Key recovery attacks on NTRU without ciphertext validation routine

Daewan Han, Jin Hong, Jae Woo Han, and Daesung Kwon

National Security Research Institute,
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, KOREA
dwh,jinhong,jwhan,ds_kwon@etri.re.kr

Abstract. NTRU is an efficient public-key cryptosystem proposed by Hoffstein, Pipher, and Silverman. Assuming access to a decryption oracle, we show ways to recover the private key of NTRU systems that do not include a ciphertext validating procedure. The strongest of our methods will employ just a single call to the oracle, and in all cases, the number of calls needed will be small enough to be realistic.

1 Introduction

NTRU cryptosystem([2]), introduced during the rump session of Crypto'96, is one of the most efficient public key cryptosystem now available. Because the encryption/decryption process of this system involves polynomials with small coefficients, it is quite fast compared to systems like RSA, ElGamal, or ECC. It also require only a small amount of memory and hence is suitable for constrained environments like smart cards, PDAs, and mobile phones. The NTRU cryptosystem is currently being considered by standards bodies([1, 6]).

In Crypto 2000, E. Jaulmes and A. Joux ([7]) showed that a chosen ciphertext attack on NTRU could succeed probabilistically. Reaction attacks([4]) were also presented on NTRU. To counter-measure these attacks, various padding schemes were proposed([3, 5]). Later, P. Nguyen and D. Pointcheval([8]) showed that there were weakness in the padding schemes proposed by the NTRU company and suggested a method which is claimed to be IND-CCA.

In this paper, we show how to obtain the private key of the NTRU cryptosystem, assuming access to an Unconditionally Decrypting Oracle. We define a UDO to be an oracle that returns a decryption of any given ciphertext without checking its validity as a ciphertext. Access to a UDO could be realistic if, for some reason, the ciphertext validating procedure was incorrectly implemented. For example, when implementing NTRU-REACT ([8]), unlike other padding schemes, it is easy to leave out the ciphertext validation procedure. This situation could also be possible if the implementer just left it out with ill intentions.

Under the UDO assumption, we show ways to *deterministically* recover the private key of NTRU cryptosystem. The first of our methods applies to the original NTRU system as given in [2]. Let N be the NTRU parameter defining the working ring size. Using $\mathcal{O}(N^2)$ calls to the UDO, we can always recover the private key. We can also recover the private key, with probability of failure less than $1/2^{90}$, using less than $2N$ calls to the UDO.

We show three more methods that apply to *optimized* NTRU ([1, 3, 6]). To the best of our knowledge, the present work is the first in acknowledging attacks specific to the optimized version of NTRU. All three of the methods will recover the private key completely. Of the three, the last one we shall present(Section 6) is the strongest, using just a *single* UDO call. Thus, the NTRU cryptosystem should never be used without a proper padding scheme.

If the use of a UDO were possible on some (flawed) implementation of RSA-OAEP or the Cramer-Shoup scheme, we would no longer be sure of their $\mathsf{IND\text{-}CCA2}$ property. But contrary to the NTRU situation presented by this work, nothing is known in the direction of key recovery attacks on the RSA or ElGamal primitive. This suggests that we should still be very careful in the use of NTRU cryptosystem.

2 Overview of the NTRU cryptosystem

The NTRU cryptosystem has gone through some changes since its first appearance. To set the grounds of our discussion, we quickly present various parameters and encryption/decryption processes for three versions of it. The readers may refer to [2] and [1, 3] for more information. The reference [10] is also helpful in understanding why the decryption process works.

Let N be an odd prime. We will be working over the ring $\mathcal{R} = \mathbf{Z}[x]/(x^N - 1)$. The ring $\mathcal{R}$ is identified with the set of integer polynomials of degree less than N. Multiplication in $\mathcal{R}$ is denoted by $*$.

The sets $\mathcal{L}_\mathbf{f}$, $\mathcal{L}_\mathbf{g}$, $\mathcal{L}_\mathbf{r}$, and $\mathcal{L}_\mathbf{m}$, to be fixed below for each version, are subsets of $\mathcal{R}$. Two parameters $\mathbf{p}$ and q are chosen so that they are relatively prime and the private key $\mathbf{f} \in \mathcal{L}_\mathbf{f}$ is taken so that it is invertible modulo q. The inverse will be denoted by $\mathbf{f}_q$, and using a random polynomial $\mathbf{g} \in \mathcal{L}_\mathbf{g}$ the public key is set to

$$\mathbf{h} \equiv \mathbf{p} * \mathbf{f}_q * \mathbf{g} \pmod{q}. \tag{1}$$

To encrypt a message $\mathbf{m} \in \mathcal{L}_\mathbf{m}$, we choose a random $\mathbf{r} \in \mathcal{L}_\mathbf{r}$ and compute

$$\mathbf{e} = \mathbf{r} * \mathbf{h} + \mathbf{m}. \tag{2}$$

The decryption process is more involved and explained below for each version of NTRU. Let us fix one notation before doing this. Given $m, n \in \mathbf{Z}$ and $t \in \mathbf{R}$, define $\mathrm{Mod}_n^t(m)$ to be the unique integer in the interval $(t - \frac{n}{2}, t + \frac{n}{2}]$ congruent to m modulo n. For a polynomial $f(x) \in \mathcal{R}$, we may similarly define $\mathrm{Mod}_n^t(f(x))$, by applying it to the coefficients.

2.1 Original version

In the original version [2], the parameter $\mathbf{p}$ is fixed to be 3. When choosing a private key $\mathbf{f}$, it is also required to have a modulo $\mathbf{p} = 3$ inverse $\mathbf{f}_p$. We define

$\mathcal{L}(d_+, d_-)$ to be the set of polynomials in $\mathcal{R}$ with d_+ coefficients equal to 1, d_- coefficients equal to -1, and all other coefficients equal to 0. Table 1 lists the various parameters as originally given in [2]. The message space $\mathcal{L}_\mathbf{m}$ is set to all

	N	q	$\mathcal{L}_\mathbf{f}$	$\mathcal{L}_\mathbf{g}$	$\mathcal{L}_\mathbf{r}$
Case A	107	64	$\mathcal{L}(15, 14)$	$\mathcal{L}(12, 12)$	$\mathcal{L}(5, 5)$
Case B	167	128	$\mathcal{L}(61, 60)$	$\mathcal{L}(20, 20)$	$\mathcal{L}(18, 18)$
Case C	503	256	$\mathcal{L}(216, 215)$	$\mathcal{L}(72, 72)$	$\mathcal{L}(55, 55)$

Table 1. Parameters for original NTRU ($\mathbf{p} = 3$)

polynomials in $\mathcal{R}$ with coefficients in $\{-1, 0, 1\}$.

To decrypt a ciphertext $\mathbf{e}$, we go through the following process.

1. $\mathbf{a} \leftarrow \mathrm{Mod}_q^0(\mathbf{f} * \mathbf{e})$.
2. Return $\mathrm{Mod}_\mathbf{p}^0(\mathbf{f}_p * \mathbf{a})$.

We say that a *wrapping* has occurred during the decryption process if

$$\mathbf{a} \neq \mathbf{f} * \mathbf{e},$$

i.e., the Mod_q^0 operation has changed at least one coefficient. Notice that even though $\mathbf{e}$ is defined only up to modulo q, when we use $\mathbf{e}$ as an input to the decryption machine, an explicit representative of $\mathbf{e}$ will be used. Hence this equation makes sense as an equation in $\mathcal{R}$.

2.2 Binary-F version

Set $\mathbf{p} = x + 2$. The private key space $\mathcal{L}_\mathbf{f}$ is taken to be polynomials of the form

$$\mathbf{f} = 1 + \mathbf{p} * \mathbf{F}, \tag{3}$$

with $\mathbf{F}$ a binary polynomial having d_F-many nonzero coefficients.

The sets $\mathcal{L}_\mathbf{g}$ and $\mathcal{L}_\mathbf{r}$ are to contain binary polynomials with d_g and d_r coefficients equal to 1, respectively. The message space $\mathcal{L}_\mathbf{m}$ is the set of all binary polynomials in $\mathcal{R}$. Table 2 lists parameters as given in [1].

Given a ciphertext $\mathbf{e}$, the decryption process goes through the following steps.

1. $I \leftarrow \mathrm{Mod}_q^{\frac{N}{2}}(\mathbf{e}(1) - \mathbf{r}(1) \cdot \mathbf{h}(1))$.
2. $\mathcal{A} \leftarrow \frac{1}{N}(\mathbf{p}(1) \cdot \mathbf{r}(1) \cdot \mathbf{g}(1) + I \cdot \mathbf{f}(1))$.
3. $\mathbf{a} \leftarrow \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{f} * \mathbf{e})$.
4. Return $\mathrm{Mod}_\mathbf{p}(\mathbf{a})$.

Here, the $\mathrm{Mod}_\mathbf{p}$ operation chooses a representative of $\mathbf{a}$ in a binary polynomial form. We refer the readers to [3] for more detail.

As before, we say that a *wrapping* has occurred during decryption if $\mathbf{a} \neq \mathbf{f} * \mathbf{e}$.

