

The NTRU lattice and related lattices: Recent progress and open questions

Jeffrey Hoffstein

Brown University, NTRU Cryptosystems

`<www.ntru.com>`

CAEN—June 1, 2005

Topics

Topics

- A quick overview of the NTRU lattice and NTRUSign
- Measuring the lattice security of key recovery
 - Advantages of the transpose lattice
- Measuring the lattice security of signature generation
 - Advantages of modifying the norm definition
- Transcript security and perturbations
 - Some open questions on perturbations
- A potential modification of the NTRUSign lattice
- Low gate count storage and distribution of the public key
- Low gate count computation of the NTRUSign private lattice

Basic setup

NTRU is best described using the ring of polynomials

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

These are polynomials with integer coefficients

$$a(X) = a_0 + a_1X + a_2X^2 + \cdots + a_{N-1}X^{N-1}$$

that are multiplied using the extra rule $X^N = 1$.

$$c(X) = a(X) * b(X)$$

with

$$c_k = a_0b_k + a_1b_{k-1} + \cdots + a_{N-1}b_{k+1} = \sum_{i+j \equiv k \pmod{N}} a_i b_j.$$

Alternative: If we write $a(X)$, $b(X)$, and $c(X)$ as vectors

$$\mathbf{a} = [a_0, \dots, a_{N-1}], \quad \mathbf{b} = [b_0, \dots, b_{N-1}], \quad \mathbf{c} = [c_0, \dots, c_{N-1}],$$

then $\mathbf{c} = \mathbf{a} * \mathbf{b}$ is the usual *convolution product*.

Basic setup

A natural measure of size in R is the centered Euclidean norm (essentially the variance) of the vector of coefficients. Thus we write $\bar{r} = (1/N) \sum_{i=0}^{N-1} r_i$ for the average of the coefficients and define the *centered* norm by the formula

$$\|r\|^2 = \sum_{i=0}^{N-1} (r_i - \bar{r})^2 = \sum_{i=0}^{N-1} r_i^2 - N\bar{r}^2.$$

If $r \in R$ satisfies $\|r\|^2 = \mathcal{O}(N)$, we will say that r is *short*. The centered norm possesses the attractive pseudo-multiplicative property $\|r * s\| \approx \|r\| \cdot \|s\|$ for most choices of short $r, s \in R$.

Basic setup

A particularly useful class of short polynomials is described in the following way.

For a given positive integer d , the space $\mathcal{T}(d)$ is defined to be the set of all $r \in R$ such that $d + 1$ coefficients of r are equal to 1, d coefficients of r are equal to -1 , and the remaining coefficients are equal to 0. Thus for $r \in \mathcal{T}(d)$, $\|r\|^2 \approx 2d$. These are called ternary polynomials. The size of the space is given by

$$|\mathcal{T}(d)| = \binom{N}{d+1} \binom{N-d-1}{d}.$$

Thus, for example, if $d \approx N/3$ then $|\mathcal{T}(d)| \approx (3.57)^N$.

For $f \in \mathcal{T}(d)$,

$$\|f\| \approx \sqrt{2d} \approx \sqrt{\delta N}.$$

For convenience we have introduced the notation $\delta = 2d/N$.

Basic setup

Given any positive integers N and q and any polynomial $h \in R$, we construct a lattice L_h contained in $R^2 \cong \mathbb{Z}^{2N}$ as follows:

$$L_h = L_h(N, q) = \{(r, r') \in R \times R \mid r' \equiv r * h \pmod{q}\}.$$

This sublattice of \mathbb{Z}^{2N} is called a *convolution modular lattice*. It has dimension equal to $2N$ and determinant equal to q^N . It can be thought of as the lattice generated by the rows of:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & | & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & 1 & \cdots & 0 & | & h_1 & h_2 & \cdots & h_0 \\ \vdots & \vdots & \ddots & \vdots & | & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & | & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \hline 0 & 0 & \cdots & 0 & | & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & | & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & | & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & | & 0 & 0 & \cdots & q \end{pmatrix}$$

Basic setup

L_h can be represented conveniently by

$$L_h = \begin{pmatrix} 1 & h \\ 0 & q \end{pmatrix}.$$

Here each entry represents the corresponding N by N circulant matrix.

