NTRU Cryptosystems Technical Report Report # 18, Version 1: Estimating Decryption Failure Probabilities for NTRUEncrypt

J.H. Silverman[1] and W. Whyte[2]

[1] NTRU Cryptosystems, Inc., 5 Burlington Woods, Burlington, MA 01803 USA
jhs@ntru.com
[2] NTRU Cryptosystems, Inc., 5 Burlington Woods, Burlington, MA 01803 USA
wwhyte@ntru.com

Abstract. We describe a theoretical method for estimating the decryption failure probability for NTRUEncrypt with different centering algorithms. We apply this method to a suggested parameter set and compare it with experiment. For the recommended parameter sets in [1], the chance of a decryption failure is less than 2^{-100}.

1 NTRUEncrypt parameters and basic definitions

An implementation of the NTRUEncrypt Public Key Cryptosystem [3] is specified by the following parameters:

N	*Degree Parameter.* A positive integer. The associated NTRU lattice has dimension $2N$.
q	*Large Modulus.* A positive integer. The associated NTRU lattice is a convolution modular lattice of modulus q.
p	*Small Modulus.* An integer or a polynomial.
$\mathcal{S}_f, \mathcal{S}_g$	*Private Key Spaces.* Sets of polynomials from which the private keys are selected.
$\mathcal{S}_m, \mathcal{S}_r$	*Plaintext Spaces.* Sets of polynomials from which the (padded and encoded) plaintexts are selected.

Definition 1. *Most operations take place in the* Ring of Convolution Polynomials

$$R = \frac{\mathbb{Z}[X]}{(X^N - 1)}.$$

Multiplication of polynomials in this ring corresponds to the convolution product of their associated vectors:

$$c(X) = a(X) * b(X) \qquad \textit{with} \qquad c_k = \sum_{i+j=k(\bmod N)} a_i * b_j$$

Remark 1. The following two parameter selection criteria are vital for secure implementation of NTRUEncrypt, although encryption and decryption will work even if they are violated.

- The degree parameter N must be prime. (See [2].)
- The small and large moduli p and q must be relatively prime in the ring R. Equivalently, the three quantities

$$p, \quad q, \quad X^N - 1$$

must generate the unit ideal in the ring $\mathbb{Z}[X]$. (In the extreme case that p divides q, the plaintext can be recovered directly by reducing the ciphertext modulo p.)

Definition 2. *A polynomial* $a(X) = a_0 + a_1 X + \cdots + a_{N-1}X^{N-1}$ *is identified with its vector of coefficients* $\mathbf{a} = [a_0, a_1, \ldots, a_{N-1}]$. *The maximum and minimum coefficients of a polynomial or vector are are denoted by*

$$\mathrm{Max}(a(X)) = \max\{a_0, a_1, \cdots, a_{N-1}\} \quad \textit{and} \quad \mathrm{Min}(a(X)) = \min\{a_0, a_1, \cdots, a_{N-1}\}.$$

The width *of a polynomial* $a(X)$ *is the difference between its largest and smallest coefficients,*

$$\mathrm{Width}(a(X)) = \mathrm{Max}(a(X)) - \mathrm{Min}(a(X)).$$

2 NTRUEncrypt basic operations

2.1 Key generation

An NTRUEncrypt *Private Key* is a pair $(f, g) \in \mathcal{S}_f \times \mathcal{S}_g$. The associated NTRUEncrypt *Public Key* is a polynomial $h \in R$ satisfying

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

Generally, f will be chosen to be invertible modulo q. If we let f_q^{-1} denote the inverse of f in the quotient ring R/qR, then $h \equiv p * f_q^{-1} * g \pmod{q}$. For practical purposes, only the polynomial f is needed for decryption.

2.2 Encryption

The plaintext M and additional random bits R are used to select a pair of encoded plaintext polynomials $(r, m) \in \mathcal{S}_r \times \mathcal{S}_m$ according to a public encoding scheme $\mathcal{E}$. A minimal property of $\mathcal{E}$ is that knowledge of (r, m) allows easy recovery of M. In practice, $\mathcal{E}$ will have the property that knowledge of m alone allows easy recovery of M and R, from which r may be recomputed using $\mathcal{E}$. The ciphertext e is the polynomial

$$e = r * h + m \pmod{q}.$$

2.3 Decryption

Decryption consists of the following steps:

1. Compute the polynomial
$$a = f * e \pmod q \tag{1}$$
and place the coefficients of a into an appropriate interval modulo q by use of an appropriate centering method, as described below in Section 2.4.
2. Reduce a modulo p and compute
$$b = f_p^{-1} * a \pmod p, \tag{2}$$
where f_p^{-1} is the inverse of f modulo p, i.e., in the ring R/pR.
3. Identify m as the (unique) element in $\mathcal{S}_m$ with the property that
$$m \equiv b \pmod p.$$
4. Reverse the encoding process $\mathcal{E}$ to recover the plaintext M and randomizing bits R. This completes the decryption process, but one normally also performs the following validation step.
5. Use $\mathcal{E}$ to regenerate r and check that $r * h + m \pmod q$ agrees with e.