2.3 Low Hamming weight-F version

This version of NTRU is identical to the previous binary-**F** version except that the polynomial **F** used in defining the private key **f** is of the form

$$\mathbf{F} = \mathbf{F}_1 * \mathbf{F}_2 + \mathbf{F}_3, \tag{4}$$

with each $\mathbf{F}_i$ a binary polynomial having d_{F_i}-many nonzero coefficients. The parameter values are given in Table 2. Notice that $d_F = d_{F_1} \cdot d_{F_2} + d_{F_3}$ in all cases. We shall call this the LHW-**F** case.

N	q	d_F	d_{F_1}	d_{F_2}	d_{F_3}	d_g	d_r
251	128	72	8	8	8	72	72
347	128	64	7	8	8	173	64
503	256	420	20	20	20	251	170

Table 2. Parameters for NTRU with $\mathbf{p} = x + 2$

When we want to refer to both the binary-**F** and LHW-**F** versions of NTRU at the same time, we shall use the phrase *optimized* NTRU.

Remark 1. Since $\mathbf{r}(1) = d_r$, $\mathbf{g}(1) = d_g$, and $\mathbf{f}(1) = 1 + 3 \cdot d_F$, the values I and $\mathcal{A}$ may be calculated from just **e** and public values.

Remark 2. The value I is calculated as

$$I = \mathrm{Mod}_q^{\frac{N}{2}} \left(\mathbf{f}_q(1) \cdot (\mathbf{a}(1) - \mathbf{p}(1) \cdot \mathbf{r}(1) \cdot \mathbf{g}(1))\right)$$

in [1, 6, 10]. This may seem different from what is done in step 1. Furthermore, in this form, knowledge of the secrete information $\mathbf{f}_q$ also seems to be required. But both are calculating the same value $\mathbf{m}(1)$, assuming that it is close to $\frac{N}{2}$.

3 Adaptive use of wrapping behavior

We shall work with the original NTRU setting (Section 2.1) in this section. While the arguments of this section may be modified and applied to other settings, it is better explained in the original setting and does not involve the use of special forms for the private key **f** that appears in other settings.

We show how to exploit the *wrapping* behavior of the modulo q reduction process done during decryption to recover the private key, using less than $2N$ calls to the decryption oracle.

3.1 Descriptive argument

We assume that we have access to a decryption oracle and may distinguish whether or not a wrapping has occurred during the decryption process. For example, this could be possible, in some situations, through timing techniques. This could also be possible if we may obtain the decrypted message, since the decrypted output of ciphertext $\mathbf{e}$ is equal to $\mathrm{Mod}^0_{\mathbf{p}}(\mathbf{e})$ if and only if no wrapping has occurred.

We first remark that any cyclic shift $\mathbf{f} * x^i$ of the private key $\mathbf{f}$ may work as a decryption key. The cyclic shift added to $\mathbf{a}$ during the first step of decryption is removed when $\mathbf{f}_p$ is applied in the second step.

Consider what would happen if we ran the ciphertext

$$\mathbf{e} = -\big(1 + x + x^2 - x^3 + (\frac{q}{2} - 4)x^4\big) \tag{5}$$

through the decryption oracle. Recall that the coefficients of $\mathbf{f}$ all belong to the set $\{-1, 0, 1\}$. Hence wrapping occur if and only if there exists some consecutive run of coefficients in $\mathbf{f}$ equal to the sequence $(1, 1, 1, -1, 1)$, in reverse order.

Suppose we do have wrapping for the above particular $\mathbf{e}$. We may then run

$$\mathbf{e}\pm = -\big(1 + x + x^2 - x^3 + x^4 \pm (\frac{q}{2} - 5)x^5\big) \tag{6}$$

through the decryption oracle. If either one brings a wrapping behavior, we may continue, knowing a longer consecutive sequence of coefficients from $\mathbf{f}$. If neither returns a wrapping behavior, we know neither $(1, 1, 1, -1, 1, 1)$ nor $(1, 1, 1, -1, 1, -1)$ is a consecutive run of coefficients from $\mathbf{f}$ (in reverse order). But we know the shorter subsequence $(1, 1, 1, -1, 1)$ is present in $\mathbf{f}$, so the sequence $(1, 1, 1, -1, 1, 0)$ must be present in $\mathbf{f}$. We may then use

$$\mathbf{e}0\pm = -\big(1 + x + x^2 - x^3 + x^4 \pm (\frac{q}{2} - 5)x^6\big)$$

to find the next coefficient of $\mathbf{f}$.

We can continue with this process until we know a consecutive sequence of coefficients from $\mathbf{f}$ that contains $\frac{q}{2}$-many ± 1's. So, for Case A of Table 1, this process obtains some cyclic shift of $\mathbf{f}$ completely.

To cope with the other two cases given in Table 1, let us go back and suppose that $\mathbf{e}$ given by (5) induces a wrapping so that $(1, 1, 1, -1, 1)$ appears as a consecutive run of coefficients in $\mathbf{f}$. Let us also assume that $\frac{q}{2} = 5$. Then we cannot use $\mathbf{e}\pm$ given by (6) since they are both just equal to $\mathbf{e}$. But, if we can assume that the probability of the subsequence $(1, 1, 1, -1)$ appearing in $\mathbf{f}$ *more than once* is negligible, we can use

$$\mathbf{e}'\pm = -(1 + x + x^2 - x^3 \pm x^5) \tag{7}$$

to find the next coefficient. The probability of some sequence of length n appearing in $\mathbf{f}$ more than once is at most $\frac{N}{3^n}$. So, in Case B, this will be at most

$167/3^{63} \sim 1/2^{92.47}$ and in Case C, this is at most $503/3^{127} \sim 1/2^{192.32}$, indeed, values that may be ignored in real world applications.

Hence, we can obtain a cyclic shift of $\mathbf{f}$, which in turn can be used later to find the plain text $\mathbf{m}$ corresponding to any given ciphertext $\mathbf{e}$.

We have included a complete example of this approach in Appendix A.

3.2 Algorithm

We summarize arguments of the previous subsection in an algorithm.

Algorithm 1

1. *Initialize secret key* $\mathbf{f} = \sum_{i=0}^{N-1} f_i x^i$, *input ciphertext* $\mathbf{e} = \sum_{i=0}^{N-1} e_i x^i$, *and integer* w *as follows.*
   (a) $f_0 = 1$, $f_j = 0$ *for* $1 \leq j \leq N-1$
   (b) e_j *= 0 for* $0 \leq j \leq N-1$
   (c) w *= 1*
2. *For* i *from 1 to* $N-1$ *do the following:*
   (a) *If* $w < q/2$, *then set* $e_{i-1} = -f_{i-1}$, *otherwise set* $e_{i-1} = 0$.
   (b) *If* $w < q/2$, *then set* $e_i = -(\frac{q}{2} - w)$, *otherwise set* $e_i = -1$.
   (c) *Run* $\mathbf{e}$ *through the decryption oracle.*
   (d) *If wrapping has occurred, set* $f_i = 1$, $w = w + 1$, *and skip (e)∼(h).*
   (e) *Set* $e_i = -e_i$.
   (f) *Run* $\mathbf{e}$ *through the decryption oracle.*
   (g) *If wrapping has occurred, set* $f_i = -1$, $w = w + 1$, *and skip (h).*
   (h) *Set* $f_i = 0$.
3. *Reverse the order of* f_i. *That is, set* $f_i = f_{N-1-i}$ *for* $i = 0, 1, \cdots, N-1$.
4. *Return* $\mathbf{f} = \sum_{i=0}^{N-1} f_i x^i$.

This algorithm obtains some cyclic shift of $\mathbf{f}$ using $2N - d_+ - 1$ calls to the decryption oracle. Probability of this algorithm returning a wrong value of $\mathbf{f}$ is at most $N/3^{\frac{q}{2}-1}$.

Remark 3. Using ideas present in this algorithm, it is possible construct an algorithm that returns a cyclic shift of $\mathbf{f}$ deterministically without failure. But the algorithm would use $\mathcal{O}(N^2)$ calls to the decryption oracle and become much more complex.

4 Pre-designed wrapping (non-adaptive use)

This section will deal with recovering the private key $\mathbf{f}$ when it is given in the form (3). So it covers both of the *optimized* NTRU versions(Sections 2.2 and 2.3).

We shall assume access to decrypted output of ciphertexts of our choice. The ciphertexts to be inserted in the machine will be determined from the public key through pre-computation and does not depend on previous output of the decryption oracle. The number of queries to the decryption machine needed is less than twice the size of the coefficient set for $\mathbf{f}$. If the coefficient set for $\mathbf{f}$ is small, the number of queries needed could be much smaller. We shall recover $\mathbf{f}$ completely.

4.1 The method

We shall use a concrete example in explaining a method for recovering the private key. Take the value $N = 251$ and $q = 128$ from Table 2 and assume that the coefficient set of $\mathbf{f}$ is $\{0, 1, 2\}$. Application of this method to situations when the coefficient set is bigger will be straightforward. We propose to run the constant polynomial $\mathbf{e} = e$ (for some $0 \leq e < q$) through the decryption machine. As stated by Remark 1, given $\mathbf{e}$ and a specific public key $\mathbf{h}$, anybody may find the value $\mathcal{A}$. Let us fix a public key $\mathbf{h}$ and write $\mathcal{A}(e)$ to denote the value $\mathcal{A}$ corresponding to the constant polynomial $\mathbf{e} = e$.