The centered norm $\| \cdot \| : R \rightarrow \mathbb{R}$ can be naturally extended to L_h as follows. For $(r, r') \in L_h(N, q)$, we set

$$\|(r, r')\| = \min_{k_1, k_2 \in R} (\|r + k_1 q\|^2 + \|r' + k_2 q\|^2)^{1/2}.$$

The basis for L_h is not particularly short, but for certain choices of h a particularly short basis will exist.

Basic setup

Choose $f, g \in \mathcal{T}(d)$ such that f and g are invertible modulo q , i.e. so that there are polynomials $f^{-1}, g^{-1} \in R$ satisfying $f * f^{-1} \equiv g * g^{-1} \equiv 1 \pmod{q}$.

Next we find polynomials $F, G \in R$ satisfying $f * G - g * F = q$. A method for accomplishing this is described in the first NTRUSign paper. If F and G are constructed using this method then they will satisfy

$$\|F\| \approx \|G\| \approx \|f\| \sqrt{N/12} \approx N \sqrt{\delta/12}.$$

Having found the 4-tuple (f, g, F, G) , we set

$$h \equiv f^{-1} * F \equiv g^{-1} * G \pmod{q}.$$

Observe that by construction there exist $k_1, k_2 \in R$ such that

$$f * h = F + k_1 q \quad \text{and} \quad g * h = G + k_2 q.$$

Basic setup

The lattice L_h is called an NTRUSign lattice, the polynomial h is called the *public key*, and the pair (f, g) is called the *private key*. The importance of this lattice is that it can be described by two distinctly different bases. There is the original, public, basis, but also

$$\begin{pmatrix} f & F \\ g & G \end{pmatrix},$$

the private basis. The two bases are related by

$$\begin{pmatrix} f & -k_1 \\ g & -k_2 \end{pmatrix} \begin{pmatrix} 1 & h \\ 0 & q \end{pmatrix} = \begin{pmatrix} f & F \\ g & G \end{pmatrix}.$$

This is the *transpose* of the matrix originally used to describe an NTRU lattice.

Basic setup

The signature algorithm takes as input a digital document D and the private key (f, g, F, G) and outputs a signature s . The recipient verifies the signature by checking that

$$\|(s, s * h - m(D))\| \leq \mathcal{N},$$

where $m(D) \in R$ is a *message representative* derived by hashing D and \mathcal{N} is a pre-specified *norm bound*.

In the transpose lattice, the norm of s is typically smaller than the norm of $s * h - m$ by a factor of $\sqrt{12/N}$. It is therefore useful to generalize the norm $\|(r, r')\|$ to include a *balancing factor* $\beta > 0$, which leads to the definition

$$\|(r, r')\|_{\beta} = \min_{k_1, k_2 \in R} \left(\|r + k_1 q\|^2 + \beta^2 \|r' + k_2 q\|^2 \right)^{1/2} .$$

Basic setup

Verification then consists of checking that

$$\|(s, s * h - m(D))\|_{\beta} \leq \mathcal{N}.$$

One way to interpret the β -norm $\|(r, r')\|_{\beta}$ is as the usual norm of the point $(r, \beta r')$ in the lattice

$$L_h(\beta) = L_h(N, q, \beta) = \{(r, \beta r') \mid r \in R \text{ and } r' \equiv r * h \pmod{q}\} .$$

The objectives are to choose \mathcal{N}

- Large enough that possession of the private key allows s to be constructed
- Small enough that lack of the private key makes it highly improbable that s can be constructed.

