NTWE: A Natural Combination of NTRU and LWE

Joel Gärtner

May 9, 2023

Abstract

Lattice-based cryptosystems are some of the primary post-quantum secure alternatives to the asymmetric cryptography that is used today. These lattice-based cryptosystems typically rely on the hardness of some version of either the NTRU or the LWE problem. In this paper, we present the NTWE problem, a natural combination of the NTRU and LWE problems, and construct a new lattice-based cryptosystem based on the hardness of the NTWE problem.

As with the NTRU and LWE problems, the NTWE problem naturally corresponds to a problem in a q-ary lattice. This allows the hardness of the NTWE problem to be estimated in the same way as it is estimated for the LWE and NTRU problems. We parametrize our cryptosystem from such a hardness estimate and the resulting scheme has performance that is competitive with that of typical lattice-based schemes.

In some sense, our NTWE-based cryptosystem can be seen as a less structured and more compact version of a cryptosystem based on the module-NTRU problem. Thus, parameters for our cryptosystem can be selected with the flexibility of a module-LWE-based scheme, while other properties of our system are more similar to those in an NTRU-based system.

1 Introduction

This NIST standardization process for post-quantum cryptography has already resulted in four algorithms being selected for standardization. Three of these selected algorithms are lattice-based and the security of these schemes rely on the hardness of versions of either the LWE or the NTRU problem. The origins of the NTRU and LWE problems are quite different, but the concrete hardness of these problems is currently estimated in very similar ways.

NTRU was introduced more than 25 years ago as a ring-based public key cryptosystem [12]. The security of this system is based on the hardness of the NTRU problem which, with somewhat different parameters than those first proposed, has remained hard to solve in practice. While not originally stated as a lattice-based cryptosystem, an NTRU instance can easily be interpreted as an instance a special type of structured lattice problem and the concrete security of current NTRU-based cryptosystems is typically estimated based on how efficiently this structured lattice problem can be solved.

The LWE problem is even more closely related to lattice problems. It was introduced in 2005 by Regev together with a quantum reduction from a worst-case lattice problem [19]. As such, the asymptotic security of LWE-based cryptosystems can be guaranteed as long as there exists instances of this lattice problem that are hard to solve with a quantum computer. This reduction does, however, say very little about the concrete security of typically considered parametrizations of LWE-based cryptosystems [4]. Instead, LWE-based cryptosystems are usually parametrized in the same way as NTRU-based systems, based on a concrete hardness estimate for the natural lattice problem that corresponds to an LWE instance.

There are several different structured versions of the LWE problem, with the most prominent being the ring- and module-LWE problems. Especially the ring-LWE problem is very similar to the NTRU problem, as it essentially corresponds to an inhomogeneous version of the NTRU problem. The module-LWE problem can be seen as a somewhat less structured version of the ring-LWE problem and is thus also quite similar to the NTRU problem.

While there are results that relate the hardness of versions of the LWE problem to the hardness of versions of the NTRU problem [18, 23, 24], these results do not directly relate the security of concrete parametrizations of currently proposed cryptosystems. As such, given similar parametrization of an NTRU-based and an LWE-based cryptosystem, we can not directly determine if one of these schemes is more secure than the other. Although the security of both types of systems are based on similar assumptions, there is a possibility that an attack lowers the concrete security of schemes based on one of the assumptions, without directly impacting schemes based on the other assumption.

Thus, lattice-based cryptography is primarily based on the hardness of two similar, yet different problems. The worst-case to average-case reduction which were the reason for the introduction of the LWE problem does not support the concrete security of typical LWE-based cryptosystems. Furthermore, the understanding of lattice algorithms have improved significantly since the introduction of the NTRU system. Therefore, it is interesting to investigate what alternative problems we can base the security of similar cryptosystems on, and if this would allow any improvements compared to the schemes that are considered today.

1.1 Our Contribution

In this paper, we introduce and investigate the NTWE problem and create an NTWE-based cryptosystem. The NTWE problem can be seen as a natural combination of the NTRU and module-LWE problems. It is easily seen that as long as either the NTRU problem or the module-LWE problem is hard, then so is the NTWE problem.