For the parameter values given in the $N = 251$ row of Table 2 we can calculate

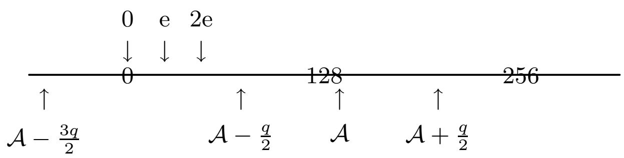
$$\begin{aligned}
&\mathbf{f}(1) = 1 + (1+2)\cdot \mathbf{F}(1) = 1 + 3 \cdot d_F = 217, \\
&\mathbf{f}_q(1) \equiv \mathbf{f}(1)^{-1} \equiv 105 \pmod{q}, \\
&\mathbf{r}(1) \cdot \mathbf{h}(1) \equiv d_r \cdot 3 \cdot \mathbf{f}_q(1) \cdot d_g \equiv 64 \pmod{q}.
\end{aligned}$$

With this, we can find the I and $\mathcal{A}$ values for various $\mathbf{e} = e$. Gather terms of the private key $\mathbf{f}$ according to their coefficients and write

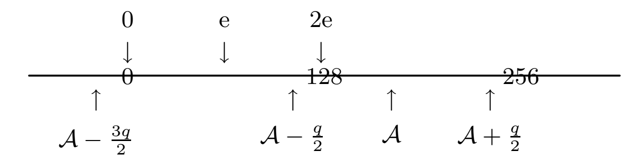
$$\mathbf{f} = 0 \cdot \mathbf{f}_0 + 1 \cdot \mathbf{f}_1 + 2 \cdot \mathbf{f}_2. \tag{8}$$

The coefficients of $\mathbf{f} * \mathbf{e}$ will belong to the set $\{0, e, 2e\}$. Below, we have drawn their position relative to $\mathcal{A}(e)$ for some chosen e values.

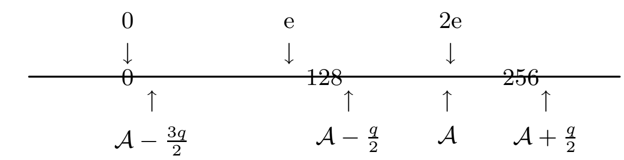
- $e = 24$, $\mathcal{A}(e) \risingdotseq 138.04$



- $e = 63$, $\mathcal{A}(e) \risingdotseq 171.76$



- $e = 105$, $\mathcal{A}(e) \risingdotseq 208.07$



Let us fix $e = 63$ and follow through the decryption steps with the help of the above drawing.

$$\begin{aligned}\mathbf{a}(e) &= \mathrm{Mod}_q^{\mathcal{A}(e)}(\mathbf{f} * \mathbf{e}) \\ &= \mathrm{Mod}_q^{\mathcal{A}(e)}(0 \cdot \mathbf{f}_0 + e \cdot \mathbf{f}_1 + 2e \cdot \mathbf{f}_2) \\ &= (0 + q) \cdot \mathbf{f}_0 + (e + q) \cdot \mathbf{f}_1 + 2e \cdot \mathbf{f}_2 \\ &= e \cdot \mathbf{f} + q \cdot (\mathbf{f}_0 + \mathbf{f}_1).\end{aligned}$$

Let us do this once more with $e' = 105$.

$$\begin{aligned}\mathbf{a}(e') &= \mathrm{Mod}_q^{\mathcal{A}(e')}(\mathbf{f} * \mathbf{e}') \\ &= \mathrm{Mod}_q^{\mathcal{A}(e')}(0 \cdot \mathbf{f}_0 + e' \cdot \mathbf{f}_1 + 2e' \cdot \mathbf{f}_2) \\ &= (0 + 2q) \cdot \mathbf{f}_0 + (e' + q) \cdot \mathbf{f}_1 + 2e' \cdot \mathbf{f}_2 \\ &= e' \cdot \mathbf{f} + q \cdot (2\mathbf{f}_0 + \mathbf{f}_1).\end{aligned}$$

We may observe that the difference of the outputs from the decryption machine satisfies

$$\begin{aligned}D(e, e') &:= \mathrm{Mod}_{\mathbf{p}}(\mathbf{a}(e)) - \mathrm{Mod}_{\mathbf{p}}(\mathbf{a}(e')) \\ &\equiv \mathbf{a}(e) - \mathbf{a}(e') \pmod{\mathbf{p}} \\ &= (e - e') \cdot \mathbf{f} - q \cdot \mathbf{f}_0 \\ &= (e - e') \cdot (1 + \mathbf{p} \cdot \mathbf{F}) - q \cdot \mathbf{f}_0 \\ &\equiv (e - e') - q \cdot \mathbf{f}_0 \pmod{\mathbf{p}}.\end{aligned}$$

Denote the modulo $\mathbf{p}$ inverse of q by $q_{\mathbf{p}}$ so that $q_{\mathbf{p}} \cdot q \equiv 1 \pmod{\mathbf{p}}$. We may now obtain

$$\mathrm{Mod}_{\mathbf{p}}\big(- q_{\mathbf{p}} \cdot \big(D(e, e') - (e - e')\big)\big) = \mathrm{Mod}_{\mathbf{p}}\big(q_{\mathbf{p}} \cdot q \cdot \mathbf{f}_0\big) = \mathrm{Mod}_{\mathbf{p}}(\mathbf{f}_0) = \mathbf{f}_0.$$

We stress that all three equalities above are true equalities in the ring $\mathcal{R}$. They are stronger than just modulo $\mathbf{p}$ equivalence relations. The last equality follows since $\mathbf{f}_0$ is a binary polynomial. We have found all terms of $\mathbf{f}$ having coefficients equal to 0 with just two queries to the decryption machine.

The above argument obtained $\mathbf{f}_0$ because the value 0 crossed over the value $\mathcal{A} - \frac{3q}{2}$ as we changed e to e' and since neither e nor $2e$ went over any $\mathrm{Mod}_q^{\mathcal{A}}$ operation boundary. If we work with $e = 24$ and $e' = 63$, we can similarly obtain $\mathbf{f}_2$. The remaining terms will now have coefficient equal to 1 and we have found the private key $\mathbf{f}$ with just three queries to the decryption machine.

Remark 4. In the above calculations, we've used the fact $\mathbf{f} \equiv 1 \pmod{\mathbf{p}}$. Hence this attack is not applicable to the original NTRU cryptosystem.

4.2 Feasibility of this approach

We shall use the notation

$$\mathrm{dc}(n, \mathcal{A}) = \frac{1}{q}(n - \mathrm{Mod}_q^{\mathcal{A}}(n)) \tag{9}$$

for any integer n and centering value $\mathcal{A}$. An equivalent definition would be

$$\mathrm{Mod}_q^{\mathcal{A}}(n) = n - q \cdot \mathrm{dc}(n, \mathcal{A}).$$

It measures how far n is from the *representative interval*. When the value $\mathcal{A}$ is clear from context, $\mathrm{dc}(n)$ will be used.

Let the coefficient set of $\mathbf{f}$ be $\{0, 1, \ldots, t\}$. To argue that the previous section is a meaningful attack on un-padded version of optimized NTRU, it remains to consider how likely it is to find an (e, e') pair with $\mathrm{dc}(i \cdot e, \mathcal{A}(i \cdot e))$ and $\mathrm{dc}(i \cdot e', \mathcal{A}(i \cdot e'))$ differing at just one $0 \le i \le t$. We have no proof that enough such pairs may always be found, but will give an informal argument showing that this is highly possible. Also, a complete solution for the $N = 251$ case is provided in the Appendices as an example.

Examine what happens to the dc-values when we increase e by just 1. If we study the procedure for calculating $\mathcal{A}(e)$, we find that setting $e \leftarrow e + 1$ increases $\mathcal{A}(e)$, in most cases (exception occurs when I goes over the modulo q boundary), by $\frac{\mathbf{f}(1)}{N}$. For the parameter values given in Table 2, this value is roughly 0.86 ($N = 251$), 0.56 ($N = 347$), and 2.51 ($N = 503$) for each case. Of course, setting $e \leftarrow e + 1$ increase $i \cdot e$ by the amount i. We want to point out that not all $t + 1$ of these values i can be close to $\frac{\mathbf{f}(1)}{N}$ at the same time.

The second point we want to make is that, as we change e from 0 to $q - 1$, the values $i \cdot e$ all start as being equal to 0 and end up spreading out over an interval of length $t \cdot q$. So the $t + 1$ dc-values will start out as the same and end up as different.

These two points convince us that not all of the dc-values can stay constant over the change of e from 0 to $q - 1$.

Finally, we want to call to attention one more point. Since the distance between $i \cdot e$ and $(i + 1) \cdot e$ is less than q, if e is big enough, it is almost impossible (again, the same exception apply) for two adjacent dc-values to change simultaneously, as we increase e by 1. Non-adjacent dc-values have a better chance of changing simultaneously, but since they have to be apart by a multiple of q for this to happen, this is not too frequent.

