Transcript secure signatures based on modular lattices

Version 2, revised 2016-04-15

Jeff Hoffstein[1], Jill Pipher[1], John M. Schanck[2,3], Joseph H. Silverman[1], and William Whyte[3]

[1] Brown University, Providence, USA
{jhoff,jpipher,jhs}@math.brown.edu
[2] University of Waterloo, Waterloo, Canada
[3] Security Innovation, Wilmington, USA
{jschanck,wwhyte}@securityinnovation.com

Abstract. We introduce a class of lattice-based digital signature schemes based on modular properties of the coordinates of lattice vectors. We also suggest a method of making such schemes transcript secure via a rejection sampling technique of Lyubashevsky (2009). A particular instantiation of this approach is given, using NTRU lattices. Although the scheme is not supported by a formal security reduction, we present arguments for its security and derive concrete parameters (first version) based on the performance of state-of-the-art lattice reduction and enumeration techniques. In the revision, we re-evaluate the security of first version of the parameter sets, under the hybrid approach of lattice reduction attack the meet-in-the-middle attack. We present new sets of parameters that are robust against this attack, as well as all previous known attacks.

1 Introduction

In the GGH and NTRUSign signature schemes [8, Sections 7.4,7.5] a document to be signed is thought of as a point $\boldsymbol{m}$ in $\mathbb{Z}^n$. A lattice L has a private basis, known only to the signer, that is reasonably short and close to orthogonal. The signer uses the private basis to solve a CVP and locate a point $\boldsymbol{s} \in L$ that lies reasonably close to $\boldsymbol{m}$. A verifier of the signature checks that $\boldsymbol{s}$ is indeed a point in the lattice L, and that the Euclidean distance between $\boldsymbol{s}$ and $\boldsymbol{m}$ is shorter than some pre-specified bound. The security assumption underlying the acceptance of the signature is that it is hard to find a point in L that is close to $\boldsymbol{m}$ unless one knows the private short basis for L.

A major difficulty with these signature schemes is the fact that when the private basis is used to locate $\boldsymbol{s}$, the difference $\boldsymbol{s} - \boldsymbol{m}$ has the form

$$\boldsymbol{s} - \boldsymbol{m} = \sum_{i=1}^{n} \epsilon_i \boldsymbol{v}_i,$$

where $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$ is the private basis and where each $|\epsilon_i| \leq 1/2$. Thus $\boldsymbol{s} - \boldsymbol{m}$ is a point in the interior of the fundamental parallelepiped associated to the private basis. If the signature is obtained by, say, Babai's rounding approach, the ϵ_i will be randomly and uniformly distributed in the interval $(-1/2, 1/2)$. A long transcript of signatures then corresponds to a large collection of points randomly and uniformly distributed inside the parallelepiped, and a sufficiently long transcript eventually reveals the vertices of the parallelepiped, and the secret basis. This was demonstrated successfully in [15, 16, 3].

It has been proposed that such an attack could be thwarted by carefully signing in such a way that the distribution of the ϵ_i was controlled, and it was proved that using such methods it is possible to construct signing protocols where the transcript contains no information pertaining to the private basis [4]. While effective at preventing information leakage, this process of controlling the distribution of the ϵ_i is computationally expensive.

The present work introduces a similar technique which does not require sampling complicated distributions.

Very roughly, the idea is as follows. Fix a public small prime p, and, rather than taking $\boldsymbol{m}$ to be a point in $\mathbb{Z}^n$, consider it instead to be a point $\boldsymbol{m}_p \in (\mathbb{Z}/p\mathbb{Z})^n$. Fix also a specific public region $\mathcal{R}$ in $\mathbb{Z}^n$. The region $\mathcal{R}$ should be sufficiently large that the volume of $\mathcal{R}$, which we denote by $|\mathcal{R}|$, satisfies

$$\frac{|\mathcal{R}|}{p^n} > C^n,$$

for a sufficiently large C. Precise examples will be given below. A signature on $\boldsymbol{m}_p$ is a point $\boldsymbol{s} \in L \cap \mathcal{R}$, with $\boldsymbol{s} \equiv \boldsymbol{m}_p \pmod{p}$.

Signing is accomplished as follows. To sign $\boldsymbol{m}_p \in (\mathbb{Z}/p\mathbb{Z})^n$, a random point $\boldsymbol{s}_0 \in L \cap \mathcal{R}$ is chosen. Let M be a matrix whose rows are the private basis, and let M_p be the reduction of this basis modulo p. Use M_p to find $\boldsymbol{v}_p \in (\mathbb{Z}/p\mathbb{Z})^n$ such that

$$\boldsymbol{s}_0 + \boldsymbol{v}_p \cdot M_p \equiv \boldsymbol{m}_p \pmod{p}.$$

Let $\boldsymbol{v}$ be the lift of $\boldsymbol{v}_p$ to $\mathbb{Z}^n$ with coefficients chosen from the interval $(-p/2, p/2)$. Then as M is a short basis and p is small, the vector $\boldsymbol{v} \cdot M$ will also be short, and $\boldsymbol{s} = \boldsymbol{s}_0 + \boldsymbol{v} \cdot M$ will satisfy $\boldsymbol{s} \equiv \boldsymbol{m}_p \pmod{p}$. Also, as $\boldsymbol{s}_0$ was chosen to lie in $L \cap \mathcal{R}$, and $\boldsymbol{v} \cdot M$ is short, there is a reasonable chance that $\boldsymbol{s}$ will also lie in $L \cap \mathcal{R}$. The algorithm of choosing $\boldsymbol{s}_0$ and solving for $\boldsymbol{s}$ is repeated until $\boldsymbol{s} \in L \cap \mathcal{R}$.

Any lattice point $\boldsymbol{s}$ satisfying $\boldsymbol{s} \equiv \boldsymbol{m}_p \pmod{p}$ is a valid signature, and such points will be well distributed throughout $\mathcal{R}$. Anyone can use a public basis to find a point in L with the desired properties modulo p, and if $\mathcal{R}$ is sufficiently large it is easy, using a short basis, to find points of $L \cap \mathcal{R}$, but if one does not know a short basis, then it is hard to satisfy both criteria simultaneously.

To create a collection of $\boldsymbol{s} - \boldsymbol{s}_0$, an attacker must also locate the nearby lattice point $\boldsymbol{s}_0$, However, for any $\boldsymbol{s} \in L \cap \mathcal{R}$, there will be many potential $\boldsymbol{s}_0'$ that are close to $\boldsymbol{s}$. In fact, if it is not only required that $\boldsymbol{s} \in L \cap \mathcal{R}$, but also that $\boldsymbol{s}$ lies at least a certain distance inside the boundary of $\mathcal{R}$, then it can be shown that with equal probability any s_0' within a fixed radius of $\boldsymbol{s}$ could have been the actual

$\boldsymbol{s}_0$ used in the signing process. This idea can be used to give a proof that the transcript contains no information about the private basis. This aspect of the approach is inspired by a rejection sampling technique of Lyubashevsky [11–13].

Another contribution of this paper is a particular, efficient, instantiation of this idea using NTRU lattices. We make this choice for two reasons. First, there is a natural dimension doubling: the dimension is $n = 2N$, where N is the number of coordinates needed to determine a point. Second, the lattice can be sufficiently well described using only half of a complete basis, and this half can be made quite short and sufficiently orthogonal. We will refer to this new signature scheme as an *NTRU Modular Lattice Signature Scheme*, or NTRUMLS for short.

1.1 Update on this revision

In this revision, we revisit the security of the proposed parameter sets of the NTRUMLS scheme. The re-evaluation of the security is based on an analysis of a hybrid attack on NTRU lattices in [5] and [6]. See Section 5.2 for more details.

We also propose a new set of parameters that are immune to the above cryptanalysis. For a given security level, we increased the dimension of the NTRU lattice, while reducing the modulus q. This gives us a very low acceptance rate (between 2% to 8%) during the rejection sampling phase. This drawback can be mitigated with parallel computation, such as [1].

2 Description of NTRUMLS

2.1 Notation

We work in the ring

$$\mathcal{R} = \mathcal{R}_N = \frac{\mathbb{Z}[x]}{\langle x^N - 1\rangle}.$$

We implicitly identify each element of $\mathcal{R}$ with the unique polynomial of degree less than N in its congruence class. Having done this, we identify a polynomial with its vector of coefficients in $\mathbb{Z}^N$. Writing an element $\boldsymbol{f} \in \mathcal{R}$ as

$$\boldsymbol{f} = \sum_{i=0}^{N-1} a_i x^i,$$

we set

$$\|\boldsymbol{f}\| = \max_{0 \le i < N} |a_i|,$$

and we define the restriction of $\mathcal{R}$ to the max-norm ball of radius k as

$$\mathcal{R}(k) = \{\boldsymbol{f} \in \mathcal{R} : \|\boldsymbol{f}\| \le k\}.$$

So that, for example, $\mathcal{R}(3/2)$ is the set of trinary polynomials.