We can thus parametrize our NTWE-based cryptosystem so that it is secure as long as either the corresponding NTRU- or LWE-based cryptosystem is secure. As the NTRU and LWE problem are quite similar, one would typically not consider using a system that relies on an module-LWE-based and an NTRU-based cryptosystem in parallel. However, this type of parametrization of our NTWE-based cryptosystem is both more efficient and compact than simply combining an NTRU-based and a module-LWE-based cryptosystem and is therefore more interesting.

While we can guarantee that the NTWE problem is no easier than versions of the NTRU and the LWE problems, it actually seems to be significantly harder than the problems that we provably can relate it to. Similarly to the module-LWE problem, we consider a module versions of the NTWE problem. A simple reduction shows that the rank k NTWE problem is no easier than the rank k module-LWE problem. However, we believe that the rank k NTWE problem is essentially as hard as a rank $k + 1$ module-LWE problem.

Similarly to the LWE and NTRU problems, the NTWE problem naturally corresponds to a lattice problem in a q-ary lattice. The lattice problem for the rank k NTWE problem in is very similar to the lattice problem for the rank $k + 1$ module-LWE problem. This motivates our concrete hardness estimate for the rank k NTWE problem. Furthermore, we are able to show that a more structured version of the NTWE problem is at least as hard as the rank $k + 1$ module-NTRU problem, providing further motivation for our hardness estimate.

New hardness assumptions must be thoroughly analyzed before significant confidence can be placed in the security of cryptosystems that rely on them. However, any assumptions similar to the ones used today can be more easily trusted. As the NTWE problem naturally corresponds to a lattice problem, it directly benefits from analysis of similar lattice problems. Furthermore, due to the similarities between the rank k NTWE problem and the NTRU and LWE problems in rank $k + 1$ modules, we believe that any improved algorithms for the NTWE problem would also result in increased understanding of these other problems.

We furthermore provide concrete parametrizations of our new NTWE-based cryptosystem. This includes parametrizations similar to the different parametrizations of CRYSTALS-Kyber [22], henceforth referred to only as Kyber. These parametrizations have essentially the same sized public-key and ciphertext as in the parametrizations of Kyber that target the same security level.

A large reason for the relatively small ciphertexts in Kyber is a method for ciphertext compression. This consists of discarding many of the bits of the ciphertext, allowing significantly smaller ciphertexts at the cost of somewhat larger decryption failure probability. In our scheme, we do not perform any such ciphertext compression but we still have a ciphertext size that is comparable to that of Kyber. Thus, compared to an LWE-based scheme without ciphertext compression, our scheme actually has significantly smaller ciphertexts.

There are multiple reasons to want to avoid ciphertext compression, one of which may be patent reasons. Although the method for ciphertext compression that is used in Kyber has not been patented, other versions of ciphertext compression seem to be protected by a patent. However, to which extent this patent actually cover the different methods for ciphertext compression that is performed in LWE-based schemes has not been entirely clear. This may be a reason to prefer our scheme where no such ciphertext compression is performed.

Another benefit of not having to use ciphertext compression is that it may allow more compact schemes that include a zero-knowledge proof that the ciphertext is correctly formed. Such a zero-knowledge proof seems to be incompatible with ciphertext compression, and such a scheme would therefore have to use uncompressed ciphertexts. Therefore, for these types of applications, the ciphertexts from our cryptosystem would be significantly smaller than the ciphertexts in a comparable LWE-based system. This advantage was one of the primary advantages of NTRU-based systems compared to LWE-based systems mentioned by Lyubashevsky and

Seiler in a paper [16] developing a more efficient version of NTRU.

An advantage of our scheme compared to NTRU is its increased flexibility in allowing module versions of the problem. While module versions of the NTRU problem have been considered [7], this module-NTRU problem does not seem to be suitable for public key encryption. The size of the public key in such a module-NTRU-based encryption scheme would grow with the square of the module rank. This would result in a significantly larger public key than in a module-LWE based scheme, where the size of the public key depends linearly on the module rank. We can thus consider our NTWE-based cryptosystem as a more compact version of a cryptosystem based on the module-NTRU problem. Furthermore, whereas the NTRU problem is significantly easier in an overstretched parameter regime, it does not seem like there is such an overstretched parameter regime for the NTWE problem.

Another potential benefit of an NTWE-based cryptosystem compared to a system based on module-LWE is its resistance against dual lattice attacks. The two main attacks against LWE-based schemes are the primal and dual lattice attacks. Recent results have indicated that the dual attack may be more efficient against concrete cryptosystem parameters [11, 17]. Although these results have been questioned [8], increased resistance against these dual attacks is still preferable.