Remark 2. It is possible to eliminate Step 2 of the decryption algorithm by choosing the private key polynomial f to have the form
$$f = 1 + p * F.$$
This may significantly improve efficiency [4]. However, the target lattice vector for the underlying closest vector problem changes from (f, g) to (F, g), so when using this form of f, one must analyze the difficulty of solving the modified vector problem.

We see how decryption works by considering (1) above. Using the definitions of e, h, we obtain
$$a = p * r * g + f * m \pmod q \ . \tag{3}$$
Because the polynomials r, g, f, m are small, their products will in general have low width. Thus we can find a mod q interval such that (3) is an exact equality. If we have selected the correct interval, (2) will clearly recover m. If we have selected the wrong interval, the recovered value will differ from m by some multiple of $q \pmod p$. A decryption failure will occur if $\text{Width}(p*r*g+f*m) \geq q$ (we refer to this as a *gap failure*), or if $\text{Width}(p * r * g + f * m) < q$ but we have reduced into the wrong interval (we refer to this as a *wrap failure*). The concern of this paper is to estimate the probability of decryption failures. In order to do this, we must consider how the reduction interval is chosen.

2.4 Centering Methods and Decryption Failures

We here consider the problem of choosing the correct interval for the coefficients of $a = f * e \pmod q$.

First, we note that if a is reduced into the correct interval, then

$$a(1) = p \cdot r(1) \cdot g(1) + f(1) \cdot m(1) \ .$$

In practice, f, g, r are chosen from sets of binary polynomials for which one knows the values of $f(1), g(1), r(1)$. Thus if the decryptor knows the value of $m(1)$, he knows $p \cdot r(1) \cdot g(1) + f(1) \cdot m(1)$ and can choose the mod q interval such that $a(1)$ is equal to this value.

Notice that the ciphertext at $X = 1$ has the value

$$e(1) \equiv r(1) \cdot h(1) + m(1) \pmod q. \tag{4}$$

Further, the average value of $m(1)$ is $N/2$. Thus the decryptor should compute the value of $m(1)$ satisfying the congruence (4) and lying in the interval

$$\frac{N-q}{2} \leq m(1) < \frac{N+q}{2}. \tag{5}$$

Remark 3. It is important that the encryptor is not able to generate allowable values of m that do not satisfy (5). In practice, the polynomial m is masked using a hash function, so the encryptor has no control over its form. Thus the relevant quantity is the probability that a random binary polynomial m does not satisfy (5). The value of $m(1)$ as m varies is simply the sum of N independent binary random variables, so $m(1)$ follows a binomial distribution. Hence

$$\text{Prob}\left(\left|m(1) - \frac{N}{2}\right| \geq \frac{q}{2}\right) = \sum_{x \leq (N-q)/2} \binom{N}{x} + \sum_{x \geq (N+q)/2} \binom{N}{x}.$$

Note that randomness is added to the message on encryption, so if the masking process results in a value of $m(1)$ that is too large or too small the encryptor can simply select a different random string and try again.

These observations prompt the following centering method, which we call `center1`, and which corresponds to the method `center1` in [7].

1. Calculate $m(1)$ as $e(1) - r(1) \cdot h(1) \pmod q$, reduced into the interval given by (5).
2. Denote a reduced into the range $[0, q-1]$ by $\underline{a}$. (The underline is intended to suggest "reduced into the lowest range possible").
3. Calculate $\underline{a}(1)$. This will differ from $p \cdot r(1) \cdot g(1) + f(1) \cdot m(1)$ by kq, for some integer k.
4. Add q to the lowest k entries of $\underline{a}$ to obtain a reduced into the correct interval.

This centering method will always recover $p*r*g+f*m$ exactly, unless there is a gap failure. To see why, assume that the width of $a \pmod q$ is w, and assume that $a \pmod q$ has been reduced into the correct range such that $a(1) = S$, which will be the case when this algorithm has run. Then either $p*r*g+f*m = a$ exactly, or $p*r*g+f*m = a+k*q$, where $k(1) = 0$, i.e. k has some positive and some negative terms. The effect of adding $k*q$ to a will be to produce an a' with a width of at least $2q-w$; but this is greater than q. Therefore the above method will only give a decryption failure if $\text{Width}(p*r*g+f*m) > q$, which gives a gap failure.

We also consider the following slightly more efficient, but less reliable, centering algorithm, which we call `center2`. This method corresponds to the method `center2` in [7].