If this does not convince the reader, we can just roughly say that we have about q equations in hand to solve for $t + 1$ variables.

For the case $N = 251$, we have given a table of dc-values in Appendix B. As we have already seen, for the parameter values given in the first row of Table 2, we have $\mathbf{r}(1)\mathbf{h}(1) = 64$. Notice that the maximum possible value for the coefficient of $\mathbf{f}$, in either the binary-$\mathbf{F}$ or the LHW-$\mathbf{F}$ case, is $28 = 3 \cdot (8 + 1) + 1$. So, for each $e = 0, 1, \ldots, q - 1$, we've listed the values $\mathrm{dc}(i \cdot e, \mathcal{A}(i \cdot e))$ for $i = 0, 1, \ldots, 28$.

In Appendix C, we used these values to give an explicit instruction for determining the private key $\mathbf{f}$ completely with 52 queries to the decryption machine. In practice, we do not expect $\mathbf{f}$ to contain coefficients as large as 28. So the process would be a lot shorter.

If the coefficient set of $\mathbf{f}$ is just $\{0, 1, 2, 3\}$, which is highly probable in the binary-$\mathbf{F}$ case, Appendix D explains how one could obtain $\mathbf{f}$ completely with just one or two queries to the decryption machine.

5 Bypassing wrapping (using $\mathbf{p}_q$)

In this section, we assume the private key is given by a binary-$\mathbf{F}$(Section 2.2). We present a chosen-ciphertext attack which makes one query to the decryption machine and recovers the private key completely.

Let us denote the modulo q inverse of $\mathbf{p}$ by $\mathbf{p}_q$, so that

$$\mathbf{p}_q * \mathbf{p} \equiv 1 \pmod{q}.$$

If $q = 2^k$, we may specifically set

$$\mathbf{p}_q = \sum_{i=1}^{k} (-2)^{i-1} x^{N-i} \pmod{q}. \tag{10}$$

5.1 Simple case

If we insert $\mathbf{p}_q$ into the decryption machine, it will calculate

$$\begin{aligned}\mathbf{a} &= \mathrm{Mod}_q^{\mathcal{A}(\mathbf{p}_q)}(\mathbf{f} * \mathbf{p}_q)\\ &= \mathrm{Mod}_q^{\mathcal{A}}\big((1 + \mathbf{p} * \mathbf{F}) * \mathbf{p}_q\big)\\ &= \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q + \mathbf{F}).\end{aligned}$$

Since all the coefficients of $\mathbf{F}$ are either 0 or 1, with high probability, we will have

$$\mathbf{a} = \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q) + \mathbf{F}. \tag{11}$$

Assume for the moment that this is true. Then, we have

$$\mathrm{Mod}_{\mathbf{p}}(\mathbf{a}) - \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q) \equiv \mathbf{a} - \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q) \equiv \mathbf{F} \pmod{\mathbf{p}}.$$

Notice that the first term on the left is the output of the decryption machine, and that the second term on the left may readily be computed. Hence we may obtain

$$\mathrm{Mod}_{\mathbf{p}}\big(\mathrm{Mod}_{\mathbf{p}}(\mathbf{a}) - \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q)\big) = \mathrm{Mod}_{\mathbf{p}}(\mathbf{F}) = \mathbf{F}.$$

The second equality holds, since $\mathbf{F}$ is a binary polynomial. We have obtained the private key $\mathbf{f} = 1 + \mathbf{p} * \mathbf{F}$ with just one query.

Remark 5. This attack obviously relies on the form of the private key $\mathbf{f} = 1 + \mathbf{p} * \mathbf{F}$. Hence this attack may not be applied to the original NTRU scheme.

It remains to justify equation (11). For parameters given in Table 2, we have calculated various values.

N	$\mathbf{f}(1)$	$\mathbf{f}_q(1)$	$\mathbf{r}(1)\mathbf{h}(1)$	$\mathbf{p}_q(1)$	$I(\mathbf{p}_q)$	$\mathcal{A}(\mathbf{p}_q)$
251	217	105	64	43	107	154.47
347	193	65	64	43	235	226.43
503	1261	229	242	171	185	718.28

Some of these values are defined only up to modulo q. Now, using equation (10) and this table, we list all coefficients (including the one corresponding to zero) of $\mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q)$ in the following table. We've also written down the lower and upper boundaries (LB,UB) of the *representative interval*. Last column contains the distance between UB and the coefficient maximum.

N	LB	coefficients	UB	headroom
251	91	129, 126, 132, 120, 144, 96, 192, 128	218	26
347	163	257, 254, 260, 248, 272, 224, 192, 256	290	18
503	591	769, 766, 772, 760, 784, 736, 832, 640, 768	846	14

So at least, for the parameter values given in Table 2, equation (11) is always satisfied.

5.2 Wrapping case

Assumption of the previous subsection, namely, equation (11), fails if and only if

1. some coefficient c of $\mathrm{Mod}_q^{\mathcal{A}(\mathbf{p}_q)}(\mathbf{p}_q)$ satisfies $c \le \mathcal{A}(\mathbf{p}_q) + \frac{q}{2} < c + 1$,
2. and the corresponding coefficient of $\mathbf{F}$ is equal to 1.

Since we know the exact polynomial $\mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q)$, we know which coefficients satisfy the first of the above conditions. Suppose some coefficient c_i of the x^i term in $\mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q)$ satisfies both conditions. Suppose further, for the moment, that such a coefficient is unique. Then

$$\begin{aligned}\mathbf{a} &= \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q + \mathbf{F}) \\ &= \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q) - qx^i + \mathbf{F}.\end{aligned}$$

And the output of the decryption machine satisfies

$$\mathrm{Mod}_{\mathbf{p}}(\mathbf{a}) \equiv \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q) - qx^i + \mathbf{F} \pmod{\mathbf{p}}.$$

As before, we may obtain the private key by computing

$$\mathrm{Mod}_{\mathbf{p}}\left(\mathrm{Mod}_{\mathbf{p}}(\mathbf{a}) - \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{p}_q) + qx^i\right) = \mathrm{Mod}_{\mathbf{p}}(\mathbf{F}) = \mathbf{F}.$$

In conclusion, if $\mathbf{p}_q$ contains t-many coefficients satisfying the above condition 1, with just one query to the decryption machine, we may find 2^t candidates for $\mathbf{F}$, one of which corresponds to the true private key $\mathbf{f} = 1 + \mathbf{p} * \mathbf{F}$.

Remark 6. If $q = 2^k$, we know from equation (10) that all of the coefficients of $\mathbf{p}_q$ are distinct modulo q. (Read next remark to see why this isn't strictly true.) Hence there can be at most one coefficient satisfying the first of the above two conditions.

Remark 7. In the $q = 2^k$ case, if it happens that some coefficient $c \equiv 0 \pmod q$ satisfies the first condition, application of this method is not feasible. But with some modifications we could use $-\mathbf{p}_q$ or even $2\mathbf{p}_q$ in a similar attack.

6 Uniform wrapping (using the public key $\mathbf{h}$)

This section contains the simplest, and perhaps, the strongest of our attacks on un-padded NTRU. Using just one query to the decryption machine, we shall obtain completely, the binary polynomial $\mathbf{g}$ used in defining the public key, with probability $(q-1)/q$. Since the public key is given by $\mathbf{h} = \mathbf{p} * \mathbf{f}_q * \mathbf{g}$, this is (almost) equivalent to having obtained the private key $\mathbf{f}$.

As before, let $\mathbf{p}_q$ be the modulo q inverse of $\mathbf{p}$. We run $\mathbf{e} = \mathbf{p}_q * \mathbf{h}$ through the decryption machine. The output of the machine will be

$$\begin{aligned}\mathrm{Mod}_{\mathbf{p}}\,\mathrm{Mod}_q^{\mathcal{A}}(\mathbf{f} * \mathbf{e}) &= \mathrm{Mod}_{\mathbf{p}}\,\mathrm{Mod}_q^{\mathcal{A}}(\mathbf{f} * \mathbf{p}_q * \mathbf{p} * \mathbf{f}_q * \mathbf{g})\\ &= \mathrm{Mod}_{\mathbf{p}}\,\mathrm{Mod}_q^{\mathcal{A}}(\mathbf{g}).\end{aligned}$$

Recall the notation (9). With probability $(q-1)/q$, we can expect to have $\mathrm{dc}(0) = \mathrm{dc}(1)$. For the parameter values of Table 2, we may easily check that they are equal.