We will frequently work in the quotient ring $\mathcal{R}/q\mathcal{R}$ with $q \in \mathbb{Z}$. We set the convention that when lifting an element of $\mathcal{R}/q\mathcal{R}$ to $\mathcal{R}(q/2)$ the lifted coefficients are chosen to satisfy $-q/2 \le a_i < q/2$ when q is even, and $-\lfloor q/2 \rfloor \le a_i \le \lfloor q/2 \rfloor$ when q is odd.

2.2 System Parameters

N	dimension parameter
p	a small odd prime
q	an integer larger, and relatively prime to, p
B_s, B_t	norm constraints

The B_s and B_t parameters serve primarily to fine tune the balance between security and performance. Reducing B_s and B_t may, for instance, allow one to choose a smaller q, but this may come at the expense of making it difficult for an honest party to compute a signature. Typical values of B_s and B_t satisfy $B_s = pB_t$, and

$$\|\boldsymbol{a} * \boldsymbol{b}\| \leq B_t \quad \text{for all } \boldsymbol{a}, \boldsymbol{b} \in \mathcal{R}\left(\frac{p}{2}\right).$$

Smaller B_s and B_t may be used provided that the signer performs an additional check during signature generation.

There will be further conditions on (N, p, q) to prevent search and lattice attacks, while still making it possible to find valid signatures; see Sections 4 and 5 for details.

2.3 Private Key

Choose polynomials

$$\boldsymbol{f} \xleftarrow{\$} p\mathcal{R}(3/2) \qquad \text{and} \qquad \boldsymbol{g} \xleftarrow{\$} \mathcal{R}(p/2).$$

Writing $\boldsymbol{f} = p\boldsymbol{F}$, so $\boldsymbol{F}$ is trinary, check that both $\boldsymbol{g}$ and $\boldsymbol{F}$ are invertible modulo q and modulo p. Sample a new pair if they are not. (We remark that the probability of $\boldsymbol{g}$ and $\boldsymbol{F}$ being invertible is quite high if $(x^N - 1)/(x - 1)$ does not have low degree factors when reduced modulo p and q.)

The private signing key is the pair $(\boldsymbol{f}, \boldsymbol{g})$.

2.4 Public Key

The public verification key is the polynomial

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

Also let

$$L_{\boldsymbol{h}} = \left\{(\boldsymbol{s}, \boldsymbol{t}) \in \mathcal{R}^2 : \boldsymbol{t} \equiv \boldsymbol{h} * \boldsymbol{s} \pmod{q}\right\}$$

be the usual NTRU lattice associated to $\boldsymbol{h}$.

We will often consider subsets of $L_{\boldsymbol{h}}$ consisting of vectors of bounded norm. This will be denoted by

$$L_{\boldsymbol{h}}(k_1, k_2) = L_{\boldsymbol{h}} \cap \left(\mathcal{R}(k_1) \times \mathcal{R}(k_2)\right).$$

2.5 Document Hashes and Valid Signatures

A document hash is a $2N$-vector

$$(\boldsymbol{s}_p, \boldsymbol{t}_p) \in \mathcal{R}(p/2) \times \mathcal{R}(p/2),$$

i.e.,

$$\big\|(\boldsymbol{s}_p, \boldsymbol{t}_p)\big\| = \max\big\{\|\boldsymbol{s}_p\|, \|\boldsymbol{t}_p\|\big\} \leq p/2.$$

We fix a hash function

$$\mathsf{Hash} : \mathcal{R}(q/2) \times \{0,1\}^* \longrightarrow \mathcal{R}(p/2) \times \mathcal{R}(p/2).$$

A valid signature on the document hash $(\boldsymbol{s}_p, \boldsymbol{t}_p)$ for the signing key $\boldsymbol{h}$ is a $2N$-vector $(\boldsymbol{s}, \boldsymbol{t}) \in \mathcal{R}^2$ satisfying:

(a) $(\boldsymbol{s}, \boldsymbol{t}) \in L_{\boldsymbol{h}}\Big(\dfrac{q}{2} - B_s, \dfrac{q}{2} - B_t\Big)$.
(b) $(\boldsymbol{s}, \boldsymbol{t}) \equiv (\boldsymbol{s}_p, \boldsymbol{t}_p) \pmod{p}$.

2.6 Algorithms

Algorithm 1 NTRUMLS Signature Algorithm

```
Input: $(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}, \mu)$, where $(\boldsymbol{f}, \boldsymbol{g})$ is a private key, $\boldsymbol{h}$ is the corresponding public key,
    and $\mu \in \{0,1\}^*$ is a document to be signed.
1: $(\boldsymbol{s}_p, \boldsymbol{t}_p) \longleftarrow \mathsf{Hash}(\boldsymbol{h}, \mu)$
2: repeat
3:     $\boldsymbol{r} \xleftarrow{\$} \mathcal{R}\left(\left\lfloor \frac{q}{2p} + \frac{1}{2} \right\rfloor\right)$
4:     $\boldsymbol{s}_0 \longleftarrow \boldsymbol{s}_p + p\boldsymbol{r}$
5:     $\boldsymbol{t}_0 \longleftarrow \boldsymbol{h} * \boldsymbol{s}_0 \pmod{q}$ with $\boldsymbol{t}_0 \in \mathcal{R}(q/2)$
6:     $\boldsymbol{a} \longleftarrow \boldsymbol{g}^{-1} * (\boldsymbol{t}_p - \boldsymbol{t}_0) \pmod{p}$ with $\boldsymbol{a} \in \mathcal{R}(p/2)$
7:     $(\boldsymbol{s}, \boldsymbol{t}) \longleftarrow (\boldsymbol{s}_0, \boldsymbol{t}_0) + (\boldsymbol{a} * \boldsymbol{f}, \boldsymbol{a} * \boldsymbol{g})$
8: until $\|\boldsymbol{a} * \boldsymbol{f}\| \leq B_s$ and $\|\boldsymbol{a} * \boldsymbol{g}\| \leq B_t$ and $\|\boldsymbol{s}\| \leq \frac{q}{2} - B_s$ and $\|\boldsymbol{t}\| \leq \frac{q}{2} - B_t$
Output: $(\boldsymbol{s}, \boldsymbol{t}, \mu)$
```

Remark 1. Notice the rejection criterion in Step 8 of the signing algorithm. We compute a potential signature $(\boldsymbol{s}, \boldsymbol{t})$, but then we reject it if it, or the correction $(\boldsymbol{a} * \boldsymbol{f}, \boldsymbol{a} * \boldsymbol{g})$, is too big; specifically, we reject $(\boldsymbol{s}, \boldsymbol{t})$ if it falls outside of $L_{\boldsymbol{h}}\left(\frac{q}{2} - B_s, \frac{q}{2} - B_t\right)$, or if $(\boldsymbol{a} * \boldsymbol{f}, \boldsymbol{a} * \boldsymbol{g})$ falls outside $L_{\boldsymbol{h}}(B_s, B_t)$.

Remark 2. Since $\boldsymbol{t} \equiv \boldsymbol{h} * \boldsymbol{s} \pmod{q}$ it does not need to be published explicitly. Furthermore since $\boldsymbol{s} \equiv \boldsymbol{s}_p \pmod{p}$ and $\boldsymbol{s}_p$ can be obtained by hashing $\boldsymbol{h}$ with the message, the signer can simply publish $(\boldsymbol{s} - \boldsymbol{s}_p)/p$ as the signature. The resulting signature is of length $N\lceil \log_2 q/p \rceil$ bits.

Algorithm 2 NTRUMLS Verification Algorithm

```
Input: $(\boldsymbol{s}, \boldsymbol{t}, \mu, \boldsymbol{h})$
$valid \longleftarrow$ yes
$(\boldsymbol{s}_p, \boldsymbol{t}_p) \longleftarrow \mathsf{Hash}(\boldsymbol{h}, \mu)$
if $\boldsymbol{t} \not\equiv \boldsymbol{h} * \boldsymbol{s} \pmod{q}$ then
    $valid \longleftarrow$ no
end if
if $\|\boldsymbol{s}\| > \frac{q}{2} - B_s$ or $\|\boldsymbol{t}\| > \frac{q}{2} - B_t$ then
    $valid \longleftarrow$ no
end if
if $(\boldsymbol{s}, \boldsymbol{t}) \not\equiv (\boldsymbol{s}_p, \boldsymbol{t}_p) \pmod{p}$ then
    $valid \longleftarrow$ no
end if
Output: $valid$
```

Proposition 1. *The Signing Algorithm produces signatures that are verified as valid by the Verification Algorithm.*

Proof. This is an easy exercise.