1. Calculate $m(1)$ as $e(1) - r(1)\cdot h(1) \pmod q$, reduced into the interval given by (5).
2. Calculate the average value of the coefficients of a, correctly centered, as
$$\alpha = \left\lfloor \frac{p\cdot r(1)\cdot g(1) + f(1)\cdot m(1)}{N} \right\rceil .$$
3. Reduce a into the interval $[\alpha - q/2, \alpha + q/2)$.

This algorithm will give a decryption failure if any coefficient of $prg+fm$ differs from α by more than $q/2$. Note that if $\alpha < q/2$ we could choose to reduce always into the interval $[0, q-1]$; however, this is not specified in [1, 7].

2.5 A typical parameter set

An NTRUEncrypt parameter set is described by specifying three integers (N, p, q), four sets of polynomials $(\mathcal{S}_f, \mathcal{S}_g, \mathcal{S}_r, \mathcal{S}_m)$, and a centering method for use in decryption. (It is also possible to take p to be a polynomial, for example $p = X+2$.) We set some notation to be used in specifying the sets of polynomials.

$\hat{R}$	The set of binary polynomials in $R = \mathbb{Z}[X]/(X^N-1)$.
$\hat{R}(d)$	The set of binary polynomials in R with exactly d ones and $N-d$ zeroes.

The parameter set that we will use for illustration in this article has

$$(N, p, q) = (251, 2, 239), \qquad f = 1 + p*F,$$
$$F, g, r \in \hat{R}(72), \qquad m \in \hat{R}, \qquad \text{centering method: } \texttt{center2}.$$

This is the parameter set `ees251ep4` in [1].

We note that for this parameter set, the probability that $m(1)$ fails to satisfy (5) is

$$\text{Prob}\left(\left| m(1) - \frac{251}{2} \right| \geq \frac{239}{2} \right) \approx 2^{-207.54}.$$

However, if we used a smaller value of q, for example $q = 128$, then the probability would be considerably larger,

$$\text{Prob}\left(\left|m(1) - \frac{251}{2}\right| \geq \frac{128}{2}\right) \approx 2^{-51.7}.$$

3 Estimating failure probabilities

NTRUEncrypt decryption will fail when the centering method fails to correctly recover $prg + fm$. It has recently been shown [6, 8, 10] that decryption failures leak significant information about the private key. Additionally, [6, 10] note that a significant probability of decryption failures on validly encrypted messages makes it impossible to construct a meaningful proof of security, and [7] proposes a padding scheme for which a proof of security can be given even in the presence of decryption failures, so long as their chance of occurring is very low. It is therefore interesting and important to be able to accurately estimate the probability of a decryption failure. In this section, we evaluate methods for determining this probability for both of the centering methods presented above. (This has previously been studied, though in less depth, in [11].)

We describe a theoretical method for evaluating coefficient and width probabilities when m is binary and f, g, and r are binary with a fixed number of ones. We then compare our theoretical formulas with experimental results, using the parameter set described in Section 2.5 We will see that theory and experiment are in close agreement.

3.1 Theoretical coefficient distribution of products of binary polynomials

In NTRUEncrypt, we are interested in the distribution of the coefficients of

$$a(X) = p * r(X) * g(X) + f(X) * m(X).$$

Recall that $\hat{R}$ denotes the set of binary polynomials in R and $\hat{R}(d)$ is the set of binary polynomials in R with exactly d ones and $N - d$ zeroes. For typical parameter sets, $r(X)$, $g(X)$ and $f(X)$ are chosen to have a fixed number of ones and the "plaintext" $m(X)$ is a random binary polynomial. However, we are interested in the probability that the spread of coefficients in $a(X)$ is large, so it is prudent to assume that the attacker chooses $m(X)$ to maximize this spread. If $m(X)$ has many ones, then all of the coefficients of $a(X)$ will tend to be larger, and similarly if $m(X)$ has few ones, then all of the coefficients of $a(X)$ will tend to be smaller. A brief examination of the binomial distribution shows that the spread from largest coefficient to smallest coefficient, or from the average to the largest or smallest coefficient, in $f(X) * m(X)$ will be largest if $m(X)$ has an equal number of zeros and ones. We will thus suppose that f, g, r, m are chosen from the sets

$$f(X) \in \hat{R}(d_f), \qquad g(X) \in \hat{R}(d_g), \qquad r(X) \in \hat{R}(d_r), \qquad m(X) \in \hat{R}(d_m),$$

where in practice we will take $d_m = \lfloor N/2 \rfloor$.

For any polynomial $P(X)$ chosen from some space of polynomials, we let $\mathrm{Coef}(P)$ denote a randomly chosen coefficient of P; and similarly we let $\mathrm{Coef}'(P)$ denote some other randomly chosen coefficient of P.