While it is possible to perform a dual attack against the NTWE problem, a primal lattice attack against the NTWE problem seems to be significantly more efficient for the parameters that we consider in this paper. However, the dual attack against NTWE does seem to be more efficient than the primal attack against some parametrizations of this problem. Thus, the dual attack should still be considered when investigating the concrete hardness of specific NTWE instances.

For efficiency, we parametrize our cryptosystem using a ring R and modulos q that enable using the Number Theoretic Transform (NTT) to efficiently multiply ring elements. Similar to an NTRU-based system, we require computing inverses of a ring element f in both R_q and R_p, for an integer q and a small integer p. While the inverse in R_q is efficiently computable by using the NTT, the inverse in R_p is less efficient to compute. However, this primarily impacts the performance of key generation, and if the same public-key is used multiple times, this cost may be considered insignificant.

To improve the efficiency of key generation, we may select f from a distribution such that the inverse in R_p is trivial, but such that the elements of f are a factor p larger. This trick results in schemes that are more efficient than the corresponding module-LWE-based schemes but with a larger decryption failure probability. If not using this trick to ensure that the inverse of f in R_p is trivial, the resulting scheme actually has a lower decryption failure probability than a corresponding module-LWE-based scheme. Thus, our NTWE-based cryptosystem is either more efficient with a larger decryption failure probability or less efficient with a smaller decryption failure than a corresponding module-LWE-based system.

1.2 Paper Outline

We begin the paper with some background in Section 2. Next, in Section 3, we introduce the NTWE problem and describe its relation to the LWE and NTRU problems.

In Section 4, we consider the concrete hardness of the natural lattice problems that correspond to the NTWE problem. For reference, we also briefly explain how lattice algorithms are used to solve the NTRU and LWE problems.

Next, in Section 5 we present our NTWE-based cryptosystem and compare it to NTRU-based and module-LWE-based cryptosystems and in Section 6 present some concrete parametrizations of this cryptosystem.

Finally, in section 7, we have some final remarks, including a note regarding how investigating the concrete hardness of the NTWE problem could be interesting also due to implications for the LWE and NTRU problems.

2 Background

2.1 Notation

We denote real matrices by bold upper case letters $\boldsymbol{A}, \boldsymbol{B}$ and real vectors by bold lower case letters $\boldsymbol{s}, \boldsymbol{e}$. Vectors and matrices over a number field are denoted similarly, but with the letters overlined $\overline{\boldsymbol{A}}, \overline{\boldsymbol{s}}$.

We denote probability distributions by calligraphic letters $\mathcal{U}$ or by Greek letters ψ. In particular, we denote the uniform probability distribution over a set S by $\mathcal{U}(S)$.

For an arbitrary distribution ψ over a ring R, we let ψ^* be the distribution corresponding to invertible elements of ψ. Rejection sampling from ψ, rejecting all non-invertible elements, allows sampling from this distribution. For the rings relevant in this work, only a small portion of the elements are not invertible.

2.2 Lattices

A lattice L is a discrete subgroup of $\mathbb{R}^d$. A lattice can always be described by a basis $\boldsymbol{B} \in \mathbb{R}^{d \times k}$ for $k \leq d$ with $L = L(\boldsymbol{B}) = \{\boldsymbol{Bx} : \boldsymbol{x} \in \mathbb{Z}^k\}$. The determinant of a lattice L is given by $\sqrt{\det(\boldsymbol{B}^T\boldsymbol{B})}$ for an arbitrary basis $\boldsymbol{B}$ of L.

We denote the length of the shortest non-zero vector in a lattice L by $\lambda_1(L)$. For a random d dimensional lattice, we expect the so called Gaussian Heuristic to hold. This heuristic predicts that the number of lattice points in a ball of volume V is $V/\det(L)$, which corresponds to estimating that

$$\lambda_1 \approx \mathrm{gh}(L) = \sqrt{\frac{d}{2\pi e}} \det(L)^{1/d} \quad . \tag{1}$$

The Gaussian Heuristic is often assumed to approximately hold even in some lattices that are not sampled uniformly at random, such as in q-ary lattices.