3 Transcript Security

In this section we prove that, under a reasonable assumption, a transcript of signatures created using the signing algorithm contains no information that is not already available to someone who knows the public verification key $\boldsymbol{h}$. We do this by showing that an honest signer produces signatures that are uniformly distributed on $L_{\boldsymbol{h}}\left(\frac{q}{2} - B_s, \frac{q}{2} - B_t\right)$. We are able to show that for any document hash, $(\boldsymbol{s}_p, \boldsymbol{t}_p)$, the signer's distribution is precisely the uniform distribution on the subset of signature points in $(\boldsymbol{s}_p, \boldsymbol{t}_p) + p\mathbb{Z}^{2N}$ (proposition 2). For uniformity on the entire signature region we must assume that each coset of $p\mathbb{Z}^{2N}$ contains roughly the same number of signature points (assumption 1).

We further show that a party who knows $\boldsymbol{h}$ alone can produce a transcript of pairs

$$(\text{Valid Signature}_i, \text{Document Hash}_i)_{i=1,2,3,\ldots}$$

that is statistically indistinguishable from an analogous transcript produced using the signing algorithm and the private key $(\boldsymbol{f}, \boldsymbol{g})$. Specifically, the signature points produced by such a party are uniform on $L_{\boldsymbol{h}}\left(\frac{q}{2} - B_s, \frac{q}{2} - B_t\right)$, and the document hashes (obtained by reducing the signature coefficients modulo p), are uniform on $\mathcal{R}(p/2)$.

We start by analyzing the transcript created using the signing algorithm and $(\boldsymbol{f}, \boldsymbol{g})$. We note that the rejection sampling condition is what allows us to prove that the resulting signatures are uniformly distributed in a certain space of allowable signatures.

We assume that our hash function outputs document hashes

$$(\boldsymbol{s}_p, \boldsymbol{t}_p) \in \mathcal{R}(p/2)^2$$

that are uniformly distributed on $\mathcal{R}(p/2)^2$. We use Steps 3 through 7 of the Signing Algorithm to define a signing function

$$(\boldsymbol{s}, \boldsymbol{t}) = \sigma'(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{s}_p, \boldsymbol{t}_p, \boldsymbol{r}).$$

Thus σ' is a map

$$\begin{aligned}\sigma' : \overbrace{p\mathcal{R}\left(\frac{3}{2}\right) \times \mathcal{R}\left(\frac{p}{2}\right)}^{\text{private key } (\boldsymbol{f},\boldsymbol{g})} \times \overbrace{\mathcal{R}\left(\frac{p}{2}\right) \times \mathcal{R}\left(\frac{p}{2}\right)}^{\text{document hash } (\boldsymbol{s}_p,\boldsymbol{t}_p)} \times \overbrace{\mathcal{R}\left(\left\lfloor \frac{q}{2p} - \frac{1}{2} \right\rfloor\right)}^{\text{random element } \boldsymbol{r}} \\ \longrightarrow \underbrace{L_{\boldsymbol{h}}\left(\frac{q}{2} + B_s, \frac{q}{2} + B_t\right)}_{\text{potential signature } (\boldsymbol{s},\boldsymbol{t})}\end{aligned}$$

given explicitly by

$$\sigma'(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{s}_p, \boldsymbol{t}_p, \boldsymbol{r}) = (\boldsymbol{s}_0 + \boldsymbol{a} * \boldsymbol{f}, \boldsymbol{t}_0 + \boldsymbol{a} * \boldsymbol{g}), \tag{1}$$

where

$$\boldsymbol{s}_0 = \boldsymbol{s}_p + p\boldsymbol{r}, \tag{2}$$

$$\boldsymbol{t}_0 \equiv \boldsymbol{h} * \boldsymbol{s}_0 \pmod{q} \quad \text{with } \boldsymbol{t}_0 \in \mathcal{R}(q/2), \tag{3}$$

$$\boldsymbol{a} \equiv \boldsymbol{g}^{-1} * (\boldsymbol{t}_p - \boldsymbol{t}_0) \pmod{p} \quad \text{with } \boldsymbol{a} \in \mathcal{R}(p/2). \tag{4}$$

We will write

$$\Omega' = p\mathcal{R}\left(\frac{3}{2}\right) \times \mathcal{R}\left(\frac{p}{2}\right) \times \mathcal{R}\left(\frac{p}{2}\right) \times \mathcal{R}\left(\frac{p}{2}\right) \times \mathcal{R}\left(\left\lfloor \frac{q}{2p} - \frac{1}{2} \right\rfloor\right)$$

for the domain of σ'.

We now introduce rejection sampling by defining

$$\Omega_{B_s,B_t} = \left\{ (\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{s}_p, \boldsymbol{t}_p, \boldsymbol{r}) \in \Omega' : \begin{array}{r} (\boldsymbol{s}, \boldsymbol{t}) := \sigma'(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{s}_p, \boldsymbol{t}_p, \boldsymbol{r}) \\ = (\boldsymbol{s}_0 + \boldsymbol{a} * \boldsymbol{f}, \boldsymbol{t}_0 + \boldsymbol{a} * \boldsymbol{g}), \\ \|\boldsymbol{s}\| \le \frac{q}{2} - B_s, \|\boldsymbol{t}\| \le \frac{q}{2} - B_t, \\ \|\boldsymbol{a} * \boldsymbol{f}\| \le B_s, \|\boldsymbol{a} * \boldsymbol{g}\| \le B_t \end{array} \right\}.$$

The restriction of σ' to Ω_{B_s,B_t}, which we denote by σ, is then a map

$$\sigma : \Omega_{B_s,B_t} \longrightarrow L_{\boldsymbol{h}}\left(\frac{q}{2} - B_s, \frac{q}{2} - B_t\right).$$

To ease notation, we let

$$A = \left\lfloor \frac{q}{2p} + \frac{1}{2} \right\rfloor,$$

so by Step 3 of the Signing Algorithm, the random element $\boldsymbol{r}$ used to generate a signature is chosen uniformly from the set $\mathcal{R}(A)$. The following proposition says that every signature that is valid for the document hash $(\boldsymbol{s}_p, \boldsymbol{t}_p)$ has the same number of preimages in $\mathcal{R}(A)$.

Proposition 2. *The signature function σ has the following property: For a given*

$$\textit{private key}\quad (\boldsymbol{f}, \boldsymbol{g}) \in p\mathcal{R} \times \mathcal{R},$$
$$\textit{document hash}\quad (\boldsymbol{s}_p, \boldsymbol{t}_p) \in \mathcal{R}\left(\frac{p}{2}\right) \times \mathcal{R}\left(\frac{p}{2}\right),$$

the output of σ, when queried on uniformly random $\boldsymbol{r} \in \mathcal{R}(A)$, is uniformly distributed over the set

$$\left\{(\boldsymbol{s}, \boldsymbol{t}) \in L_h\left(\frac{q}{2} - B_s, \frac{q}{2} - B_t\right) : (\boldsymbol{s}, \boldsymbol{t}) \equiv (\boldsymbol{s}_p, \boldsymbol{t}_p) \pmod{p}\right\}.$$

of valid signatures for $(\boldsymbol{s}_p, \boldsymbol{t}_p)$. Equivalently, the size of the set

$$\{\boldsymbol{r} \in \mathcal{R}(A) : \sigma(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{s}_p, \boldsymbol{t}_p, \boldsymbol{r}) = (\boldsymbol{s}, \boldsymbol{t})\}$$

is the same for all

$$(\boldsymbol{s}, \boldsymbol{t}) \in L_{\boldsymbol{h}}\left(\frac{q}{2} - B_s, \frac{q}{2} - B_t\right) \quad \textit{satisfying} \quad (\boldsymbol{s}, \boldsymbol{t}) \equiv (\boldsymbol{s}_p, \boldsymbol{t}_p) \pmod{p}.$$

Proof. Since we know from Proposition 1 that $\sigma(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{s}_p, \boldsymbol{t}_p, \boldsymbol{r})$ is congruent to $(\boldsymbol{s}_p, \boldsymbol{t}_p)$ modulo p, it is clear that there is zero probability of generating the signature $(\boldsymbol{s}, \boldsymbol{t})$ if $(\boldsymbol{s}, \boldsymbol{t}) \not\equiv (\boldsymbol{s}_p, \boldsymbol{t}_p) \pmod{p}$. So we assume henceforth that

$$(\boldsymbol{s}, \boldsymbol{t}) \equiv (\boldsymbol{s}_p, \boldsymbol{t}_p) \pmod{p}. \tag{5}$$

The random element $\boldsymbol{r}$ used to generate a signature is chosen uniformly from the set $\mathcal{R}(A)$, so there are $(2A+1)^N$ possible choices for $\boldsymbol{r}$. Hence the probability of obtaining $(\boldsymbol{s}, \boldsymbol{t})$ as a signature on $(\boldsymbol{s}_p, \boldsymbol{t}_p)$ is equal to $(2A+1)^{-N}$ times the number of elements in the set

$$\Sigma(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{t}) = \big\{\boldsymbol{r} \in \mathcal{R}(A) : \sigma(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{s}_p, \boldsymbol{t}_p, \boldsymbol{r}) = (\boldsymbol{s}, \boldsymbol{t})\big\}. \tag{6}$$

The key to counting the size of the set $\Sigma(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{t})$ is the bijection described in the following lemma.