We consider first the distribution of the coefficients of a product $f * m$. A coefficient of $f*m$ is formed by adding up d_m elements chosen at random *without replacement* from a pool of d_f ones and $N-d_f$ zeros. Thus each coefficient of $f*m$ satisfies a *hypergeometric distribution*,

$$\mathrm{Prob}(\mathrm{Coef}(f*m) = u) = \frac{\binom{d_f}{u}\binom{N-d_f}{d_m-u}}{\binom{N}{d_m}}. \tag{6}$$

Next we turn to the product $r * g$. The analysis is identical, only the parameters have changed. Thus each coefficient of $r * g$ satisfies

$$\mathrm{Prob}(\mathrm{Coef}(r*g) = v) = \frac{\binom{d_g}{v}\binom{N-d_g}{d_r-v}}{\binom{N}{d_r}}. \tag{7}$$

We really need the coefficient distribution for the triple product $p*r*g$. If p is an integer, the distribution of coefficients is again hypergeometric, although the values only take on the multiples of p. If p is not an integer, say $p = X + 2$, the analysis is similar, but a bit more complicated. For simplicity of exposition, and for consistency with the parameter sets in [1], we will restrict ourselves to the case that p is an integer.

Finally we turn to coefficients of the quantity

$$a = p*r*g + f*m.$$

The coefficients of $p*r*g$ and $f*m$ are certainly independent of one another, so we find that

$$\begin{aligned}\mathrm{Prob}(\mathrm{Coef}(p*r*g+f*m) = x) \\ = \sum_{\beta} \mathrm{Prob}(\mathrm{Coef}(p*r*g) = x-\beta) \cdot \mathrm{Prob}(\mathrm{Coef}(f*m) = \beta).\end{aligned} \tag{8}$$

Remark 4. When implementing these formulas, much time may be saved by summing only over the values for which the associated probabilities are nonzero. For example,

$$\mathrm{Prob}(\mathrm{Coef}(f*m) = u) = \frac{\binom{d_f}{u}\binom{N-d_f}{d_m-u}}{\binom{N}{d_m}} \quad \text{and} \quad \mathrm{Prob}(\mathrm{Coef}(r*g) = v) = \frac{\binom{d_g}{v}\binom{N-d_g}{d_r-v}}{\binom{N}{d_r}}$$

are nonzero, respectively, only for

$$\max\{0, d_m+d_f-N\} \le u \le \min\{d_m, d_f\} \text{ and } \max\{0, d_r+d_g-N\} \le v \le \min\{d_r, d_g\}.$$

3.2 Coefficient distributions for polynomials whose coefficients are independent random variables

Let $c(X) \in \mathbb{Z}[X]$ be a polynomial whose coefficients are chosen independently according to some known probability distribution. In other words, for each value of t one knows the value of the probability $\mathrm{Prob}(c_k = t)$, and this value is independent of the choice of index k. Then the probability that the largest coefficient of $c(X)$ is larger than a given value T may be computed by the following formula.

$$\begin{aligned}
&\mathrm{Prob}(\,\mathrm{Max}(c(X)) > T) \\
&\quad = 1 - \mathrm{Prob}(\mathrm{Max}(c(X)) \le T) \\
&\quad = 1 - \mathrm{Prob}(c_0 \le T) \cdot \mathrm{Prob}(c_1 \le T) \cdots \mathrm{Prob}(c_{N-1} \le T) \\
&\quad = 1 - \mathrm{Prob}(c_k \le T)^N \\
&\quad = 1 - (1 - \mathrm{Prob}(c_k > T))^N \\
&\quad = 1 - \left(1 - \sum_{t>T} \mathrm{Prob}(c_k = t)\right)^N .
\end{aligned}$$

The probability that the smallest coefficient of $c(X)$ is smaller than T may be computed by the analogous formula

$$\mathrm{Prob}(\mathrm{Min}(c(X)) < T) = 1 - \left(1 - \sum_{t<T} \mathrm{Prob}(c_k = t)\right)^N .$$

The width of a polynomial is the difference of its maximum and minimum coefficients, so

$$\mathrm{Prob}(\mathrm{Width}(c(X)) > T) = \sum_t \mathrm{Prob}(\mathrm{Min}(c(X)) = t)\,\mathrm{Prob}(\mathrm{Max}(c(X)) > T + t).$$

Note that the probability that $\mathrm{Min}(c(X))$ equals t may be computed (using the earlier formulas) as the difference

$$\mathrm{Prob}(\mathrm{Min}(c(X)) = t) = \mathrm{Prob}(\mathrm{Min}(c(X)) < t + 1) - \mathrm{Prob}(\mathrm{Min}(c(X)) < t).$$