N	$I(\mathbf{p}_q * \mathbf{h})$	$\mathcal{A}(\mathbf{p}_q * \mathbf{h})$	LB	UB	dc(0)	dc(1)
251	72	124.21	61	188	-1	-1
347	173	191.95	128	255	-1	-1
503	149	628.03	501	756	-2	-2

Assume $\mathrm{dc}(0) = \mathrm{dc}(1)$ and let d denote this common value. Set

$$S(x) = 1 + x + \cdots + x^{N-1}.$$

Then, we may write

$$\mathrm{Mod}_q^{\mathcal{A}}(\mathbf{g}) = \mathbf{g} - d \cdot q \cdot S(x).$$

Hence

$$\mathrm{Mod}_{\mathbf{p}}\left(\mathrm{Mod}_q^{\mathcal{A}}(\mathbf{f} * \mathbf{e})\right) + d \cdot q \cdot S(x) \equiv \mathrm{Mod}_q^{\mathcal{A}}(\mathbf{g}) + d \cdot q \cdot S(x) \equiv \mathbf{g} \pmod{\mathbf{p}}$$

and we may obtain

$$\mathrm{Mod}_{\mathbf{p}}\Big(\mathrm{Mod}_{\mathbf{p}}\left(\mathrm{Mod}_q^{\mathcal{A}}(\mathbf{f} * \mathbf{e})\right) + d \cdot q \cdot S(x)\Big) = \mathbf{g}$$

from just one query to the decryption machine. The value

$$\mathbf{f} * \mathbf{h} \equiv \mathbf{f} * \mathbf{p} * \mathbf{f}_q * \mathbf{g} \equiv \mathbf{p} * \mathbf{g} \pmod{q}$$

is in our hands. Now, if $\mathbf{h}$ is invertible modulo q, or equivalently, if $\mathbf{g}$ is invertible modulo q, we can obtain $\mathbf{f}$ modulo q. We know the form of $\mathbf{f}$, so can find $\mathbf{f}$ exactly. Furthermore, the random binary polynomial $\mathbf{g}$ is invertible with a very high probability. Even if it is not, we still have the possibility of using a pseudo inverse of $\mathbf{h}$ to obtain $\mathbf{f}$.

Remark 8. Arguments of this section cannot be applied to the original NTRU scheme. Fundamentally, this approach is only possible because in the optimized version of NTRU, we have $\mathbf{f_p} = 1$ and hence the process of multiplying $\mathbf{f_p}$ to $\mathbf{a}$, present in the original scheme, is no longer carried out. Simplifying the decryption process has opened up a new weakness in NTRU.

Remark 9. In the case $\mathrm{dc}(0) \neq \mathrm{dc}(1)$, we may use $-\mathbf{p}_q * \mathbf{h}$ in a similar attack. Again, we have about $1/q$ chance of encountering the same problem.

Remark 10. Suppose $\mathrm{dc}(0) \neq \mathrm{dc}(1)$, or equivalently, $\mathrm{dc}(0) + 1 = \mathrm{dc}(1)$. We may write $\mathbf{g} = 0 \cdot \mathbf{g}_0 + 1 \cdot \mathbf{g}_1$ with $S(x) = \mathbf{g}_0 + \mathbf{g}_1$. Then

$$\begin{aligned}
&\mathrm{Mod}_{\mathbf{p}}\big(\mathrm{Mod}_q^{\mathcal{A}}(\mathbf{g})\big) + q \cdot \mathrm{dc}(0) \cdot S(x) \\
&\qquad \equiv \mathbf{g} - q \cdot \mathrm{dc}(0) \cdot \mathbf{g}_0 - q \cdot \mathrm{dc}(1) \cdot \mathbf{g}_1 + q \cdot \mathrm{dc}(0) \cdot (\mathbf{g}_0 + \mathbf{g}_1) \pmod{\mathbf{p}} \\
&\qquad = 0 \cdot \mathbf{g}_0 + (1 - q) \cdot \mathbf{g}_1 \\
&\qquad = (1 - q) \cdot (0 \cdot \mathbf{g}_0 + 1 \cdot \mathbf{g}_1) = (1 - q) \cdot \mathbf{g}.
\end{aligned}$$

Hence, if $1 - q$ is invertible modulo $\mathbf{p}$ in $\mathcal{R}$, we may find $\mathbf{g}$. This is true for the values $N = 251$ with $q = 128$ and $N = 347$ with $q = 128$.

7 Conclusion

We've seen four chosen-ciphertext attacks applicable to un-padded versions of NTRU cryptosystem.

The first of these, given in Section 3, is an improvement to previous reaction attacks. It applies to the most general NTRU cryptosystem and uses less than $2N$ queries to the decryption oracle.

The approaches of the next three sections apply to optimized versions of NTRU. In Section 4, we pre-compute wrappings to be expected in the decryption process and use differences of these wrappings to recover the private key. A very small number of queries to the decryption oracle was needed. Other two approaches presented here, given in Section 5 and Section 6, uses just one query to the decryption machine to recover the private key completely, under realistic parameter values. These three methods depend on the private key being of the form $\mathbf{f} = 1 + \mathbf{p} * \mathbf{F}$. None of these three attacks on optimized NTRU are applicable to the original NTRU cryptosystem. By giving special forms for the private key, the key generation and decryption processes of NTRU cryptosystem became simpler. But at the same time, it has opened new ways of recovering the private key.

The number of decryption oracle use needed in approaches of this paper is small enough to be realistic. Hence NTRU should never be used without some form of padding, protecting it from chosen ciphertext attacks.

However, we believe any reasonable padding scheme will provide the optimized NTRU cryptosystem protection from our attacks. Of course, with explicit hash functions chosen to be used in the padding schemes, the story could be different. This part still remains to be considered.

References

1. Consortium for Efficient Embedded Security, Efficient embedded security standards #1: Implementation aspects of NTRUEncrypt and NTRUSign. Draft version 5. Available from `http://www.ceesstandards.org`.
2. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman, NTRU: A ring-based public key cryptosystem. In *Proc. of ANTS III*, LNCS 1423. Springer-Verlag, 1998.
3. Jeffrey Hoffstein and Joseph Silverman, Optimizations for NTRU. In *Public-Key Cryptogrphy and Computational Number Theory.* DeGruyter, 2002. Available from [9].
4. Jeffrey Hoffstein and Joseph H. Silverman, Reaction attacks against the NTRU public key cryptosystem. Techinal report #015, NTRU Cryptosystems. Available from [9].
5. Jeffrey Hoffstein and Joseph H. Silverman, Protecting NTRU Against Chosen Ciphertext and Reaction Attacks, Technical Report #016, NTRU Cryptosystems. Available from [9].
6. IEEE Standard P1363.1/D4, Standard specifications for public key cryptography : Techniques based on hard problems over lattices, IEEE. Available from `http://grouper.ieee.org/group/1363`.
7. Éliane Jaulmes and Antoine Joux, A chosen-ciphertext attack against NTRU. *Advances in Cryptology - CRYPTO 2000*, LNCS 1880. Springer-Verlag, 2000.
8. Phong Q. Nguyen and David Pointcheval, Analysis and improvements of NTRU encryption paddings. *Advances in Cryptology - CRYPTO 2002*, LNCS 2442. Springer-Verlag, 2002.
9. NTRU Cryptosystems, Technical reports. Available from `http://www.ntru.com`.
10. NTRU Cryptosystems, The NTRU public key cryptosystem - A tutorial. Available from `http://www.ntru.com`.

A Adaptive use of wrapping - Example

We give an example of the approach given in Section 3. Let the parameters and private key be as follows.

$$(N, \mathbf{p}, q) = (11, 3, 32), \quad \mathbf{f} = -1 + x + x^2 - x^4 + x^6 + x^9 - x^{10}.$$

We referred to [10] for the above NTRU parameter. Other parameters like $\mathbf{g}$ or $\mathbf{h}$ will not be needed.

We first try the ciphertext

$$\mathbf{e} = -1 - (\frac{q}{2} - 1)x = -1 - 15x.$$

With this, the intermediate polynomial $\mathbf{a} = \mathbf{f} * \mathbf{e}$ will be

$$a = \langle 16, 14, -16, -15, 1, 15, -1, -15, 0, -1, -14\rangle$$

We have written down just the coefficients of $\mathbf{a}$ for ease of writing. Since $a_2 = -16 \leq \frac{q}{2}$, we will observe a *wrapping*. Thus, we know that, among the coefficients f_i of $\mathbf{f}$, there exist a t such that $(f_t, f_{t-1}) = (1, 1)$. In this case, the real value of t

is 2, but we don't have access to this information. Anyway, we set a polynomial $\mathbf{f}' = \langle 1, 1\rangle$ to be a candidate polynomial for the private key.