Lemma 1. *Let*

$$\mathcal{C} = \left\{\boldsymbol{b} \in \mathcal{R}\left(\frac{p}{2}\right) : \big\|\boldsymbol{b} * \boldsymbol{f}\big\| \le B_s \textit{ and } \big\|\boldsymbol{b} * \boldsymbol{g}\big\| \le B_t\right\},$$

and let

$$(\boldsymbol{s}, \boldsymbol{t}) \in L_{\boldsymbol{h}}\left(\frac{q}{2} - B_s, \frac{q}{2} - B_t\right) \quad \textit{satisfy} \quad (\boldsymbol{s}, \boldsymbol{t}) \equiv (\boldsymbol{s}_p, \boldsymbol{t}_p) \pmod{p}.$$

Then there is a well-defined bijection of sets

$$\begin{aligned} \phi : \mathcal{C} &\longrightarrow \Sigma(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{t}), \\ \boldsymbol{b} &\longmapsto \frac{\boldsymbol{s} - \boldsymbol{s}_p}{p} - \boldsymbol{b} * \frac{\boldsymbol{f}}{p}. \end{aligned} \tag{7}$$

Proof. First, since the coefficients of $\boldsymbol{s}-\boldsymbol{s}_p$ are multiples of p, and similarly $\boldsymbol{f}\in p\mathcal{R}(3/2)$ has coefficients divisible by p, we see that the polynomial on the right-hand side of (7) has coefficients in $\mathbb{Z}$.

We next need to show that $\phi(\boldsymbol{b})\in\Sigma(\boldsymbol{f},\boldsymbol{g},\boldsymbol{s},\boldsymbol{t})$, which by the definition of $\Sigma(\boldsymbol{f},\boldsymbol{g},\boldsymbol{s},\boldsymbol{t})$ means showing that $\phi(\boldsymbol{b})\in\mathcal{R}(A)$ and

$$\sigma\bigl(\boldsymbol{f},\boldsymbol{g},\boldsymbol{s}_p,\boldsymbol{t}_p,\phi(\boldsymbol{b})\bigr)=(\boldsymbol{s},\boldsymbol{t}).$$

First note that because $\boldsymbol{s}\in\mathcal{R}\left(\frac{q}{2}-B_s\right)$, $\boldsymbol{s}_p\in\mathcal{R}\left(\frac{p}{2}\right)$, and $\boldsymbol{b}\in\mathcal{C}$, the triangle inequality gives

$$\bigl\|\phi(\boldsymbol{b})\bigr\|=\left\|\frac{1}{p}\left(\boldsymbol{s}-\boldsymbol{s}_p-\boldsymbol{b}*\boldsymbol{f}\right)\right\|\le\left\lfloor\frac{\frac{q}{2}-B_s+\frac{p}{2}+B_s}{p}\right\rfloor=A.$$

The use of the floor function is justified by noting that $\phi(\boldsymbol{b})$ has integer coefficients. This establishes that $\phi(\boldsymbol{b})\in\mathcal{R}\,(A)$.

Next we use the four formulas (1)–(4) to compute the signature $\sigma\bigl(\boldsymbol{f},\boldsymbol{g},\boldsymbol{s}_p,\boldsymbol{t}_p,\phi(\boldsymbol{b})\bigr)$:

$$\begin{aligned}
\boldsymbol{s}_0&=\boldsymbol{s}_p+p\phi(\boldsymbol{b})\\
&=\boldsymbol{s}_p+p\left(\frac{\boldsymbol{s}-\boldsymbol{s}_p}{p}-\boldsymbol{b}*\frac{\boldsymbol{f}}{p}\right)\\
&=\boldsymbol{s}-\boldsymbol{b}*\boldsymbol{f}, &&(8)\\
\boldsymbol{t}_0&\equiv\boldsymbol{h}*\boldsymbol{s}_0 \pmod{q}\\
&\equiv\boldsymbol{h}*(\boldsymbol{s}-\boldsymbol{b}*\boldsymbol{f}) \pmod{q}\\
&\equiv\boldsymbol{h}*\boldsymbol{s}-\boldsymbol{b}*\boldsymbol{g} \pmod{q}\quad\text{since } \boldsymbol{h}\equiv\boldsymbol{f}^{-1}*\boldsymbol{g} \pmod{q},\\
&\equiv\boldsymbol{t}-\boldsymbol{b}*\boldsymbol{g} \pmod{q}\quad\text{since } (\boldsymbol{s},\boldsymbol{t})\in L_{\boldsymbol{h}}. &&(9)
\end{aligned}$$

Since $(\boldsymbol{s},\boldsymbol{t})\in L_{\boldsymbol{h}}\left(\frac{q}{2}-B_s,\frac{q}{2}-B_t\right)$ and $\boldsymbol{b}\in\mathcal{C}$, we have

$$\begin{aligned}
\bigl\|\boldsymbol{s}_0\bigr\|&\le\bigl\|\boldsymbol{s}\bigr\|+\bigl\|\boldsymbol{b}*\boldsymbol{f}\bigr\|=\frac{q}{2}-B_s+B_s=\frac{q}{2},\\
\bigl\|\boldsymbol{t}_0\bigr\|&\le\bigl\|\boldsymbol{t}\bigr\|+\bigl\|\boldsymbol{b}*\boldsymbol{g}\bigr\|=\frac{q}{2}-B_t+B_t=\frac{q}{2},
\end{aligned}$$

i.e. (9), similar to (8), is an equality, not just a congruence. Continuing with the computation of $\sigma\bigl(\boldsymbol{f},\boldsymbol{g},\boldsymbol{s}_p,\boldsymbol{t}_p,\phi(\boldsymbol{b})\bigr)$, we use (5) to compute

$$\boldsymbol{a}\equiv\boldsymbol{g}^{-1}*(\boldsymbol{t}_p-\boldsymbol{t}_0)\equiv\boldsymbol{b}\pmod{p}.$$

(Note that $\boldsymbol{t}\equiv\boldsymbol{t}_p\pmod{p}$ from (4).) Since both $\boldsymbol{a}$ and $\boldsymbol{b}$ are in $\mathcal{R}(p/2)$, this tells us that $\boldsymbol{a}=\boldsymbol{b}$.

We now use (1) to compute the signature

$$\begin{aligned}
\sigma\bigl(\boldsymbol{f},\boldsymbol{g},\boldsymbol{s}_p,\boldsymbol{t}_p,\phi(\boldsymbol{b})\bigr)&=(\boldsymbol{s}_0+\boldsymbol{a}*\boldsymbol{f},\boldsymbol{t}_0+\boldsymbol{a}*\boldsymbol{g})\quad\text{definition of }\sigma,\\
&=(\boldsymbol{s}-\boldsymbol{b}*\boldsymbol{f}+\boldsymbol{a}*\boldsymbol{f},\boldsymbol{t}-\boldsymbol{b}*\boldsymbol{g}+\boldsymbol{a}*\boldsymbol{g})\\
&\qquad\qquad\qquad\text{from (8) and (9),}\\
&=(\boldsymbol{s},\boldsymbol{t})\quad\text{since }\boldsymbol{a}=\boldsymbol{b}.
\end{aligned}$$

Hence directly from the definition (6) of the set $\Sigma(\boldsymbol{f},\boldsymbol{g},\boldsymbol{s},\boldsymbol{t})$, we see that

$$\phi(\boldsymbol{b}) \in \Sigma(\boldsymbol{f},\boldsymbol{g},\boldsymbol{s},\boldsymbol{t}).$$

We next fix an $\boldsymbol{r} \in \Sigma(\boldsymbol{f},\boldsymbol{g},\boldsymbol{s},\boldsymbol{t})$ and compute how many $\boldsymbol{b} \in \mathcal{C}$ satisfy $\phi(\boldsymbol{b}) = \boldsymbol{r}$. Since all coefficients of the polyomials $\boldsymbol{s} - \boldsymbol{s}_p$ and $\boldsymbol{f}$ are divisible by p, to ease notation we write

$$\boldsymbol{s} - \boldsymbol{s}_p = p\boldsymbol{S} \quad \text{and} \quad \boldsymbol{f} = p\boldsymbol{F}.$$

We recall that by assumption, the polynomial $\boldsymbol{F}$ is invertible modulo p. We have

$$\begin{aligned} \phi(\boldsymbol{b}) = \boldsymbol{r} &\Longleftrightarrow \boldsymbol{S} - \boldsymbol{b} * \boldsymbol{F} = \boldsymbol{r} \\ &\Longleftrightarrow \boldsymbol{b} \equiv \boldsymbol{F}^{-1} * (\boldsymbol{S} - \boldsymbol{r}) \pmod{p} \quad \text{and} \quad \|\boldsymbol{b}\| \le \frac{p}{2}. \end{aligned}$$

There is thus exactly one value of $\boldsymbol{b}$ in $\mathcal{C}$ satisfying $\phi(\boldsymbol{b}) = \boldsymbol{r}$, namely the unique element of $\mathcal{C}$ that is congruent modulo p to $\boldsymbol{F}^{-1} * (\boldsymbol{S} - \boldsymbol{r})$. This shows that ϕ is bijective, which concludes the proof of Lemma 1.