3.3 Theoretical width distributions of sums and products of binary polynomials

Throughout this section, we let

$$a = p * r * g + f * m$$

with p an integer. We denote by $\mathrm{Max}(a)$, $\mathrm{Min}(a)$, and $\mathrm{Width}(a)$ the random variables that, respectively, assign to a polynomial a the maximum, minimum, and width of its coefficients, where

$$\mathrm{Width}(a) = \mathrm{Max}(a) - \mathrm{Min}(a).$$

We now make the (slightly incorrect) assumption that the coefficients of a are independent. This independence assumption is discussed below in Section 3.5. Under this assumption, we compute

$$\begin{aligned}\text{Prob}(\text{Max}(a) \geq \mu) &= 1 - \text{Prob}(\text{Max}(a) < \mu) \\ &= 1 - \text{Prob}(\text{Coef}_i(a) < \mu \text{ for all } 0 \leq i < N) \\ &\approx 1 - \text{Prob}(\text{Coef}(a) < \mu)^N \\ &= 1 - \big(1 - \text{Prob}(\text{Coef}(a) \geq \mu)\big)^N.\end{aligned}$$

Note that the quantity $\text{Prob}(\text{Coef}(a) \geq \mu)$ may be computed as

$$\text{Prob}(\text{Coef}(a) \geq \mu) = \sum_{x \geq \mu} \text{Prob}(\text{Coef}(a) = x),$$

where the individual probabilities $\text{Prob}(\text{Coef}(a) = x)$ in the sum are given by (8). Finally, we compute

$$\text{Prob}(\text{Max}(a) = \mu) = \text{Prob}(\text{Max}(a) \geq \mu) - \text{Prob}(\text{Max}(a) \geq \mu + 1).$$

Remark 5. Expanding the quantity $\big(1 - \text{Prob}(\text{Coef}(a) \geq \mu)\big)^N$ using the binomial theorem, we see that if $\text{Prob}(\text{Coef}(a) \geq \mu)$ is small, then

$$\text{Prob}(\text{Max}(a) \geq \mu) \approx N \,\text{Prob}(\text{Coef}(a) \geq \mu).$$

This makes sense, since it is unlikely there will be more than one coefficient larger than μ. Similarly, the main term for $\text{Prob}(\text{Max}(a) = \mu)$ is $N \,\text{Prob}(\text{Coef}(a) = \mu)$. If we take the first two terms in the expansion, we get

$$\begin{aligned}\text{Prob}(\text{Max}(a) = \mu) \approx N \,\text{Prob}(\text{Coef}(a) = \mu) &- \frac{N^2 + N}{2} \,\text{Prob}(\text{Coef}(a) = \mu) \\ &\times \big[\text{Prob}(\text{Coef}(a) \geq \mu) + \text{Prob}(\text{Coef}(a) \geq \mu + 1)\big].\end{aligned}$$

The importance of the second order terms depends on the relative sizes of N and the various probabilities.

An analogous calculation gives us formulas for the probability distribution of the minimum coefficient of a. Thus

$$\begin{aligned}\text{Prob}(\text{Min}(a) \leq \nu) &= 1 - \text{Prob}(\text{Min}(a) > \nu) \\ &= 1 - \text{Prob}(\text{Coef}_i(a) > \nu \text{ for all } 0 \leq i < N) \\ &\approx 1 - \text{Prob}(\text{Coef}(a) > \nu)^N \\ &= 1 - \big(1 - \text{Prob}(\text{Coef}(a) \leq \nu)\big)^N.\end{aligned}$$

The quantity $\text{Prob}(\text{Coef}(a) \leq \nu)$ may be computed as

$$\text{Prob}(\text{Coef}(a) \leq \nu) = \sum_{x \leq \nu} \text{Prob}(\text{Coef}(a) = x),$$

where the individual probabilities $\mathrm{Prob}(\mathrm{Coef}(a) = x)$ in the sum are given by (8), and finally

$$\mathrm{Prob}(\mathrm{Min}(a) = \nu) = \mathrm{Prob}(\mathrm{Min}(a) \le \nu) - \mathrm{Prob}(\mathrm{Min}(a) \le \nu - 1).$$

It only remains to give a formula for the probability distribution of the width of a. This is easy using the previously computed distributions for $\mathrm{Max}(a)$ and $\mathrm{Min}(a)$ and the definition $\mathrm{Width} = \mathrm{Max} - \mathrm{Min}$. Thus

$$\mathrm{Prob}(\mathrm{Width}(a) = w) = \sum_{\lambda} \mathrm{Prob}(\mathrm{Max}(a) = \lambda) \cdot \mathrm{Prob}(\mathrm{Min}(a) = \lambda - w).$$