Key security

The first question we address is: given a public key h , how difficult is it to recover f, g, F, G ? Recall

$$h \equiv f^{-1} * F \equiv g^{-1} * G \pmod{q}.$$

First of all, one can do a brute force meet in the middle attack on the space $\mathcal{T}(d)$. The bit security of this attack is $\approx \sqrt{|\mathcal{T}(d)|}$. This is at the present moment the most efficient known method for extracting f from h (for most practical choices of parameters).

Lattice reduction methods, particularly the KZ-block method, can be stronger in low dimensions. As $\|F\|$ is larger than $\|f\|$ by a factor of $\sqrt{N/12}$, key recovery by these methods is harder than in the standard NTRU lattice. This is because the difficulty of applying the KZ-block method to NTRU key recovery can be quantified by a certain lattice constant.

Key security

One can define σ , the length of the expected shortest vector in the lattice L_h of dimension $2N$ and determinant q^N , by the relation

$$\sigma^{2N} V_{2N} = q^N.$$

Here $V_{2N} = \pi^N / \Gamma(N + 1)$ is the volume of a unit $2N$ -sphere. If the shortest vector in L_h has length $\tau < \sigma$ then one defines c by:

$$\tau = c\sigma / \sqrt{2N}.$$

With high probability the shortest vector in L_h is (f, F) , and after “balancing” its effective length becomes $\sqrt{2\|f\|\|F\|}$. This leads to

$$c = N^{1/4} \sqrt{2\pi e \delta a / \sqrt{3}},$$

where $a = N/q$ is generally taken to be constant as we move through an sequence of dimensions N and associated q .

Key security

The point here is that c increases as N increases, and increasing c leads to considerably greater breaking times. In particular for a fixed c , and $a = N/q$, as N increases

$$\log T \geq A(c, a)N + B(c, a),$$

where T denotes the time necessary to recover (f, F) via KZ-block reduction, and $A(c, a)$ increases for fixed a and increasing c . For example, we have the following table of experimental results:

bound for c	$A(c, a)$	$B(c, a)$
$c > 3.7$	0.451	-0.218
$c > 5.3$	0.649	-5.436
$c > 6.8$	1.539	-102.59

Table 1. Constants used to calculate bit security against lattice key attacks, based on experimental evidence for different values of c

Signing

For any $a \in \mathbb{Q}$, let $\lfloor a \rceil$ denote the integer closest to a , and define $\{a\} = a - \lfloor a \rceil$. If A is a polynomial with rational (or real) coefficients, let $\lfloor A \rceil$ and $\{A\}$ be A with rounded coefficients.

Suppose we are given a point $(0, m)$, where m is the image of some digital document \mathcal{D} under the hash function H . Our object is to find a point $(s, t) \in L_h(\beta)$ such that $\|s\|^2 + \beta^2 \|t - m\|^2$ is as small as possible. This is accomplished by the following process. Solve for real (x, y) satisfying

$$(0, m) = (x, y) \begin{pmatrix} f & F \\ g & G \end{pmatrix}$$

by writing

$$(x, y) = (0, m) \begin{pmatrix} G & -F \\ -g & f \end{pmatrix} / q = (-m * g/q, m * f/q).$$

Signing

Define ϵ and ϵ' with rational coefficients varying uniformly between $-1/2$ and $1/2$ by the formulas

$$\lfloor x \rfloor = x + \epsilon \quad \text{and} \quad \lfloor y \rfloor = y + \epsilon'.$$

Thus

$$\epsilon = -\{x\} = \{m * g/q\} \quad \text{and} \quad \epsilon' = -\{y\} = -\{m * f/q\}.$$