Resuming the proof of Proposition 2, we have, for all $(\boldsymbol{s},\boldsymbol{t}) \equiv (\boldsymbol{s}_p,\boldsymbol{t}_p) \pmod{p}$,

$$\begin{aligned} &\mathrm{Prob}_{\boldsymbol{r}\leftarrow\mathcal{R}(A)}\left(\begin{matrix}\text{signature}\\ \text{is } (\boldsymbol{s},\boldsymbol{t})\end{matrix} \middle| \begin{matrix}\text{private key is } (\boldsymbol{f},\boldsymbol{g}) \text{ and}\\ \text{document hash is } (\boldsymbol{s}_p,\boldsymbol{t}_p)\end{matrix}\right) \\ &\qquad\qquad\qquad = \frac{\#\Sigma(\boldsymbol{f},\boldsymbol{g},\boldsymbol{s},\boldsymbol{t})}{\#\mathcal{R}(A)} = \frac{\#\mathcal{C}}{\#\mathcal{R}(A)}, \end{aligned}$$

where the penultimate equality follows from Lemma 1. This completes the proof of Proposition 2.

To give a complete proof of transcript security we need a slightly stronger version of Proposition 2 to be true:

Proposition 3. *The distribution of signatures produced by querying σ on uniformly random $(\boldsymbol{s}_p,\boldsymbol{t}_p) \in R(p/2)^2$ and uniformly random $\boldsymbol{r} \in \mathcal{R}(A)$ is indistinguishable from the uniform distribution on $L_{\boldsymbol{h}}\left(\frac{q}{2} - B_s, \frac{q}{2} - B_t\right)$.*

Proposition 3 is an immediate consequence of proposition 2 under the assumption that, for any given $\boldsymbol{h}$, the number of lattice vectors of bounded norm in each coset of $p\mathbb{Z}^{2N}$ is essentially constant. This certainly fails to be the case for some lattices, for instance $\boldsymbol{h} = 1$ has vectors in only p^N distinct cosets. However, it seems likely that assumption 1 holds for the lattices used in NTRUMLS.

Assumption 1 *There are constants C, ϵ such that $\epsilon = 1/\mathrm{poly}(N)$ and for all $(\boldsymbol{s}_p,\boldsymbol{t}_p) \in \mathcal{R}(p/2)$*

$$(1-\epsilon)C \le \left| L_{\boldsymbol{h}}\left(\frac{q}{2} - B_s, \frac{q}{2} - B_t\right) \cap \left((\boldsymbol{s}_p,\boldsymbol{t}_p) + p\mathbb{Z}^{2N}\right)\right| \le (1+\epsilon)C.$$

We conclude this section by noting that any party with access to $\boldsymbol{h}$ can sample the uniform distribution on $L_{\boldsymbol{h}}\left(\frac{q}{2}-B_s, \frac{q}{2}-B_t\right)$. One simply generates random $\boldsymbol{s} \in \mathcal{R}(\frac{q}{2}-B_s)$ until $\boldsymbol{h} * \boldsymbol{s} \in \mathcal{R}(\frac{q}{2}-B_t)$. Since the signing region contains a large fraction of $L_{\boldsymbol{h}}\left(\frac{q}{2}, \frac{q}{2}\right)$ (at least 30% for the parameter sets we consider), this suceeds after a small number of iterations. A transcript of

$$((\boldsymbol{s}, \boldsymbol{t})_i, (\boldsymbol{s}_p, \boldsymbol{t}_p)_i)_{i=1,2,3,\ldots}$$

where $(\boldsymbol{s}, \boldsymbol{t})_i$ is produced in this manner and $(\boldsymbol{s}_p, \boldsymbol{t}_p)_i = (\boldsymbol{s}, \boldsymbol{t})_i \pmod p$ is uniformly distributed on $L_{\boldsymbol{h}}\left(\frac{q}{2}-B_s, \frac{q}{2}-B_t\right) \times \mathcal{R}(p/2)$ by assumption 1. By proposition 3, and the assumption that the output of Hash is uniform on $\mathcal{R}(p/2)^2$, this transcript is indistinguishable from one produced by an honest signer. The only difference between the two transcripts is that the party who used $\boldsymbol{h}$ alone does not know messages, μ_i, such that $\mathsf{Hash}(\boldsymbol{h}, \mu_i) = (\boldsymbol{s}_p, \boldsymbol{t}_p)_i$.

4 Probability of Generating a Valid Signature

To simplify our analysis we let $B = \lceil p^2 N/4 \rceil$ and take

$$B_s = B_t = B.$$

With this assumption there is zero probability of rejecting a candidate signature due to $\left\|\boldsymbol{a} * \boldsymbol{s}\right\| > B_s$ or $\left\|\boldsymbol{a} * \boldsymbol{t}\right\| > B_t$, but the probability of rejection due to non-inclusion in $\mathcal{R}(q/2-B) \times \mathcal{R}(q/2-B)$ is significant. Regardless, we can show that the probability of generating a valid signature is approximately $e^{-8/k}$, which is still practical. Further, the probability of rejection can be made significantly lower by fine-tuning B_s and B_t; our proposed parameters in section 6 reflect this optimization.

For this section we assume that the various parameters satisfy the conditions given in Table 1.

N	a moderate sized prime, say $200 < N < 5000$
p	a small prime chosen so that $N \log_2(p)$ is greater than the desired bit security
B	$\leq \lceil p^2 N/4 \rceil$
k	a small constant, say $2 \leq k \leq 50$
q	an integer coprime with p and satisfying $q \approx kNB \approx kp^2N^2/4$

Table 1. Parameter guidelines

The rejection criterion says that we only accept signatures whose norm is smaller than $q/2 - B$, so we want q to be a lot larger than B, or it will be too hard to find an acceptable signature. We consider the sup norm of a potential signature

$$(\boldsymbol{s}, \boldsymbol{t}) = (\boldsymbol{s}_0, \boldsymbol{t}_0) + (\boldsymbol{a} * \boldsymbol{f}, \boldsymbol{a} * \boldsymbol{g})$$

produced in Step 7 of the signing algorithm. The coefficients of $\boldsymbol{s}_0$ and $\boldsymbol{t}_0$ are in $\mathcal{R}(q/2)$, the coefficents of $\boldsymbol{a} * \boldsymbol{f}$ are in $\mathcal{R}(p^2N/4)$, and the coefficients of $\boldsymbol{a} * \boldsymbol{g}$ are in $\mathcal{R}(pN/2)$. Hence the coefficients of an $(\boldsymbol{s}, \boldsymbol{t})$ pair produced by Step 7 satisfy

$$\big\|(\boldsymbol{s}, \boldsymbol{t})\big\| \le \frac{q}{2} + B. \tag{10}$$

We will make the simplifying assumption[4] that the coefficients of $\boldsymbol{s}$ and $\boldsymbol{t}$ are equally likely to take on each of the values in the interval (10). The rejection criterion says that we only accept signatures whose coefficents are at most $q/2 - B$. Since we need all $2N$ of the coefficients of $(\boldsymbol{s}, \boldsymbol{t})$ to satisfy this condition, we find that

$$\operatorname{Prob}\big((\boldsymbol{s}, \boldsymbol{t}) \text{ is accepted}\big) \approx \left(\frac{q/2 - B}{q/2 + B}\right)^{2N}.$$

Using the chosen value

$$q \approx \frac{kp^2N^2}{4} \approx kNB$$

from Table 1, we find that

$$\begin{aligned}\operatorname{Prob}\big((\boldsymbol{s}, \boldsymbol{t}) \text{ is accepted}\big) &\approx \left(\frac{1 - 2B/q}{1 + 2B/q}\right)^{2N} \\ &\approx \left(\frac{1 - 2/kN}{1 + 2/kN}\right)^{2N} \\ &\approx e^{-8/k},\end{aligned}$$

where for the last equality we use the estimate $(1 + t/n)^n \approx e^t$, valid when t is small and n is large.

5 Lattice Problems Associated to NTRUMLS

We consider both lattice based forgery attacks and key recovery attacks. For all of the proposed parameter sets in Section 6, the parameters have been chosen so that the difficulty of the private key lattice problem is roughly equal to that of the lattice forgery problem, taking into account the heuristic fact that solving unique-SVP tends to be a bit easier in practice than it is in theory.

[4] In actuality, the coefficients of the products $\boldsymbol{a} * \boldsymbol{f}$ and $\boldsymbol{a} * \boldsymbol{g}$ tend to cluster more towards 0, since they are more-or-less hypergeometrically distributed.