3.4 Probability Distributions for $f = 1 + p * F$

As noted in Remark 2, it is often advantageous to select f to have the form $f = 1 + p * F$, since it eliminates one multiplication from the decryption process. When f has this form, the polynomial $a = p * r * g + f * m$ becomes

$$a = p * (r * g + F * m) + m.$$

If m is binary and p is an integer, then the max and min probabilities satisfy

$$\mathrm{Prob}(\mathrm{Max}(a) \ge \mu) \le \mathrm{Prob}\left(\mathrm{Max}(r * g + F * m) \ge \lfloor \mu/p \rfloor\right), \tag{9}$$

$$\mathrm{Prob}(\mathrm{Min}(a) \le \mu) \le \mathrm{Prob}\left(\mathrm{Min}(r * g + F * m) \le \lfloor (\mu + p - 1)/p \rfloor\right). \tag{10}$$

This allows us to calculate decryption failure probabilities when centering method `center2` is used.

Additionally, the width of a satisfies

$$\mathrm{Width}(a) \le p * \mathrm{Width}(r * g + F * m) + 1, \tag{11}$$

and hence

$$\mathrm{Prob}(\mathrm{Width}(a) \ge q) \le \mathrm{Prob}\left(\mathrm{Width}(r * g + F * m) \ge \frac{q-1}{p}\right). \tag{12}$$

If m is a random binary polynomial, then (11) will be an equality at least 25% of the time, so we lose very little in using the inequality (12) to estimate the probability of a given width. This allows us to calculate decryption failure probabilities when centering method `center1` is used.

Note that we do not need to derive new formulas to calculate the probability distribution of $\mathrm{Width}(r * g + F * m)$. Each of r, g, F, and m will be taken from the set of binary polyomials with a fixed number of ones, so the formulas that we derived earlier are the ones that we need, albeit with the letter f replaced by the letter F.

3.5 The independence assumption

The formula for the width in the previous section was derived assuming that if polynomials $A(X)$ and $B(X)$ are drawn at random from sets of binary polynomials $\hat{R}(d_A)$ and $\hat{R}(d_B)$, then the coefficients of the product $C(X) = A(X) * B(X)$ are independent. In fact, this is not exactly true. Intuitively, the fact that $C(1) = A(1)B(1) = d_A d_B$ is constant implies that if some coefficient of $C(X)$ is particularly large, then the others will be somewhat smaller than usual. Thus the coefficients of $C(X)$ should be anticorrelated. Indeed, it is a simple exercise to show that if i and j are distinct indices, then the correlation coefficient between C_i and C_j is

$$\mathrm{Corr}(C_i, C_j) = -\frac{1}{N-1}\,.$$

The effect of this should be that calculations of width probabilities based on assuming the independence of the coefficients of a will tend to underestimate the actual width probabilities, but if N is large, the effect will be small. In Section 4, we compare our theoretical results with the experimental ones to see if this is in fact the case.

4 Theory and experiment for a typical parameter set

We recall the parameter set `ees251ep4` from [1] described in Section 2.5:

$$(N, p, q) = (251, 2, 239), \qquad f = 1 + p * F,$$
$$F, g, r \in \hat{R}(72), \qquad m \in \hat{R}, \qquad \text{centering method: } \texttt{center2}.$$

In this section we will use our theoretical formulas to compute probabilities for max, min, and width for $a = p * r * g + f * m$, and then we will compare these values with the results of experiments.

As noted in Section 3.1, the width of a will be largest when m has an equal number of ones and zeros. This will also give a for which we expect the greatest difference between an individual coefficient and the average. We will thus consider the case where m is chosen from the set

$$m \in \hat{R}(125).$$

We set

$$A = r * g + F * m$$

and use the formulas from Section 3.1 to compute the probabilities

$$\mathrm{Prob}(\mathrm{Max}(A) \geq T), \qquad \mathrm{Prob}(\mathrm{Min}(A) \leq T), \qquad \mathrm{Prob}(\mathrm{Width}(A) \leq T).$$

Calculations were performed using Pari with 56 decimal places of precision.

As noted in Section 3.4, the probability of a decryption failure can be obtained using (9) for centering method `center2`, or using (12) for centering method `center1`. For the particular parameter set in question, the average value of a

coefficient of $prg + gfm$ is 113, and we have $p = 2$ and $q = 239$ and are using `center2`, so we are interested in the probability

$$\text{Prob}(\text{Max}(A) \geq \lfloor (113 + 239/2)/2 \rfloor) = \text{Prob}(\text{Max}(A) \geq 116) \ .$$

We do not need to check the corresponding probability for Min, because Min < 0 and no coefficient of $prg + fm$ can be less than 0. Looking at Table 2, we see that

$$\text{Prob}(\text{Fail}(\texttt{center2})) = \text{Prob}(\text{Max}(A) \geq 116) < 2^{-104.0} \ .$$