Note that \mathcal{E}_ϵ , the expected size of $\|\epsilon\|$, equals the expected size of $\|\epsilon'\|$ and

$$\mathcal{E}_\epsilon = \sqrt{N/12}.$$

Letting

$$(s, t) = (\lfloor x \rfloor, \lfloor y \rfloor) \begin{pmatrix} f & F \\ g & G \end{pmatrix}$$

we then obtain

$$(s, t - m) = (\epsilon f + \epsilon' g, \epsilon F + \epsilon' G).$$

Forgery security

We are now in a position to measure the expected size of a signature. First, we have

$$\|s\| \approx \sqrt{\|\epsilon_1\|^2 \|f\|^2 + \|\epsilon_2\|^2 \|g\|^2}.$$

Thus

$$\mathcal{E}_s = \sqrt{\frac{N^2 \delta}{6}}.$$

Similarly,

$$\mathcal{E}_t = \sqrt{\frac{N^3 \delta}{72}}.$$

and thus the expected signature size, with the parameter β is

$$\mathcal{E} = \sqrt{\mathcal{E}_s^2 + \beta^2 \mathcal{E}_t^2} = \sqrt{\frac{N^2 \delta}{6}} \sqrt{1 + \beta^2 N/12}.$$

We then set the norm bound \mathcal{N} to $\mathcal{N} = \rho \mathcal{E}$, where ρ is slightly larger than 1. This ensures that most valid signatures will pass the norm bound test.

Forgery security

Given a hash of a digital document m , a person without knowledge of the private key can attempt to produce a valid signature s on m by various methods. These can be by combinatorics, or by lattice reduction methods, or by some combination of both.

The combinatorial approach would begin with a random choice of a small s , and a hope that $\|h * s - m\|$ is sufficiently small. The probability of success of this approach can be bounded above by

$$\frac{V_N \mathcal{N}^N}{(q\beta)^N}.$$

In fact, it is reasonable to assume that a meet in the middle attack can be applied to this, which would square root this probability.

As β decreases, the probability of success by this method increases.

Forgery security

The difficulty of creating a forgery via lattice reduction can be quantified as follows. An adversary can attempt to use lattice reduction methods to locate a point $(s, \beta t) \in L_h(\beta)$ sufficiently close to $(0, \beta m)$ that $\|(s, \beta(t - m))\| < \mathcal{N}$. We'll refer to $\|(s, \beta(t - m))\|$ as the norm of the intended forgery.

The difficulty of using lattice reduction methods to accomplish this can be tied to another important lattice constant defined by:

$$\mathcal{N} = \gamma\sqrt{2N}\sigma$$

This is the ratio of the required norm of the intended forgery over the norm of the expected smallest vector of $L_h(\beta)$, scaled by $\sqrt{2N}$. For usual NTRUSign parameters the ratio, $\gamma\sqrt{2N}$, will be larger than 1. Thus with high probability there will exist many points of $L_h(\beta)$ that will work as forgeries.

Forgery security

The task of an adversary is to find one of these without the advantage that knowledge of the private key gives. As γ decreases and the ratio approaches 1 this becomes measurably harder. It appears that decreasing γ leads to considerably greater breaking times. In particular for a fixed γ , and $a = N/q$, if T denotes breaking time then as N increases

$$\log T \geq C(\gamma, a)N + D(\gamma, a),$$

and $C(\gamma, a)$ increases for fixed a as gamma decreases and increases for fixed γ as a decreases. The following table has been compiled from numerous experiments.

Forgery security

γ bound	N/q bound	$\omega_{lf}(N)$
$\gamma < 0.1774$	$N/q < 1.305$	$0.995113N - 82.6612$
$\gamma < 0.1413$	$N/q < 0.707$	$1.16536N - 78.4659$
$\gamma < 0.1400$	$N/q < 0.824$	$1.14133N - 76.9158$

Table 2. Bit security against lattice forgery attacks, ω_{lf} , based on experimental evidence for different values of $(\gamma, N/q)$