2.3 Algebraic Number Theory

A number field K is a finite-degree field extension of the rational numbers $\mathbb{Q}$. This corresponds to $K = \mathbb{Q}(\zeta)$, the rational numbers adjoined with some element ζ that satisfies $f(\zeta) = 0$ for some irreducible polynomial $f \in \mathbb{Q}[x]$. This polynomial is called the minimal polynomial of ζ and the degree of the number field K is the degree of this polynomial. In this work, n denotes the degree of number fields where applicable.

The ring of integers $\mathcal{O}_K$ for a number field K is the set of algebraic integers in K, meaning that it is the elements in K that are a root of some monic polynomial in $\mathbb{Z}[x]$. For the concrete number fields we are considering in this paper, the ring of integers for the number field $K = \mathbb{Q}(\zeta)$ is always equal to $\mathbb{Z}(\zeta)$, but this is not the case in general. In particular, we only consider rings isomorphic to $\mathbb{Z}[X]/(X^n + 1)$ for $n = 2^\ell$ for some integer ℓ. These are the rings of integers of power of two cyclotomic fields.

As we only consider rings of integers of power of two cyclotomic fields, a coefficient representation of elements in $\mathcal{O}_K$ is suitable. As such, we represent an element $\overline{\boldsymbol{v}} \in \mathcal{O}_K$ by the vector $\boldsymbol{v} \in \mathbb{Z}^n$ containing the coefficients of its natural representative in $\mathbb{Z}[X]/(X^n + 1)$. We let the norm $\|\overline{\boldsymbol{v}}\|$ be given by the ℓ_2 norm of the coefficient vector, which we extend to modules $\mathcal{O}_K^k$ in the natural way. For an element $\overline{\boldsymbol{v}}$ in R^k, we also consider a corresponding matrix in $\mathbb{Z}^{kn \times n}$, given by the coefficient vectors of $\overline{\boldsymbol{v}}X^i$ for every integer i with $0 \leq i < n$.

2.4 LWE and NTRU

The version of the Learning With Errors (LWE) problem considered in this work is defined in terms of a module-LWE distribution, as defined below.

Definition 1 (Module-LWE distribution)**.** Let q be a prime and R the ring of integers for a number field K. For $\overline{\boldsymbol{s}} \in R_q^k$ and ψ some distribution on R_q, a sample from the module-LWE distribution $A_{\overline{\boldsymbol{s}},\psi}$ is given by $(\overline{\boldsymbol{a}}, b = \overline{\boldsymbol{a}} \cdot \overline{\boldsymbol{s}} + e)$, where $\overline{\boldsymbol{a}} \leftarrow \mathcal{U}(R_q^k)$ and $e \leftarrow \psi$.

In the original definition of module- and ring-LWE distributions and problems, the secrets have elements in the dual ideal $R_q^\vee$ and the error distribution has a continuous support. However, using secret elements in R_q is equivalent when R is the ring of integers of a power of two cyclotomic number field [15]. Furthermore, it is easily seen that the problem with a dicretized error distribution is no easier than the original problem with a continuous distribution.

The version of the LWE problem that is relevant in this work is the normal-form decision module-LWE problem, as defined next.

Definition 2 (Normal form decision module-LWE problem)**.** Let q be a prime, R the ring of integers for a number field K and ψ be some distribution on R_q. Then, the normal form decision module-LWE problem is to distinguish samples from $A_{\overline{\boldsymbol{s}},\psi}$ from uniformly random in $R_q^k \times R_q$ when $\overline{\boldsymbol{s}}$ is a vector with elements sampled from ψ.

We also use the following definition of a decision version of the NTRU problem, where multiple samples are provided from a distribution. The NTRU problem is not typically considered in terms of such a distribution from which it is possible to get multiple samples. However, this multi-sample problem has been considered previously and does not seem to be significantly easier than the traditional, single sample, NTRU problem. Our definition is for a module-version of the NTRU problem and a more traditional NTRU problem is recovered with module rank $k = 1$.

Definition 3 (Decision module-NTRU problem)**.** Let k be some integer, q be some prime, R be the ring of integers for some number field K and ψ be some distribution on R_q. Let $\overline{\boldsymbol{F}} \in R_q^{k \times k}$ have elements sampled from ψ and assume that $\overline{\boldsymbol{F}}$ is invertible. Then, the rank k decision module-NTRU problem is to distinguish samples of the form $\overline{\boldsymbol{h}} = \overline{\boldsymbol{g}} \cdot \overline{\boldsymbol{F}}^{-1} \in R_q^k$ from uniformly random in R_q^k where $\overline{\boldsymbol{g}} \leftarrow \psi^k$.