5.1 Forgery attack

In this section we consider the lattice problems underlying signature keys and signature forgery. We note that shortest and closest vector problems (SVP and CVP) are analysed using the L^2-norm, not the L^∞-norm. We write

$$\|\boldsymbol{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots}$$

for the L^2-norm of the vector $\boldsymbol{v} = (v_1, v_2, \ldots)$.

We will use the following elementary lattice result, whose proof we defer to Section A of the appendix.

Proposition 4. *Let $L_1 \subset \mathbb{Z}^r$ and $L_2 \subset \mathbb{Z}^r$ be lattices of rank r, let $\boldsymbol{t}_1, \boldsymbol{t}_2 \in \mathbb{Z}^r$ be arbitrary vectors, and let*

$$M = (L_1 + \boldsymbol{t}_1) \cap (L_2 + \boldsymbol{t}_2)$$

be the intersection of the indicated translations of L_1 and L_2. We make the following assumptions:

(i) $\gcd\bigl(\det(L_1), \det(L_2)\bigr) = 1$.
(ii) *Either $\boldsymbol{t}_1 \notin L_1$ or $\boldsymbol{t}_2 \notin L_2$ (or both), so in particular $M \neq L_1 \cap L_2$.*

Then the following are true:

(a) $\det(L_1 \cap L_2) = \det(L_1) \cdot \det(L_2)$.
(b) *$M \neq \emptyset$.*
(c) *For every $\boldsymbol{w}_0 \in M$, the map*

$$L_1 \cap L_2 \longrightarrow M, \qquad \boldsymbol{v} \longmapsto \boldsymbol{v} + \boldsymbol{w}_0 \tag{11}$$

is a bijection.
(d) *Let $\boldsymbol{w}_0 \in M$, and let $\boldsymbol{w}' \in M$ be a shortest non-zero vector in M. Then $\boldsymbol{w}_0 - \boldsymbol{w}'$ solves the the closest vector problem in $L_1 \cap L_2$ for the vector $\boldsymbol{w}_0$.* (*This is true for any norm on $\mathbb{Z}^r$, so in particular it is true for both the L^∞ norm and the L^2 norm.*)

We recall two key quantities associated to lattice problems.

Heuristic 1 *The* Gaussian heuristic *says that the likely L^2-size of a solution to SVP or CVP in a "random" lattice L of reasonably large dimension is approximately*

$$\gamma(L) = \sqrt{\frac{\dim L}{2\pi e}} \cdot \det(L)^{1/\dim(L)}.$$

In other words, for "most" lattices L and "most" target vectors $\boldsymbol{v}_0$,

$$\min_{\boldsymbol{v} \in L \smallsetminus \boldsymbol{0}} \|\boldsymbol{v}\|_2 \approx \gamma(L) \quad \textit{and} \quad \min_{\boldsymbol{v} \in L} \|\boldsymbol{v} - \boldsymbol{v}_0\|_2 \approx \gamma(L).$$

Heuristic 2 *Let $L \subset \mathbb{Z}^n$ be a lattice for which we want to solve either τ-appr-SVP or τ-appr-CVP. In other words, let $\boldsymbol{v}_0 \in \mathbb{Z}^n$, and suppose that we want to find a vector $\boldsymbol{v} \in L$ satisfying either*

$$0 < \|\boldsymbol{v}\|_2 \le \tau \quad \textit{or} \quad \|\boldsymbol{v} - \boldsymbol{v}_0\|_2 \le \tau.$$

We call τ the target length *of the problem. The* Gaussian defect *of the problem is the ratio*

$$\rho(L, \tau) = \frac{\tau}{\gamma(L)}.$$

*Let $0 < \delta < 2$. The δ-*LLL heuristic*, which has been confirmed in numerous experiments, says that solving the τ-appr-SVP or τ-appr-CVP problem is (exponentially) hard as a function of* $\dim(L)$, *provided that the Gaussian defect $\rho(L, \tau)$ is no more than a small multiple of* $\dim(L)^\delta$.

We consider the problem of forging a signature. The forger needs to find a vector $(\boldsymbol{s}, \boldsymbol{t}) \in L_{\boldsymbol{h}}$ satisfying:

$$\text{Congruence Condition :} \qquad (\boldsymbol{s}, \boldsymbol{t}) = (\boldsymbol{s}_p, \boldsymbol{t}_p) \pmod{p}. \tag{12}$$

$$\text{Norm Condition :} \qquad \|\boldsymbol{s}\| \le \frac{q}{2} - B_s \tag{13}$$

$$\|\boldsymbol{t}\| \le \frac{q}{2} - B_t. \tag{14}$$

N.B. The norm condition (13) is an L^∞-norm condition.

The vectors $\boldsymbol{s}_p, \boldsymbol{t}_p \in R(p/2)$ are given, so the congruence condition (12) may be rephrased as saying that the target vector $(\boldsymbol{s}, \boldsymbol{t})$ is in the translation of the lattice $p\mathbb{Z}^{2N}$ by the vector $(\boldsymbol{s}_p, \boldsymbol{t}_p)$. Thus the forger is looking for an L^∞-short vector in the intersection

$$(\boldsymbol{s}, \boldsymbol{t}) \in L_{\boldsymbol{h}} \cap \bigl(p\mathbb{Z}^{2N} + (\boldsymbol{s}_p, \boldsymbol{t}_p)\bigr).$$

The determinants

$$\det(L_{\boldsymbol{h}}) = q^N \quad \text{and} \quad \det(p\mathbb{Z}^{2N}) = p^{2N}$$

are relatively prime, so we can use Proposition 4(a) to conclude that

$$\det(L_{\boldsymbol{h}} \cap p\mathbb{Z}^{2N}) = p^{2N} q^N.$$

Then Proposition 4(d) tells us that finding a short vector in the intersection $L_{\boldsymbol{h}} \cap \bigl(p\mathbb{Z}^{2N} + (\boldsymbol{s}_p, \boldsymbol{t}_p)\bigr)$ is equivalent to solving an appr-CVP problem in the lattice $L_{\boldsymbol{h}} \cap p\mathbb{Z}^{2N}$. Since the Gaussian heuristic of $L_{\boldsymbol{h}} \cap p\mathbb{Z}^{2N}$ is

$$\gamma(L_{\boldsymbol{h}} \cap p\mathbb{Z}^{2N}) = \sqrt{\frac{N}{\pi e}}(p^{2N} q^N)^{1/2N} = \sqrt{\frac{p^2 q N}{\pi e}},$$

it only remains to estimate the target length.

The rejection criterion in the signature algorithm says that a valid signature $(\boldsymbol{s}, \boldsymbol{t})$ has sup norm at most $q/2 - \min(B_s, B_t)$. Hence in particular a valid signature satisfies the L^2-norm bound

$$\left\|(\boldsymbol{s}, \boldsymbol{t})\right\|_2 \leq \left(\frac{q}{2} - \min(B_s, B_t)\right) \sqrt{2N}, \tag{15}$$

but not every vector in $L_{\boldsymbol{h}}$ satisfying the L^2-norm condition (15) and the congruence condition (12) will be a valid signature. We are going to simplify the life of a potential forger and assume that she only needs to satisfy the L^2-norm condition (15), rather than the more stringent L^∞-norm condition (13). Furthermore we will assume, again in the forger's favor, that $B_s = B_t = 0$, so that the she need only find a vector in $\mathcal{R}(\frac{q}{2}) \times \mathcal{R}(\frac{q}{2})$. This gives a target length

$$\tau = q\sqrt{N/2}.$$

Hence the Gaussian defect for our appr-CVP problem is

$$\rho = \frac{q\sqrt{N/2}}{\sqrt{p^2 qN/2\pi e}},$$

and using the relations in Table 1 between the various parameters, a little bit of algebra yields

$$\rho = N\sqrt{\frac{k\pi e}{8}}.$$

Thus ρ is a small multiple of $\dim(L_{\boldsymbol{h}} \cap p\mathbb{Z}^{2N})$, so the LLL-heuristic says that solving the associated appr-CVP is a hard problem provided that the dimension is chosen appropriately. Of course, in practice one needs to do experiments with current LLL technology to obtain extrapolated estimates for the actual running time when N is moderately large, say in the range from 500 to 5000.

5.2 Key recovery attack

To recover the secret key $(\boldsymbol{f}, \boldsymbol{g})$ from the public key $\boldsymbol{h}$ is a well known problem associated with NTRU lattices. It has previously been studied intensively, such as in [7, 5, 6]. In this subsection, we analyse the key strength against best known attack, namely, the hybrid approach [5] of a lattice reduction attack and a meet-in-the-middle search. The results presented in this subsection follows a recent analysis of NTRU lattice in [6].