Forgery security

Our analysis shows that lattice strength against forgery is maximized, for a fixed N/q , when $\gamma(N, q, \beta)$ is as small as possible. With the notation we have established

$$\gamma(N, q, \beta) = \rho \sqrt{\frac{\pi e}{2N^2 q} \cdot (\mathcal{E}_s^2 / \beta + \beta \mathcal{E}_t^2)}$$

and so clearly the value for β which minimizes γ is $\beta = \mathcal{E}_s / \mathcal{E}_t$. This optimal choice yields

$$\gamma = \rho \sqrt{\frac{\pi e \mathcal{E}_s \mathcal{E}_t}{N^2 q}}.$$

We previously noted that increasing β has the effect of improving combinatorial forgery security. Thus the optimal choice will be the minimal $\beta \geq \mathcal{E}_s / \mathcal{E}_t$ such that the probability of forgery by combinatorial means is sufficiently small.

Forgery security

Setting $\beta = \mathcal{E}_s / \mathcal{E}_t = \sqrt{12/N}$ we obtain an optimal value for γ of

$$\gamma = \rho N^{-1/4} \sqrt{\frac{\pi e \delta a}{3\sqrt{12}}}.$$

As N increases, γ decreases, improving the resistance against lattice forgery.

Transcript analysis

An adversary studying a long transcript of valid signatures will have at his disposal a long list of pairs of polynomials of the form

$$s = \epsilon * f + \epsilon' * g$$

and

$$t - m = \epsilon * F + \epsilon' * G.$$

Here

$$\epsilon = \left\{ \frac{m * g}{q} \right\}, \quad \epsilon' = - \left\{ \frac{m * f}{q} \right\}.$$

Let $a(X) = \sum a_i X^i \in R$ be a polynomial. The *reversal* of a is the polynomial

$$\bar{a}(X) = a(X^{-1}) = a_0 + \sum_{i=1}^{N-1} a_{N-i} X^i.$$

Transcript analysis

We then set

$$\hat{a}(X) = a(X) * \bar{a}(X).$$

Notice that \hat{a} has the form

$$\hat{a} = \sum_{k=0}^{N-1} \left(\sum_{i=0}^{N-1} a_i a_{i+k} \right) X^k.$$

The expectation of \hat{s} and $\hat{t} - \hat{m}$ is (up to lower order terms)

$$E(\hat{s}) = (N/12)(\hat{f} + \hat{g})$$

and

$$E(\hat{t} - \hat{m}) = (N/12)(\hat{F} + \hat{G})$$

This is because the cross terms are uncorrelated and vanish.

We refer to these as the second moments. If these second moments could be recovered then the problem of recovering the private key would reduce to the (possibly easier) problem of factoring a Gram matrix $U^T U$.

Transcript analysis

Given a sufficiently long transcript of signatures, it is possible that an adversary could compute a good approximation to $E(\hat{s})$ and $E(\hat{t} - \hat{m})$. Because of this danger, it is desirable to perturb the original point $(0, m)$ before signing, by adding a small vector (δ_s, δ_t) . The effect of this will be to produce a transcript that resembles

$$s = \epsilon * f + \epsilon' * g + \delta_s$$

and

$$t - m = \epsilon * F + \epsilon' * G + \delta_t.$$

This would eventually yield

$$E(\hat{s}) = (N/12)(\hat{f} + \hat{g}) + E(\hat{\delta}_s)$$

and

$$E(\hat{t} - \hat{m}) = (N/12)(\hat{F} + \hat{G}) + E(\hat{\delta}_t).$$

Transcript analysis

Now if $E(\hat{\delta}_s) = 0$, or $E(\hat{\delta}_t) = 0$, or even if $E(\delta_s * \delta_t) = 0$ this would accomplish nothing. However, if this is not the case, then recovering the private key from a long transcript of signatures appears to become a much harder problem. This leaves us with an interesting open question:

- Find a simple form for a perturbation that is very difficult to remove.