If we were to be using `center1`, our failure criterion would be

$$\text{Prob}(\text{Width}(A) \geq 119).$$

In this case, the probability of a decryption failure is even smaller. Indeed, Table 1 says that

$$\text{Prob}(\text{Width}(A) \geq 76) \approx 2^{-80.9} \quad \text{and} \quad \text{Prob}(\text{Width}(A) \geq 100) \approx 2^{-156.2}.$$

4.1 Comparison of theory and experiment

In this section we compare the results of experiments with our theoretical formulas. We consider polynomials of the following form:

$$N = 251, \quad p = 2, \quad f = 1 + pF, \quad m \in \hat{R}(125), \quad F, g, r \in \hat{R}(72).$$

We compare theoretical and experimental values for the width of

$$A = r * g + m * F.$$

The results are described in Table 4, and show good agreement.

We therefore conclude that for the recommended parameter sets, the chance of a decryption failure is less than 2^{-100}, considerably better than the desired level of 2^{-80}.

5 Further Notes

This section outlines areas of continuing research.

- It would be useful to perform additional experiments to gather more accurate data for extreme tail probabilities, and also to gather data for other parameter sets.
- It would be interesting to study and quantify the effect of the existing weak coefficient correlation on tail probabilities. Preliminary research, both theoretical and experimental, indicates that these effects are very small. However, more precise information could give greater assurance to the use of the `center1` centering method.

x	$\log_2(\text{Prob}(\text{Width}(A) = x))$	$\log_2(\text{Prob}(\text{Width}(A) \geq x))$
16	−28.63203	−0.00000
18	−17.50276	−0.00000
20	−10.25059	−0.00014
22	−5.87059	−0.00802
24	−3.57235	−0.09962
26	−2.73942	−0.46465
28	−2.91087	−1.23985
30	−3.76420	−2.42182
32	−5.08934	−3.94546
34	−6.75736	−5.74646
36	−8.69158	−7.77791
38	−10.84601	−10.00830
40	−13.19193	−12.41644
42	−15.71049	−14.98781
44	−18.38897	−17.71241
46	−21.21891	−20.58354
48	−24.19498	−23.59709
50	−27.31421	−26.75088
52	−30.57532	−30.04418
54	−33.97822	−33.47726
56	−37.52360	−37.05106
58	−41.21264	−40.76694
60	−45.04689	−44.62659
62	−49.02811	−48.63187
64	−53.15827	−52.78484
66	−57.43946	−57.08768
68	−61.87396	−61.54275
70	−66.46419	−66.15253
72	−71.21275	−70.91966
74	−76.12239	−75.84698
76	−81.19607	−80.93749
78	−86.43695	−86.19439
80	−91.84843	−91.62112
82	−97.43414	−97.22136
84	−103.19798	−102.99904
86	−109.14416	−108.95841
88	−115.27721	−115.10404
90	−121.60203	−121.44084
92	−128.12394	−127.97417
94	−134.84869	−134.70980
96	−141.78258	−141.65406
98	−148.93245	−148.81381
100	−156.30582	−156.19659

Table 1. Theoretical Probabilities for $A = r * g + F * m$, $[N, dF, dg, dm, dr] = [251, 72, 72, 125, 72]$

x	$\log_2(\text{Prob}(\text{Max}(A) = x))$	$\log_2(\text{Prob}(\text{Max}(A) \geq x))$
61	−59.31708	−0.00000
63	−28.04184	−0.00000
65	−11.74975	−0.00000
67	−4.49427	−0.00969
69	−2.26278	−0.28959
71	−2.57445	−1.34536
73	−4.04775	−3.08091
75	−6.06569	−5.23025
77	−8.40989	−7.66634
79	−11.01854	−10.35123
81	−13.87850	−13.27806
83	−16.98915	−16.44828
85	−20.35274	−19.86525
87	−23.97235	−23.53290
89	−27.85161	−27.45551
91	−31.99462	−31.63775
93	−36.40602	−36.08474
95	−41.09104	−40.80210
97	−46.05555	−45.79606
99	−51.30614	−51.07350
101	−56.85022	−56.64210
103	−62.69611	−62.51040
105	−68.85321	−68.68798
107	−75.33208	−75.18559
109	−82.14473	−82.01538
111	−89.30479	−89.19113
113	−96.82788	−96.72856
115	−104.73193	−104.64570
117	−113.03773	−112.96344
119	−121.76953	−121.70612
121	−130.95595	−130.90241
123	−140.63108	−140.58647
125	−150.83614	−150.79957