Recall the problem of finding the private key $(\boldsymbol{f}, \boldsymbol{g})$ in an isomorphic module from the public key $\boldsymbol{h}$ can be seen as finding a short vector with in the lattice

$$\left(\begin{array}{c|c} qI_N & 0 \\ \hline H & I_N \end{array}\right).$$

Indeed, there exists N unique shortest vectors with in the lattice, namely $(\boldsymbol{f}, \boldsymbol{g})$, and its cyclic rotations. Recovering any of them, or a small linear combination of them, allows the attacker to break the system. For simplicity, we assume the task for the attacker is to find $(\boldsymbol{f}, \boldsymbol{g})$. Also note the attacker knows that $\boldsymbol{f} = p\boldsymbol{F}$, and standard methods allow him to reduce to the problem of finding the shorter vector $(\boldsymbol{F}, \mathbf{g})$.

The hybrid attack works as follows: one first chooses r_1, r_2 such that $1 \leq r_1 < r_2 < N$ and defines $m = r_2 - r_1$ and $N_1 = N - r_1$. One then extracts an m by m block of coefficients from the center of L and calls this L_2. Equivalently, one partitions the original matrix as:

$$\left(\begin{array}{c|c} qI_N & 0 \\ \hline H & I_N \end{array}\right) = \left(\begin{array}{c|c|c} qI_{r_1} & 0 & 0 \\ \hline * & L_2 & 0 \\ \hline * & * & I_{2N-r_2} \end{array}\right). \tag{16}$$

This yields L_2 of the form:

$$\left(\begin{array}{c|c} qI_{N_1} & 0 \\ \hline H_2 & I_{r_2-N} \end{array}\right). \tag{17}$$

Here H_2 is a truncated piece of the circulant matrix H corresponding to $\boldsymbol{h}$.

Let us suppose that an attacker must use a minimum of k_1 bits of effort to reduce L_2 until all N_1 of the q-vectors are removed. When this is done and L_2 is put in lower triangular form the entries on the diagonal will have values $\{q^{\alpha_1}, q^{\alpha_2}, \ldots, q^{\alpha_{2N_1}}\}$, where $\alpha_1 + \cdots + \alpha_{2N_1} = N_1$, and the α_i will come very close to decreasing linearly, with

$$1 \approx \alpha_1 > \cdots > \alpha_{2N_1} \approx 0.$$

That is to say, L_2 will roughly obey the geometric series assumption, or GSA. This reduction will translate back to a corresponding reduction of L, which when reduced to lower triangular form will have a diagonal of the form

$$\{q, q, \ldots, q, q^{\alpha_1}, q^{\alpha_2}, \ldots, q^{\alpha_{2N_1}}, 1, 1, \ldots, 1\}.$$

The attacker can mount a brute force attack in a block of size $K = 2N - r_2$. The attacker guesses the coefficients of $\boldsymbol{f}$ that fall into the K block and then uses the reduced basis for L to check if his guess is correct. Standard meet-in-the-middle techniques can be used to effectively square-root the running time of the algorithm. The optimum approach for the attacker is when the effort of lattice reduction is approximately the same as the effort to search via meet-in-the-middle attack[5].

For typical NTRU instantiations, the preprocessing for the hybrid attack requires lattice reduction that achieves a root Hermite factor, δ, satisfying:

$$\det(L_2)^{1/r_2} \cdot \delta^{r_2} \leq q \tag{18}$$

[5] Following [6], for simplicity we assume that if a guess is correct, the attacker will be able to confirm it with a probability $P_s = 100\%$. Note that this is not a guaranteed event due to unavoidable inaccuracies in the lattice reduction process. In fact this will give the attacker several bits of advantage.

	Set #1	Set #2	Set #3	Set #4
N	401	439	593	743
p	3	3	3	3
$\log_2 q$	18	19	19	20
B_s	240	264	300	336
B_t	80	88	100	112
d_1, d_2, d_3	8,8,6	9, 8, 5	10, 10, 8	11, 11, 15
Key & signature size (bytes)	853	988	1335	1765
≈ Prob[accept]	38%	55%	41%	53%
≈ orig. bit security	112	128	192	256
≈ rev. bit security	65	70	110	146

The original parameter sets are vulnerable to the hybrid attack in Section 5.2. See Table 4 for new parameter sets that are robust against those attacks.

Table 2. Original NTRUMLS Parameters

$$\det(L_2)^{1/r_2} \cdot \delta^{-r_2} \geq 2 \tag{19}$$

To achieve optimal attack, we estimate the cost of BKZ 2.0 to achieve the required δ using BKZ 2.0 simulator [2]. In the meantime, we require the cost of lattice reduction with BKZ 2.0 is the same as MITM attack on the last K coefficients. The revised bit security given in Table 2 and bit security of new parameter sets given in Table 4 are derived from this approach.

6 Proposed Parameter Sets and Implementation

We have implemented NTRUMLS and made it available at `https://github.com/NTRUOpenSourceProject/NTRUMLS`. The parameter sets we have implemented are listed in Tables 2 and 3.

The only feature of our implementation not documented above is the use of product form polynomials for $\boldsymbol{f}$ and $\boldsymbol{g}$. Precisely we specify three small integers d_1, d_2, and d_3 and take

$$\begin{aligned} \boldsymbol{f} &= p(\boldsymbol{F}_1 * \boldsymbol{F}_2 + \boldsymbol{F}_3 + 1), \text{and} \\ \boldsymbol{g} &= \boldsymbol{G}_1 * \boldsymbol{G}_2 + \boldsymbol{G}_3 + 1 \end{aligned}$$

where the polynomials $\boldsymbol{F}_i$ and $\boldsymbol{G}_i$ have exactly d_i coefficients equal to $+1$ and d_i coefficients equal to -1. The extra constant terms are to ensure that $\boldsymbol{f}(1) \neq 0$ and $\boldsymbol{g}(1) \neq 0$. Product form keys were introduced to NTRUEncrypt in [10].

6.1 Revised parameter sets

As noted in Section 5.2, the original parameter sets $\{\#1, \#2, \#3, \#4\}$ are vulnerable to the hybrid attack. We present a new version of the parameter sets $\{\#5, \#6, \#7, \#8, \#9\}$ that are robust against those attacks.

	Set #1	Set #2	Set #3	Set #4
KeyGen (μs)	2431	2928	5183	7855
Sign (μs)	575	436	1033	1000
Verify (μs)	92	102	179	231

Table 3. Performance results. Average time for each operation, in microseconds, over 10000 iterations. Code was run on an Intel Core i7-2640M. More extensive benchmarks on a variety of machines are available at `http://bench.cr.yp.to/`.

	Set #5	Set #6	Set #7	Set #8	Set #9
N	401	443	563	743	907
p	3	3	3	3	3
$\log_2 q$	15	16	16	17	17
B_s	138	138	174	186	225
B_t	46	46	58	62	75
d_1, d_2, d_3	8, 8, 6	9, 8, 5	10, 9, 8	11, 11, 6	13, 12, 7
Public Key size (bytes)	752	886	1126	1579	1927
Signature size (bytes)	702	831	1056	1486	1814
≈ Prob[accept]	1%	8%	2%	6%	2%
≈ bit security	82	88	126	179	269

Table 4. Revised NTRUMLS Parameters

In addition, in those new parameter sets, we uses a smaller modulus (i.e, $q = 2^{16}$ for $N = 563$ in #7 while $q = 2^{19}$ for $N = 593$ in #3). This allows for a smaller signature size. In the meantime, smaller q makes it harder for an attacker to find short vectors within the lattices of a same dimension, which increase the overall security of the scheme.

However, as q getting smaller, it also becomes harder to generate a valid signature (the probability of vectors satisfying Eq. 10 is reduced). Therefore, a more aggressive approach is adopted during the rejection sampling. A larger number of candidate signatures (over 90%) are rejected. A recent work [1] shows that due to the highly parallelism during rejection sampling process, the performance can be improved drastically using GPU acceleration.

References

1. Wei Dai, John Schanck, Berk Sunar, William Whyte and Zhenfei Zhang, *NTRU Modular Lattice Signature Scheme on CUDA GPUs*, TBA.
2. Yuanmi Chen and Phong Q. Nguyen, *BKZ 2.0: Better Lattice Security Estimates*, Advances in Cryptology – ASIACRYPT 2011, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 1–20.