Next, we try the ciphertext

$$\mathbf{e} = -1 - x - (\frac{q}{2} - 2)x = \langle -1, -1, -14\rangle$$

Then,

$$\mathbf{a} = \mathbf{f} * \mathbf{e} = \langle -12, 14, 12, -15, -13, 1, 13, -1, -14, -1\rangle.$$

This will not cause a *wrapping*, which implies that there is no t such that $\langle f_t, f_{t-1}, f_{t-2}\rangle$ is equal to $\langle 1, 1, 1\rangle$. We next try the ciphertext

$$\mathbf{e} = -1 - x + (q/2 - 2)x = \langle -1, -1, 14\rangle.$$

Then, we have

$$\mathbf{a} = \langle 16, -14, -16, 13, 15, 1, -15, -1, 14, -1\rangle.$$

This causes a wrapping to occur, which is due to (f_2, f_1, f_0) being equal to $(1, 1, -1)$. Although we don't have access to the exact value of $\mathbf{a}$, we know from the existence of wrapping that there is a t such that $(f_t, f_{t-1}, f_{t-2}) = (1, 1, -1)$. We now set $\mathbf{f}' = \langle 1, 1, -1\rangle$ as a candidate for the private key.

step	testing ciphertext $\mathbf{e}$	wrapping	candidate polynomial
1	$(-1, -15)$	Yes	$(1, 1)$
2	$(-1, -1, -14)$	No	
3	$(-1, -1, 14)$	Yes	$(1, 1, -1)$
4	$(-1, -1, 1, -13)$	No	
5	$(-1, -1, 1, 13)$	Yes	$(1, 1, -1, -1)$
6	$(-1, -1, 1, 1, -12)$	Yes	$(1, 1, -1, -1, 1)$
7	$(-1, -1, 1, 1, -1, -11)$	No	
8	$(-1, -1, 1, 1, -1, 11)$	No	$(1, 1, -1, -1, 1, 0)$
9	$(1, 1, -1, -1, 1, 0, -11)$	No	
10	$(1, 1, -1, -1, 1, 0, 11)$	No	$(1, 1, -1, -1, 1, 0, 0)$
11	$(1, 1, -1, -1, 1, 0, 0, -11)$	Yes	$(1, 1, -1, -1, 1, 0, 0, 1)$
12	$(1, 1, -1, -1, 1, 0, 0, 1, -10)$	No	
13	$(1, 1, -1, -1, 1, 0, 0, 1, 10)$	No	$(1, 1, -1, -1, 1, 0, 0, 1, 0)$
14	$(1, 1, -1, -1, 1, 0, 0, 1, 0, -10)$	No	
15	$(1, 1, -1, -1, 1, 0, 0, 1, 0, 10)$	Yes	$(1, 1, -1, -1, 1, 0, 0, 1, 0, -1)$
16	$(1, 1, -1, -1, 1, 0, 0, 1, 0, -1, -9)$	No	
17	$(1, 1, -1, -1, 1, 0, 0, 1, 0, -1, 9)$	No	$(1, 1, -1, -1, 1, 0, 0, 1, 0, -1, 0)$
	reverse candidate polynomial		$(0, -1, 0, 1, 0, 0, 1, -1, -1, 1, 1)$

Table 3. Process of recovering the private key

In this way, we construct ciphertexts from the candidate polynomial and try them step by step to recover the whole coefficients of the private key. Table 3 shows a summary of this process.

When we reach step 17 of Table 3, we reverse the candidate polynomial $\mathbf{f}'$. That is, we set

$$f'_i = f'_{N-i-1} \quad \text{for } 0 \le i \le N-1.$$

Then, we get the polynomial

$$\mathbf{f} * x^8 = \langle 0, -1, 0, 1, 0, 0, 1, -1, -1, 1, 1 \rangle,$$

which may be used in place of the real $\mathbf{f}$ to decrypt any given ciphertext.

B dc-value list - Example

We have used $N = 251$, $q = 128$, and $d_F = d_g = d_r = 72$ in the following. We list $\mathcal{A}(i \cdot e)$ and values $\mathrm{dc}(i \cdot e, \mathcal{A}(i \cdot e))$ for each $i = 0, 1, \ldots, 28$ and $e = 0, 1, \ldots, q-1$.

```
 N = 251,  q = 128,  r(1)h(1) = 64

 e    A(e)    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

 0, 117.29, -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 1, 118.16, -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 2, 119.02, -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0
 3, 119.88, -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0
 4, 120.75, -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 5, 121.61, -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 6, 122.48, -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 7, 123.34, -1 -1 -1 -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1
 8, 124.21, -1 -1 -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
 9, 125.07, -1 -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1
10, 125.94, -1 -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1
11, 126.80, -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1
12, 127.67, -1 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1  2  2
13, 128.53, -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  2  2  2  2
14, 129.39, -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2
15, 130.26, -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2
16, 131.12, -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2
17, 131.99, -1 -1 -1 -1  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  3  3
18, 132.85, -1 -1 -1 -1  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  3  3  3
19, 133.72, -1 -1 -1 -1  0  0  0  0  0  0  0  1  1  1  1  1  1  1  2  2  2  2  2  2  3  3  3  3  3
20, 134.58, -1 -1 -1 -1  0  0  0  0  0  0  1  1  1  1  1  1  1  2  2  2  2  2  2  3  3  3  3  3  3
21, 135.45, -1 -1 -1 -1  0  0  0  0  0  0  1  1  1  1  1  1  2  2  2  2  2  2  3  3  3  3  3  3  4
22, 136.31, -1 -1 -1 -1  0  0  0  0  0  0  1  1  1  1  1  2  2  2  2  2  2  3  3  3  3  3  3  4  4
23, 137.18, -1 -1 -1 -1  0  0  0  0  0  1  1  1  1  1  1  2  2  2  2  2  3  3  3  3  3  3  4  4  4
24, 138.04, -1 -1 -1 -1  0  0  0  0  0  1  1  1  1  1  2  2  2  2  2  2  3  3  3  3  3  4  4  4  4
25, 138.90, -1 -1 -1  0  0  0  0  0  0  1  1  1  1  1  2  2  2  2  2  3  3  3  3  3  4  4  4  4  4
26, 139.77, -1 -1 -1  0  0  0  0  0  1  1  1  1  1  2  2  2  2  2  3  3  3  3  3  4  4  4  4  4  5
27, 140.63, -1 -1 -1  0  0  0  0  0  1  1  1  1  1  2  2  2  2  2  3  3  3  3  4  4  4  4  4  5  5
28, 141.50, -1 -1 -1  0  0  0  0  0  1  1  1  1  2  2  2  2  2  3  3  3  3  3  4  4  4  4  5  5  5
29, 142.36, -1 -1 -1  0  0  0  0  0  1  1  1  1  2  2  2  2  3  3  3  3  3  4  4  4  4  5  5  5  5
30, 143.23, -1 -1 -1  0  0  0  0  1  1  1  1  1  2  2  2  2  3  3  3  3  4  4  4  4  5  5  5  5  5
31, 144.09, -1 -1 -1  0  0  0  0  1  1  1  1  2  2  2  2  3  3  3  3  3  4  4  4  4  5  5  5  5  6
32, 144.96, -1 -1 -1  0  0  0  0  1  1  1  1  2  2  2  2  3  3  3  3  4  4  4  4  5  5  5  5  6  6
33, 145.82, -1 -1 -1  0  0  0  0  1  1  1  1  2  2  2  2  3  3  3  4  4  4  4  5  5  5  5  6  6  6
34, 146.69, -1 -1 -1  0  0  0  0  1  1  1  2  2  2  2  3  3  3  3  4  4  4  4  5  5  5  5  6  6  6
35, 147.55, -1 -1 -1  0  0  0  0  1  1  1  2  2  2  2  3  3  3  3  4  4  4  5  5  5  5  6  6  6  7
36, 148.41, -1 -1 -1  0  0  0  1  1  1  1  2  2  2  2  3  3  3  4  4  4  4  5  5  5  6  6  6  6  7
37, 149.28, -1 -1 -1  0  0  0  1  1  1  1  2  2  2  3  3  3  3  4  4  4  5  5  5  5  6  6  6  7  7
38, 150.14, -1 -1 -1  0  0  0  1  1  1  1  2  2  2  3  3  3  4  4  4  4  5  5  5  6  6  6  7  7  7
39, 151.01, -1 -1 -1  0  0  0  1  1  1  2  2  2  2  3  3  3  4  4  4  5  5  5  6  6  6  6  7  7  7
40, 151.87, -1 -1 -1  0  0  0  1  1  1  2  2  2  3  3  3  4  4  4  4  5  5  5  6  6  6  7  7  7  8
41, 152.74, -1 -1 -1  0  0  0  1  1  1  2  2  2  3  3  3  4  4  4  5  5  5  6  6  6  6  7  7  7  8
42, 153.60, -1 -1 -1  0  0  0  1  1  1  2  2  2  3  3  3  4  4  4  5  5  5  6  6  6  7  7  7  8  8
43, 154.47, -1 -1 -1  0  0  0  1  1  1  2  2  2  3  3  3  4  4  5  5  5  6  6  6  7  7  7  8  8  8
44, 155.33, -1 -1 -1  0  0  1  1  1  2  2  2  3  3  3  4  4  4  5  5  5  6  6  6  7  7  7  8  8  8
45, 156.20, -1 -1 -1  0  0  1  1  1  2  2  2  3  3  3  4  4  4  5  5  5  6  6  7  7  7  8  8  8  9
46, 157.06, -1 -1 -1  0  0  1  1  1  2  2  2  3  3  3  4  4  5  5  5  6  6  6  7  7  7  8  8  8  9
47, 157.92, -1 -1  0  0  0  1  1  1  2  2  2  3  3  4  4  4  5  5  5  6  6  6  7  7  8  8  8  9  9
48, 158.79, -1 -1  0  0  0  1  1  1  2  2  3  3  3  4  4  4  5  5  6  6  6  7  7  7  8  8  9  9  9
49, 159.65, -1 -1  0  0  0  1  1  1  2  2  3  3  3  4  4  4  5  5  6  6  6  7  7  8  8  8  9  9  9
50, 160.52, -1 -1  0  0  0  1  1  1  2  2  3  3  3  4  4  5  5  5  6  6  7  7  7  8  8  9  9  9 10
51, 161.38, -1 -1  0  0  0  1  1  2  2  2  3  3  4  4  4  5  5  6  6  6  7  7  8  8  8  9  9  9 10
52, 162.25, -1 -1  0  0  0  1  1  2  2  2  3  3  4  4  4  5  5  6  6  6  7  7  8  8  8  9  9 10 10
53, 163.11, -1 -1  0  0  0  1  1  2  2  2  3  3  4  4  5  5  5  6  6  7  7  7  8  8  9  9  9 10 10
54, 163.98, -1 -1  0  0  0  1  1  2  2  3  3  3  4  4  5  5  5  6  6  7  7  8  8  8  9  9 10 10 11
55, 164.84, -1 -1  0  0  0  1  1  2  2  3  3  3  4  4  5  5  6  6  6  7  7  8  8  9  9  9 10 10 11
56, 165.71, -1 -1  0  0  0  1  1  2  2  3  3  4  4  4  5  5  6  6  7  7  7  8  8  9  9 10 10 11 11
57, 166.57, -1 -1  0  0  0  1  1  2  2  3  3  4  4  4  5  5  6  6  7  7  8  8  8  9  9 10 10 11 11
```