Table 2. Theoretical Probabilities for $A = r * g + F * m$, $[N, dF, dg, dm, dr] = [251, 72, 72, 125, 72]$

x	$\log_2(\text{Prob}(\text{Min}(A) = x))$	$\log_2(\text{Prob}(\text{Min}(A) \leq x))$
20	−40.65323	−43.18977
22	−35.46258	−37.76018
24	−30.65940	−32.72656
26	−26.22237	−28.06513
28	−22.13352	−23.75581
30	−18.37752	−19.78140
32	−14.94125	−16.12695
34	−11.81359	−12.77941
36	−8.98669	−9.72763
38	−6.46351	−6.96454
40	−4.29038	−4.49790
42	−2.66199	−2.38748
44	−2.17074	−0.82352
46	−4.19768	−0.09439
48	−11.26592	−0.00059
50	−27.47108	−0.00000
52	−58.86814	−0.00000
54	−112.70222	−0.00000

Table 3. Theoretical Probabilities for $A = r * g + F * m$, $[N, dF, dg, dm, dr] = [251, 72, 72, 125, 72]$

– Previous parameter sets for NTRUEncrypt have used a *product form* for the quantities f and r, as described in [3,4]. Theoretical probability calculations for polynomials of this type are much harder to do than for the more simply-structured binary polynomials investigated in this note. Advances in this area might allow considerable efficiency gains for NTRUEncrypt, and thus would be of great practical interest.

6 Acknowledgements

We would like to thank Florian Hess, Nigel Smart and Frederik Vercauteren for comments on a previous version of this note.

References

1. EESS: Consortium for Efficient Embedded Security. Efficient Embedded Security Standards #1: Implementation Aspects of NTRUEncrypt and NTRUSign. Version 2.0 available at `http://www.ceesstandards.org`, May 2003.
2. C. Gentry. Key Recovery and Message Attacks on NTRU-Composite. In *Eurocrypt '01*, LNCS 2045, pages 182–194. Springer-Verlag, Berlin, 2001.
3. J. Hoffstein, J. Pipher, J. H. Silverman. NTRU: A new high speed public key cryptosystem. Algorithmic Number Theory (ANTS III), Portland, OR, June 1998, Lecture Notes in Computer Science 1423, J.P. Buhler (ed.), Springer-Verlag, Berlin, 1998, 267–288

x	Prob(Width$(A) \geq x$)		
	Experimental	Theoretical	Ratio
20	99.931%	99.990%	1.00
21	99.447%	99.909%	1.00
22	97.806%	99.446%	0.98
23	93.757%	97.737%	0.96
24	85.678%	93.328%	0.92
25	73.662%	84.921%	0.87
26	58.116%	72.465%	0.80
27	42.869%	57.490%	0.75
28	29.504%	42.342%	0.70
29	18.621%	29.045%	0.64
30	11.313%	18.662%	0.61
31	6.520%	11.302%	0.58
32	3.701%	6.491%	0.57
33	1.780%	3.553%	0.50
34	0.840%	1.863%	0.45
35	0.460%	0.938%	0.49
36	0.230%	0.456%	0.50
37	0.120%	0.214%	0.56
38	0.060%	0.097%	0.62
39	0.020%	0.043%	0.47
40	0.010%	0.018%	0.55

Table 4. Probabilities for $A = r * g + F * m$, 10000 Samples, $[N, dF, dg, dm, dr] = [251, 72, 72, 125, 72]$

4. J. Hoffstein and J. H. Silverman. Optimizations for NTRU. In *Public-key Cryptography and Computational Number Theory.* DeGruyter, 2000. To appear, available at [9].
5. J. Hoffstein and J. H. Silverman. Random Small Hamming Weight Products With Applications To Cryptography. *Discrete Applied Mathematics.* To appear, available at [9].
6. N. Howgrave-Graham, P. Q. Nguyen, D. Pointcheval, J. Proos, J. H. Silverman, A. Singer, W. Whyte, The Impact of Decryption Failures on the Security of NTRU Encryption Proc. Crypto 2003 To appear, available at [9].
7. N. Howgrave-Graham, J. H. Silverman, A. Singer, W. Whyte, NAEP: Provable Security in the Presence of Decryption Failures Submitted, available at [9].
8. T. Meskanen and A. Renvall. Wrap Error Attack Against NTRUEncrypt. To appear in *Proc. of WCC '03.* Available from `http://www.tucs.fi/Research/Series/techreports/techrep.php?year=2003`.
9. NTRU Cryptosystems. Technical reports. Available at `http://www.ntru.com`, 2003.
10. J. Proos. Imperfect Decryption and an Attack on the NTRU Encryption Scheme. Cryptology ePrint Archive: Report 2003/002.
11. J. H. Silverman. NTRU Technical Report #11, version 2: Wraps, Gaps, and Lattice Constants 2001, available at [9].

Comments and questions concerning this technical report should be addressed to `techsupport@ntru.com`

Additional information concerning NTRU Cryptosystems and the NTRU Public Key Cryptosystem are available at `www.ntru.com`