2.5 Lattice Reduction

In practice, the most efficient algorithm for finding relatively short vectors in a lattice is the lattice reduction algorithm BKZ [21, 20]. BKZ works by iteratively improving the lattice basis by solving SVP instances in projected sublattices of dimension β.

The effectiveness of BKZ is often estimated through its Hermite factor δ_β, with BKZ finding a vector of length $\delta_\beta^d \det(L)^{1/d}$ in a d-dimensional lattice L. The specific value of this factor depends on the blocksize β BKZ is used with. A typical estimate is that

$$\delta_\beta = \left(\frac{\beta}{2\pi e}(\pi\beta)^{1/\beta}\right)^{\frac{1}{2(\beta-1)}} \tag{2}$$

which is heuristically proven to be the asymptotic performance of BKZ on random lattices [6].

In NTRU and LWE lattices, the secret vectors are significantly shorter than would be expected in a random lattice. This enables BKZ to recover the secret vector faster than a simple estimate based on δ_β predicts. Instead, when estimating the hardness of these problems, one often considers the so called 2016 estimate [1] that predicts that BKZ with block size β finds an unusually short vector $\boldsymbol{v}$ in an d-dimensional lattice L if

$$\frac{\sqrt{\beta} \cdot \|\boldsymbol{v}\|}{\sqrt{d}} \leq \delta_\beta^{2\beta-d} \det(L)^{1/d} \quad . \tag{3}$$

A conservative estimate for the cost of using BKZ with block size β is the core SVP hardness, as introduced in [1]. This estimates that running BKZ with block-size β is no more expensive than solving a single SVP instance in dimension β. We further estimate the hardness of SVP in dimension β based on the performance of known algorithms.

For our parametrizations, we consider the performance of the best known classical algorithm for solving SVP, ignoring its memory requirements and subexponential factors in its running time. This performance is given by a heuristic algorithm [2] with complexity $\sqrt{3/2}^{\beta} \approx 2^{0.292\beta}$ for lattice dimension β. We represent the core SVP hardness by the logarithm of this, and thus given by 0.292β.

There are quantum algorithms that solve SVP more efficiently than this algorithm. However, these algorithms improve attacks against lattice-based cryptosystems less than Grover's quantum search algorithm improves attacks against symmetric primitives with comparable classical security. As such, when comparing the security of a lattice-based cryptosystem with the security of a symmetric primitive, the performance of current quantum attacks does not have to be considered. In this work, we therefore do not consider these quantum lattice algorithms, but we still claim that our system is post-quantum secure.

3 The NTWE Problem

The NTWE problem combines the NTRU and LWE problems in a natural way. Similarly to the NTRU problem, an instance of the NTWE problem is of the form $h = gf^{-1}$ where $g \leftarrow \psi_1$ and $f \leftarrow \psi_2$. However, unlike standard NTRU instances, we do not use $\psi_1 = \psi_2$, nor do we expect g to be a small element. Instead, we let the distribution ψ_1 be a module-LWE distribution and samples from this distribution are thus expected to be hard to distinguish from uniformly random.

A more formal definition of this problem follows, where we, similarly to the definition of the LWE problem, consider it in terms of an NTWE distribution. As with the module-LWE problem, we primarily consider the problem with the rank k some small integer, while the degree n of the underlying ring is fixed to some power of two, such as 256.

Definition 4 (NTWE distribution $\mathcal{W}(\overline{\boldsymbol{s}}, f, \psi)$)**.** Let q be a prime, k be an integer, n be some power of 2, $R = \mathbb{Z}[X]/(X^n + 1)$ and ψ be some distribution on R_q. Furthermore, let $\overline{\boldsymbol{s}}$ be a vector in R_q^k and f be an invertible element in R_q. A sample from the NTWE distribution $\mathcal{W}(\overline{\boldsymbol{s}}, f, \psi)$ is given by

$$(\overline{\boldsymbol{a}}, b = (\overline{\boldsymbol{a}} \cdot \overline{\boldsymbol{s}} + e)f^{-1}) \in R_q^k \times R_q$$

where $\overline{\boldsymbol{a}} \leftarrow \mathcal{U}(R_q^k)$ and $e \leftarrow \psi$.

