frac{2}{y}} + q^{\frac{2}{y}} + \dots + q^{\frac{2}{y}})(a_0^2 + a_1^2 + \dots + a_{N-1}^2) = Nq^{\frac{2}{y}}(a_0^2 + a_1^2 + \dots + a_{N-1}^2) < (q - q^{\frac{1}{y}})^2.$$
 Therefore, we have

(A.3) 
$$|l_1 a_{i_1} + l_2 a_{i_2} + \dots + l_N a_{i_n}| < q - q^{\frac{1}{y}}.$$

Since v does not belong to the lattice generated by the rows of the matrix,

$$[I_N \mid C(\mathbf{a})],$$

not all the integers  $l_{N+1}, l_{N+2}, \dots, l_{2N}$  will be zero. Say  $l_{N+1} \neq 0$ . Thus, we get

$$\|\mathbf{v}\| \ge |l_1 a_0 + l_2 a_{N-1} + \dots + l_N a_1 + l_{N+1} q| \ge |l_{N+1}|q - |l_1 a_0 + l_2 a_{N-1} + \dots + l_N a_1| \ge q - |l_1 a_0 + l_2 a_{N-1} + \dots + l_N a_1| > q^{\frac{1}{y}},$$

which is a contradiction. The Proposition follows.

This Proposition provide us some constraints on how to choose **a**. The advantage here is that, we do not need the inequality (3.2), which is our assumption. Inequality (A.1) can be checked easily for every N, q, and y, whereas our assumption is hard to be checked for large values of N.

Appendix B. Some examples that satisfy the inequalities (3.2) and (3.3)

To produce instances<sup>10</sup> that satisfy the two inequalities we need to execute exact SVP to compute  $\lambda_1$ . We use moderate values of N and the SVP function of Fpylll. The heuristic inequality (3.3) may be true, whereas the assumption (3.2) may be false. Usually, for large enough values of y both inequalities are satisfied. For instance, for

$$(N,q,y) = (21, 2^5, 1.5), (23, 2^5, 2), (25, 2^9, 2), (27, 2^9, 2)$$

and for randomly chosen vector a, both the inequalities are satisfied.

Department of Mathematics, Aristotle University of Thessaloniki, 54 124, Thessaloniki, Greece

Email address: aamarios@math.auth.gr

Department of Informatics, Aristotle University of Thessaloniki, 54 124, Thessaloniki, Greece

Email address: drazioti@csd.auth.gr

 $<sup>^{10}</sup>$ For the code see https://github.com/drazioti/ntru/blob/main/appendix.ipynb