3. Léo Ducas and Phong Q. Nguyen, *Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures*, Advances in Cryptology ASIACRYPT 2012 (Xiaoyun Wang and Kazue Sako, eds.), Lecture Notes in Computer Science, no. 7658, Springer Berlin Heidelberg, January 2012, pp. 433–450.
4. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan, *Trapdoors for hard lattices and new cryptographic constructions*, Proceedings of the 40th annual ACM symposium on Theory of computing (New York, NY, USA), STOC '08, ACM, 2008, p. 197206.
5. Nick Howgrave-Graham, *A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU*, Proceedings of the 27th Annual International Cryptology Conference, CRYPTO 2007, Santa Barbara, CA, USA, August 19-23, 2007, p.150169.
6. Jeff Hoffstein, Jill Pipher, John M. Schanck, Joseph H. Silverman, William Whyte and Zhenfei Zhang, *Choosing Parameters for NTRUEncrypt*, Cryptology ePrint Archive, Report 2015/708, `http://eprint.iacr.org/`.
7. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman, *NTRU: a ring-based public key cryptosystem*, Algorithmic number theory (Portland, OR, 1998), Lecture Notes in Comput. Sci., vol. 1423, Springer, Berlin, 1998, pp. 267–288.
8. ———, *An Introduction to Mathematical Cryptography*, Undergraduate Texts in Mathematics, Springer, New York, 2008.
9. Jeffrey Hoffstein and Joseph Silverman, *Optimizations for NTRU*, Public-key cryptography and computational number theory (Warsaw, 2000), de Gruyter, Berlin, 2001, pp. 77–88.
10. Jeffrey Hoffstein and Joseph H. Silverman, *Random small Hamming weight products with applications to cryptography*, Discrete Applied Mathematics **130** (2003), no. 1, 37–49.
11. Vadim Lyubashevsky, *Lattice-based identification schemes secure under active attacks*, Public key cryptography—PKC 2008, Lecture Notes in Comput. Sci., vol. 4939, Springer, Berlin, 2008, pp. 162–179.
12. ———, *Fiat-Shamir with aborts: applications to lattice and factoring-based signatures*, Advances in cryptology—ASIACRYPT 2009, Lecture Notes in Comput. Sci., vol. 5912, Springer, Berlin, 2009, pp. 598–616.
13. ———, *Lattice signatures without trapdoors*, Advances in cryptology—EUROCRYPT 2012, Lecture Notes in Comput. Sci., vol. 7237, Springer, Heidelberg, 2012, pp. 738–755.
14. Alexander May and Joseph H. Silverman, *Dimension reduction methods for convolution modular lattices*, Cryptography and lattices (Providence, RI, 2001), Lecture Notes in Comput. Sci., vol. 2146, Springer, Berlin, 2001, pp. 110–125.
15. Phong Q. Nguyen and Oded Regev, *Learning a parallelepiped: cryptanalysis of GGH and NTRU signatures*, Advances in cryptology—EUROCRYPT 2006, Lecture Notes in Comput. Sci., vol. 4004, Springer, Berlin, 2006, pp. 271–288.
16. ———, *Learning a parallelepiped: cryptanalysis of GGH and NTRU signatures*, J. Cryptology **22** (2009), no. 2, 139–160.

A Short Vectors in Intersections of Translated Lattices

In this appendix we prove Proposition 4, which relates the problem of finding short vectors in intersections of translated lattices to the problem of finding close vectors in the associated intersection of lattices. We applied this result in Section 5 to the intersection of an NTRU lattice $L_{\boldsymbol{h}}$ and the lattice $p\mathbb{Z}^{2N}$.

Proof (Proof of Propostion 4). (a) The fact that the determinants multiply is a standard fact from the theory of lattices.
(b) We let $D_i = \det(L_i)$ for $i = 1, 2$. We use the fact that for any lattice $L \subset \mathbb{Z}^r$ of determinant D, we have $D\mathbb{Z}^r \subset L$. The assumption that $\gcd(D_1, D_2) = 1$ means that we can find $(x, y) \in \mathbb{Z}$ such that

$$xD_1 + yD_2 = 1.$$

We let

$$e_1 = yD_2 = 1 - xD_1, \qquad e_2 = xD_1 = 1 - yD_2.$$

We now consider the vector

$$\boldsymbol{t} = e_1\boldsymbol{t}_1 + e_2\boldsymbol{t}_2.$$

Then

$$\boldsymbol{t} - \boldsymbol{t}_1 = (e_1 - 1)\boldsymbol{t}_1 + e_2\boldsymbol{t}_2 = -xD_1\boldsymbol{t}_1 + xD_1\boldsymbol{t}_2 \in D_1\mathbb{Z}^r \subset L_1,$$

and similarly,

$$\boldsymbol{t} - \boldsymbol{t}_2 = e_1\boldsymbol{t}_1 + (e_2 - 1)\boldsymbol{t}_2 = yD_2\boldsymbol{t}_1 - yD_2\boldsymbol{t}_2 \in D_2\mathbb{Z}^r \subset L_2.$$

Hence $\boldsymbol{t}$ is in M, so $M \neq \emptyset$.
(c) In order to prove that (11) is a bijection, we will show that

$$\boldsymbol{v} \in L_1 \cap L_2 \quad \Longrightarrow \quad \boldsymbol{v} + \boldsymbol{w}_0 \in M \tag{20}$$

and

$$\boldsymbol{w} \in M \quad \Longrightarrow \quad \boldsymbol{w} - \boldsymbol{w}_0 \in L_1 \cap L_2. \tag{21}$$

For (20), we know that $\boldsymbol{w}_0 \in M$, so by definition of M,

$$\boldsymbol{w}_0 = \boldsymbol{v}_1 + \boldsymbol{t}_1 = \boldsymbol{v}_2 + \boldsymbol{t}_2 \quad \text{with } \boldsymbol{v}_i \in L_1 \text{ and } \boldsymbol{v}_2 \in L_2.$$

Then

$$\boldsymbol{v} + \boldsymbol{w}_0 = \underbrace{(\boldsymbol{v} + \boldsymbol{v}_1)}_{\text{in } L_1} + \boldsymbol{t}_1 = \underbrace{(\boldsymbol{v} + \boldsymbol{v}_2)}_{\text{in } L_2} + \boldsymbol{t}_2,$$

so $\boldsymbol{v} + \boldsymbol{w}_0 \in M$. For (21), we write the given $\boldsymbol{w} \in M$ as

$$\boldsymbol{w} = \boldsymbol{v}_1' + \boldsymbol{t}_1 = \boldsymbol{v}_2' + \boldsymbol{t}_2 \quad \text{with } \boldsymbol{v}_i' \in L_1 \text{ and } \boldsymbol{v}_2' \in L_2.$$

Then

$$\boldsymbol{w} - \boldsymbol{w}_0 = \underbrace{\boldsymbol{v}_1' - \boldsymbol{v}_1}_{\text{in } L_1} = \underbrace{\boldsymbol{v}_2' - \boldsymbol{v}_2}_{\text{in } L_2},$$

so $\boldsymbol{w} - \boldsymbol{w}_0 \in L_1 \cap L_2$.
(d) We are given that $\boldsymbol{w}_0, \boldsymbol{w}' \in M$ and that

$$\|\boldsymbol{w}'\|_2 = \min_{\boldsymbol{w} \in M \smallsetminus \boldsymbol{0}} \|\boldsymbol{w}\|_2.$$

To ease notation, we set

$$\boldsymbol{v}' = \boldsymbol{w}_0 - \boldsymbol{w}'.$$

We know from (c) that $\boldsymbol{w}' - \boldsymbol{w}_0 \in L_1 \cap L_2$, and $L_1 \cap L_2$ is a lattice, so $\boldsymbol{v}' \in L_1 \cap L_2$. We estimate

$$\begin{aligned} &\|\boldsymbol{v}' - \boldsymbol{w}_0\|_2 \\ &\quad = \|\boldsymbol{w}'\|_2 \quad \text{by definition of } \boldsymbol{v}', \\ &\quad = \min_{\boldsymbol{w} \in M \smallsetminus \boldsymbol{0}} \|\boldsymbol{w}\|_2 \quad \text{by definition of } \boldsymbol{w}', \\ &\quad = \min_{\boldsymbol{v} \in (L_1 \cap L_2) \smallsetminus \boldsymbol{w}_0} \|-\boldsymbol{v} + \boldsymbol{w}_0\|_2 \quad \text{since (c) says } M = (L_1 \cap L_2) + \boldsymbol{w}_0. \end{aligned}$$

Hence if $\boldsymbol{w}_0 \notin L_1 \cap L_2$, then we have shown that

$$\|\boldsymbol{v}' - \boldsymbol{w}_0\|_2 = \min_{\boldsymbol{v} \in (L_1 \cap L_2)} \|\boldsymbol{v} - \boldsymbol{w}_0\|_2,$$

which is the desired result.

Finally, suppose that $\boldsymbol{w}_0 \in L_1 \cap L_2$. Since also

$$\boldsymbol{w}_0 \in M = (L_1 + \boldsymbol{t}_1) + (L_2 + \boldsymbol{t}_2),$$

we can write

$$\boldsymbol{w}_0 = \boldsymbol{v}_1 + \boldsymbol{t}_1 \quad \text{and} \quad \boldsymbol{w}_0 = \boldsymbol{v}_2 + \boldsymbol{t}_2 \quad \text{with } \boldsymbol{v}_1 \in L_1 \text{ and } \boldsymbol{v}_2 \in L_2.$$

But then $\boldsymbol{t}_1 = \boldsymbol{w}_0 - \boldsymbol{v}_1 \in L_1$ and $\boldsymbol{t}_2 = \boldsymbol{w}_0 - \boldsymbol{v}_2 \in L_2$, contradicting the initial assumption on $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$. Hence $\boldsymbol{w}_0 \notin L_1 \cap L_2$, which completes the proof of Proposition 4.