One approach is to have a completely secret basis f_1, g_1, F_1, G_1 where the public key h_1 is *not* revealed. The procedure is:

- Sign $(0, m)$ with secret basis, obtaining (s_1, t_1) .
- Now sign (s_1, t_1) using the original basis.
- Result is

$$s = \epsilon f + \epsilon' g + \epsilon_1 f_1 + \epsilon'_1 g_1$$

and

$$t - m = \epsilon F + \epsilon' G + \epsilon_1 F_1 + \epsilon'_1 G_1.$$

Transcript analysis

The output of a long transcript analysis would then be

$$E(\hat{s}) = (N/12)(\hat{f} + \hat{g} + \hat{f}_1 + \hat{g}_1)$$

and

$$E(\hat{t} - \hat{m}) = (N/12)(\hat{F} + \hat{G} + \hat{F}_1 + \hat{G}_1).$$

The question is: can one eliminate the extra variables and reduce to the possibly easier Gram matrix problem. The answer is: maybe, but one would at the very least need to be able to compute up to the sixth moment, requiring considerably longer transcripts to obtain a good enough approximation.

Open questions on perturbations

- How large does the search space need to be for the perturbation basis or bases? Is there a meet in the middle attack?
- If two perturbation bases are used, can one be peeled off?
- Can the perturbation bases be simpler than the original basis?

Tables

k	N	d	q	β	\mathcal{N}
80	157	29	256	0.38407	150.02
112	197	28	256	0.51492	206.91
128	223	32	256	0.65515	277.52
160	263	45	512	0.31583	276.53
192	313	50	512	0.40600	384.41
256	349	75	512	0.18543	368.62

Table 3. Parameters for trinary keys,
one perturbation, $\rho = 1.1$, $q = \text{power of } 2$

Tables

ω_{cmb}	c	ω_{lk}	ω_{frg}	γ	ω_{lf}	$\log_2(\tau)$
104.43	5.34	93.319	80	0.139	102.27	31.9
112.71	5.55	117.71	112	0.142	113.38	31.2
128.63	6.11	134.5	128	0.164	139.25	32.2
169.2	5.33	161.31	160	0.108	228.02	34.9
193.87	5.86	193.22	192	0.119	280.32	35.6
256.48	7.37	426.19	744	0.125	328.24	38.9

Table 4. Relevant security measures for trinary keys, one perturbation, $\rho = 1.1$, $q =$ power of 2

Tables

k	N	d	q	NTRU	ECC	RSA
80	157	29	256	1256	192	1024
112	197	28	256	1576	224	~ 2048
128	223	32	256	1784	256	3072
160	263	45	512	2367	320	4096
192	313	50	512	2817	384	7680
256	349	75	512	3141	512	15360

Parameters b_{pk}

Table 5. Performance measures for the recommended parameter sets

Tables

NTRU	ECDSA	Gain
61073	112210	1.84
82937	170356	2.05
106817	277280	2.60
163849	—	—
233169	936618	4.20
331201	1595434	4.82

σ_S

NTRU	ECDSA	Gain
24649	130912	5.31
38809	198749	5.12
49729	323493	6.51
69169	—	—
97969	1092721	11.15
121801	1861340	15.28

σ_V

d/N	N/k
0.185	1.963
0.142	1.759
0.143	1.742
0.131	1.644
0.159	1.630
0.215	1.363

other

Table 5 (continued)

Performance measures for the recommended parameter sets

Tables

k	N	d	q	β	\mathcal{N}	c	γ
80	127	31	256	0.37264	122.94	5.33	0.133
112	191	29	256	0.45615	176.14	5.60	0.127
128	223	32	256	0.65515	277.52	6.11	0.164
160	263	45	512	0.31583	276.53	5.33	0.108
192	313	50	512	0.40600	384.41	5.86	0.119
256	349	75	512	0.18543	368.62	7.37	0.125

Table 6. Trinary keys, one perturbation, $\rho = 1$, $q =$ power of 2