58,	167.43,	-1	-1	0	0	1	1	1	2	2	3	3	4	4	5	5	5	6	6	7	7	8	8	9	9	10	10	10	11	11
59,	168.30,	-1	-1	0	0	1	1	1	2	2	3	3	4	4	5	5	6	6	7	7	7	8	8	9	9	10	10	11	11	12
60,	169.16,	-1	-1	0	0	1	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	9	10	10	11	11	12
61,	170.03,	-1	-1	0	0	1	1	2	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12
62,	170.89,	-1	-1	0	0	1	1	2	2	3	3	4	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12
63,	171.76,	-1	-1	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	9	10	10	11	11	12	12
64,	172.62,	-1	-1	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13
65,	173.49,	-1	-1	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13
66,	174.35,	-1	-1	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	12	12	13	13
67,	175.22,	-1	-1	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	8	8	9	9	10	10	11	11	12	12	13	13
68,	176.08,	-1	-1	0	0	1	1	2	2	3	3	4	4	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13
69,	176.94,	-1	-1	0	0	1	1	2	2	3	3	4	5	5	6	6	7	7	8	8	9	9	10	10	11	12	12	13	13	14
70,	177.81,	-1	-1	0	0	1	1	2	2	3	4	4	5	5	6	6	7	7	8	8	9	10	10	11	11	12	12	13	13	14
71,	178.67,	-1	-1	0	0	1	1	2	2	3	4	4	5	5	6	6	7	7	8	9	9	10	10	11	11	12	12	13	14	14
72,	179.54,	-1	-1	0	0	1	1	2	3	3	4	4	5	5	6	6	7	8	8	9	9	10	10	11	12	12	13	13	14	14
73,	180.40,	-1	-1	0	0	1	1	2	3	3	4	4	5	5	6	7	7	8	8	9	9	10	11	11	12	12	13	13	14	15
74,	181.27,	-1	-1	0	0	1	1	2	3	3	4	4	5	6	6	7	7	8	8	9	10	10	11	11	12	12	13	14	14	15
75,	182.13,	-1	-1	0	0	1	2	2	3	3	4	4	5	6	6	7	7	8	9	9	10	10	11	11	12	13	13	14	14	15
76,	183.00,	-1	-1	0	0	1	2	2	3	3	4	5	5	6	6	7	7	8	9	9	10	10	11	12	12	13	13	14	15	15
77,	183.86,	-1	-1	0	0	1	2	2	3	3	4	5	5	6	6	7	8	8	9	9	10	11	11	12	12	13	14	14	15	15
78,	184.73,	-1	-1	0	0	1	2	2	3	3	4	5	5	6	6	7	8	8	9	10	10	11	11	12	13	13	14	14	15	16
79,	185.59,	-1	-1	0	0	1	2	2	3	3	4	5	5	6	7	7	8	8	9	10	10	11	12	12	13	13	14	15	15	16
80,	186.45,	-1	-1	0	0	1	2	2	3	4	4	5	5	6	7	7	8	9	9	10	10	11	12	12	13	14	14	15	15	16
81,	187.32,	-1	-1	0	0	1	2	2	3	4	4	5	5	6	7	7	8	9	9	10	11	11	12	12	13	14	14	15	16	16
82,	188.18,	-1	-1	0	0	1	2	2	3	4	4	5	6	6	7	7	8	9	9	10	11	11	12	13	13	14	15	15	16	16
83,	189.05,	-1	-1	0	0	1	2	2	3	4	4	5	6	6	7	8	8	9	10	10	11	11	12	13	13	14	15	15	16	17
84,	189.91,	-1	-1	0	0	1	2	2	3	4	4	5	6	6	7	8	8	9	10	10	11	12	12	13	14	14	15	16	16	17
85,	190.78,	-1	-1	0	1	1	2	2	3	4	4	5	6	6	7	8	8	9	10	10	11	12	12	13	14	14	15	16	16	17
86,	191.64,	-1	-1	0	1	1	2	3	3	4	5	5	6	7	7	8	9	9	10	11	11	12	13	13	14	15	15	16	17	17
87,	192.51,	-2	-1	0	1	1	2	3	3	4	5	5	6	7	7	8	9	9	10	11	11	12	13	13	14	15	15	16	17	18
88,	193.37,	-2	-1	0	1	1	2	3	3	4	5	5	6	7	7	8	9	9	10	11	12	12	13	14	14	15	16	16	17	18
89,	194.24,	-2	-1	0	1	1	2	3	3	4	5	5	6	7	8	8	9	10	10	11	12	12	13	14	14	15	16	17	17	18
90,	195.10,	-2	-1	0	1	1	2	3	3	4	5	6	6	7	8	8	9	10	10	11	12	13	13	14	15	15	16	17	17	18
91,	195.96,	-2	-1	0	1	1	2	3	3	4	5	6	6	7	8	8	9	10	11	11	12	13	13	14	15	16	16	17	18	18
92,	196.83,	-2	-1	0	1	1	2	3	3	4	5	6	6	7	8	9	9	10	11	11	12	13	14	14	15	16	16	17	18	19
93,	197.69,	-2	-1	0	1	1	2	3	4	4	5	6	6	7	8	9	9	10	11	12	12	13	14	14	15	16	17	17	18	19
94,	198.56,	-2	-1	0	1	1	2	3	4	4	5	6	7	7	8	9	9	10	11	12	12	13	14	15	15	16	17	18	18	19
95,	199.42,	-2	-1	0	1	1	2	3	4	4	5	6	7	7	8	9	10	10	11	12	13	13	14	15	16	16	17	18	18	19
96,	200.29,	-2	-1	0	1	1	2	3	4	4	5	6	7	7	8	9	10	10	11	12	13	13	14	15	16	16	17	18	19	19
97,	201.15,	-2	-1	0	1	1	2	3	4	4	5	6	7	8	8	9	10	11	11	12	13	14	14	15	16	17	17	18	19	20
98,	202.02,	-2	-1	0	1	1	2	3	4	5	5	6	7	8	8	9	10	11	11	12	13	14	14	15	16	17	18	18	19	20
99,	202.88,	-2	-1	0	1	2	2	3	4	5	5	6	7	8	8	9	10	11	12	12	13	14	15	15	16	17	18	19	19	20
100,	203.75,	-2	-1	0	1	2	2	3	4	5	5	6	7	8	9	9	10	11	12	12	13	14	15	16	16	17	18	19	20	20
101,	204.61,	-2	-1	0	1	2	2	3	4	5	6	6	7	8	9	9	10	11	12	13	13	14	15	16	17	17	18	19	20	20
102,	205.47,	-2	-1	0	1	2	2	3	4	5	6	6	7	8	9	10	10	11	12	13	14	14	15	16	17	18	18	19	20	21
103,	206.34,	-2	-1	0	1	2	2	3	4	5	6	6	7	8	9	10	10	11	12	13	14	14	15	16	17	18	19	19	20	21
104,	207.20,	-2	-1	0	1	2	2	3	4	5	6	7	7	8	9	10	11	11	12	13	14	15	15	16	17	18	19	20	20	21
105,	208.07,	-2	-1	0	1	2	2	3	4	5	6	7	7	8	9	10	11	11	12	13	14	15	16	16	17	18	19	20	21	21
106,	208.93,	-2	-1	0	1	2	3	3	4	5	6	7	7	8	9	10	11	12	12	13	14	15	16	17	17	18	19	20	21	22
107,	209.80,	-2	-1	0	1	2	3	3	4	5	6	7	8	8	9	10	11	12	13	13	14	15	16	17	18	18	19	20	21	22
108,	210.66,	-2	-1	0	1	2	3	3	4	5	6	7	8	8	9	10	11	12	13	14	14	15	16	17	18	19	19	20	21	22
109,	211.53,	-2	-1	0	1	2	3	3	4	5	6	7	8	9	9	10	11	12	13	14	15	15	16	17	18	19	20	20	21	22
110,	212.39,	-2	-1	0	1	2	3	3	4	5	6	7	8	9	10	10	11	12	13	14	15	16	16	17	18	19	20	21	22	22
111,	213.25,	-2	-1	0	1	2	3	4	4	5	6	7	8	9	10	10	11	12	13	14	15	16	17	17	18	19	20	21	22	23
112,	214.12,	-2	-1	0	1	2	3	4	4	5	6	7	8	9	10	11	11	12	13	14	15	16	17	18	18	19	20	21	22	23
113,	214.98,	-2	-1	0	1	2	3	4	5	5	6	7	8	9	10	11	12	12	13	14	15	16	17	18	19	20	20	21	22	23
114,	215.85,	-2	-1	0	1	2	3	4	5	5	6	7	8	9	10	11	12	13	13	14	15	16	17	18	19	20	21	21	22	23
115,	216.71,	-2	-1	0	1	2	3	4	5	5	6	7	8	9	10	11	12	13	14	14	15	16	17	18	19	20	21	22	23	23
116,	217.58,	-2	-1	0	1	2	3	4	5	6	6	7	8	9	10	11	12	13	14	15	16	16	17	18	19	20	21	22	23	24
117,	218.44,	-2	-1	0	1	2	3	4	5	6	7	7	8	9	10	11	12	13	14	15	16	17	17	18	19	20	21	22	23	24
118,	219.31,	-2	-1	0	1	2	3	4	5	6	7	8	8	9	10	11	12	13	14	15	16	17	18	19	19	20	21	22	23	24
119,	220.17,	-2	-1	0	1	2	3	4	5	6	7	8	9	9	10	11	12	13	14	15	16	17	18	19	20	21	22	22	23	24
120,	221.04,	-2	-1	0	1	2	3	4	5	6	7	8	9	10	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
121,	221.90,	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	12	13	14	15	16	17	18	19	20	21	22	23	24	25
122,	222.76,	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	14	15	16	17	18	19	20	21	22	23	24	25
123,	223.63,	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	17	18	19	20	21	22	23	24	25
124,	224.49,	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	21	22	23	24	25
125,	225.36,	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
126,	115.56,	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
127,	116.43,	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27

C Designed use of wrapping - Example

For the parameters given by the $N = 251$ row of Table 2, we give an explicit set of instructions for determining the private key $\mathbf{f}$ assuming that it is given by a LHW-$\mathbf{F}$. The first column contains the e values. Applying methods of Section 4 with e and $e' = e + 1$, we can obtain the term or sum of terms given in the third column. Using this together with information already obtained in higher rows, the term given in the last column is obtained. This is constructed from information given in Appendix B.

e	e'			e	e'		
1	2	$\mathbf{f}_{28}$	$\mathbf{f}_{28}$	61	62	$\mathbf{f}_8 + \mathbf{f}_{10}$	$\mathbf{f}_8$
15	16	$\mathbf{f}_{21}$	$\mathbf{f}_{21}$	86	87	$\mathbf{f}_0 + \mathbf{f}_{28}$	$\mathbf{f}_0$
48	49	$\mathbf{f}_{23}$	$\mathbf{f}_{23}$	116	117	$\mathbf{f}_9 + \mathbf{f}_{20}$	$\mathbf{f}_9$
51	52	$\mathbf{f}_{27}$	$\mathbf{f}_{27}$	120	121	$\mathbf{f}_{13} + \mathbf{f}_{14}$	$\mathbf{f}_{13}$
56	57	$\mathbf{f}_{20}$	$\mathbf{f}_{20}$	121	122	$\mathbf{f}_{15} + \mathbf{f}_{16}$	$\mathbf{f}_{15}$
84	85	$\mathbf{f}_3$	$\mathbf{f}_3$	5	6	$\mathbf{f}_{10} + \mathbf{f}_{11}$	$\mathbf{f}_{10}$
102	103	$\mathbf{f}_{25}$	$\mathbf{f}_{25}$	10	11	$\mathbf{f}_6 + \mathbf{f}_{18}$	$\mathbf{f}_6$
23	24	$\mathbf{f}_{14} + \mathbf{f}_{25}$	$\mathbf{f}_{14}$	17	18	$\mathbf{f}_{11} + \mathbf{f}_{19} + \mathbf{f}_{26}$	$\mathbf{f}_{26}$
26	27	$\mathbf{f}_{22} + \mathbf{f}_{27}$	$\mathbf{f}_{22}$	19	20	$\mathbf{f}_{10} + \mathbf{f}_{17} + \mathbf{f}_{23}$	$\mathbf{f}_{17}$
33	34	$\mathbf{f}_{10} + \mathbf{f}_{14}$	$\mathbf{f}_{10}$	4	5	$\mathbf{f}_{12} + \mathbf{f}_{13} + \mathbf{f}_{14}$	$\mathbf{f}_{12}$
40	41	$\mathbf{f}_{18} + \mathbf{f}_{21}$	$\mathbf{f}_{18}$	8	9	$\mathbf{f}_7 + \mathbf{f}_{22} + \mathbf{f}_{23}$	$\mathbf{f}_7$
41	42	$\mathbf{f}_{24} + \mathbf{f}_{27}$	$\mathbf{f}_{24}$	74	75	$\mathbf{f}_5 + \mathbf{f}_{17} + \mathbf{f}_{24}$	$\mathbf{f}_5$
54	55	$\mathbf{f}_{16} + \mathbf{f}_{23}$	$\mathbf{f}_{16}$	57	58	$\mathbf{f}_4 + \mathbf{f}_{13} + \mathbf{f}_{22} + \mathbf{f}_{24}$	$\mathbf{f}_4$
59	60	$\mathbf{f}_{19} + \mathbf{f}_{21}$	$\mathbf{f}_{19}$	46	47	$\mathbf{f}_2 + \mathbf{f}_{13} + \mathbf{f}_{24} + \mathbf{f}_{27}$	$\mathbf{f}_2$

The reader can check that the only term not appearing in the last column is $\mathbf{f}_1$. It may readily be set to all remaining terms.

Careful counting will show that 52 queries were needed to determine $\mathbf{f}$ completely. Since we've been very lazy in making this table, there would be ways to reduce this number.

D Entropy of f in the binary-F case

Content of this section may not qualify as an attack on NTRU, but contains information which could be useful when used together with some form of attack.

We work in the binary-$\mathbf{F}$ setting and assume that the constant term of $\mathbf{f}$ is not 4, so that the coefficient set of $\mathbf{f}$ is $\{0, 1, 2, 3\}$.

Let us denote by $(f_0, f_1, \dots, f_{N-1})$, the coefficients of $\mathbf{f}$. Likewise, the coefficients of $\mathbf{F}$ will be denoted with $(F_0, \dots, F_{N-1})$. Notice

$$f_i = 2 \cdot F_i + F_{i-1}$$

for all $i \neq 0$. So the parity (E/O) of f_i determines F_{i-1} completely. Similarly, knowing whether f_i belongs to the set $L = \{0, 1\}$ or $H = \{2, 3\}$ determines F_i completely. Once more, if f_i belongs to $I = \{1, 2\}$, then $F_i = 1 - F_{i-1}$ and if f_i belongs to $B = \{0, 3\}$, then $F_i = F_{i-1}$.

We may use this argument as follows. Suppose we know that $\mathbf{f}$ is of the form

$$\mathbf{f} = (?, L, H, H, H, L, L, H, L, H, \dots).$$

Then we must have

$$\mathbf{F} = (?, 0, 1, 1, 1, 0, 0, 1, 0, 1, \dots)$$

and hence

$$\mathbf{f} = (?, ?, 2, 3, 3, 1, 0, 2, 1, 2, \dots)$$

We may conclude that, even though the coefficient set of $\mathbf{f}$ is of size 4, knowing one bit of information for each f_i ($i \neq 0$), in the form of E/O, L/H, or I/B, is enough to determine $\mathbf{f}$ almost completely.

This may not have any impact on the security of NTRU cryptosystem by itself, but may be useful when combined with other methods of attacks. For example, for the $N = 251$ case, still assuming that the coefficients of $\mathbf{f}$ belong to the set $\{0, 1, 2, 3\}$, using Appendix B, we see that

$$\mathbf{a}(70) = 70 \cdot \mathbf{f} + q(\mathbf{f}_0 + \mathbf{f}_1),$$

in the notation of Section 4. Hence, with just one query to the decryption machine, we can obtain

$$\mathrm{Mod}_{\mathbf{p}}\Big(q_{\mathbf{p}}\big(\mathrm{Mod}_{\mathbf{p}}(\mathbf{a}(70)) - 70\big)\Big) = \mathbf{f}_0 + \mathbf{f}_1.$$

This determines whether each f_i belongs to L or H, so determines $\mathbf{f}$ almost completely.

If we are not so lucky as to find such an e, we could use the difference of two queries to the decryption machine in a similar attack.