We consider an average case distribution of problem instances where the secret vector $\overline{\boldsymbol{s}}$ and secret element f are sampled from the error distribution ψ. This is similar to the normal-form module-LWE problem. The definition of the search and decision versions of this average-case problem follows.

Definition 5 (Decision NTWE problem (DNTWE(ψ, h)))**.** Let ψ be some distribution on R_q and let h be some integer. An instance of the Decision NTWE problem DNTWE(ψ, h) is given by an unknown distribution $\mathcal{D}$ that is either uniformly random or the $\mathcal{W}(\overline{\boldsymbol{s}}, f, \psi)$ distribution for some $\overline{\boldsymbol{s}} \leftarrow \psi^k$ and $f \leftarrow \psi^*$. The DNTWE(ψ, h) problem is to determine which is the case when given at most h samples from the unknown distribution.

For the search version of the NTWE problem, the actual secrets $\overline{\boldsymbol{s}}$, f used to generate the NTWE distribution need not be recovered. Instead it suffices to recover $\overline{\boldsymbol{s}}X^i$ and fX^i for some i as these alternative solutions would generate the same NTWE distribution as the actual secrets. Furthermore, for the rings R and error distributions ψ we consider, all of these solutions are equally likely to be sampled as secrets for the problem instance.

Definition 6 (Search NTWE problem SNTWE(ψ, h))**.** Let ψ be some distribution on R_q and h be some integer. An instance of the Search NTWE problem SNTWE(ψ, h) is to recover $\overline{\boldsymbol{s}}X^i$ and fX^i, for some i, when given at most h samples from the $\mathcal{W}(\overline{\boldsymbol{s}}, f, \psi)$ distribution, where $\overline{\boldsymbol{s}} \leftarrow \psi^k$ and $f \leftarrow \psi^*$.

3.1 Relation to Other Problems

It is easily seen that an instance of the search/decision NTWE problem is at least as hard as an instance of the search/decision rank k module-LWE problem. This relation is formalized in the following lemma:

Lemma 3.1. *Assume that there is an algorithm W that is able to solve the (search/decision) NTWE problem with advantage ε. Then, using W once, with a negligible amount of additional computations, provides a solution to the corresponding (search/decision) normal form rank k module-LWE problem with advantage ε.*

Proof. Given an algorithm that solves the NTWE problem, we can easily solve the corresponding module-LWE problem. This is accomplished by sampling $f \leftarrow \psi^*$ and transforming samples from the input distribution in the module-LWE problem instance into $(\overline{\boldsymbol{a}}, bf^{-1})$.

If the input samples are from a module-LWE distribution, the transformed samples are from an NTWE distribution. With these samples as input, an algorithm that solves the search NTWE problem recovers $\overline{\boldsymbol{s}}X^i$ and fX^i for some i. As f is known, this allows recovering $\overline{\boldsymbol{s}}$ and solving the search module-LWE problem.

If instead the input samples are from an uniformly random distribution, the transformed samples are also from a uniformly random distribution. As such, using an algorithm that solves the decision NTWE problem a single time provides a solution to the decision module-LWE problem with the same advantage. □

It is also easily seen that the NTWE problem is no easier than a similarly parametrized version of the rank 1 module-NTRU problem.

Lemma 3.2. *Assume that there is an algorithm W that, when given h samples from the input distribution, is able to solve the (search/decision) NTWE problem with advantage ε. Then, using W once, with a negligible amount of additional computations, provides a solution to a rank 1 (search/decision) NTRU problem with advantage ε. This is accomplished by using $k + h$ samples from the input distribution in the given instance of the NTRU problem.*

Proof. Let $\overline{\boldsymbol{h}} = \overline{\boldsymbol{g}}f^{-1}$ be $h + k$ samples from the input distribution for the NTRU problem. By splitting $\overline{\boldsymbol{h}}$ as $(\overline{\boldsymbol{s}}f^{-1}, \overline{\boldsymbol{e}}f^{-1}) \in R_q^k \times R_q^h$ and letting $\overline{\boldsymbol{A}} \leftarrow \mathcal{U}(R_q^{h\times k})$, we can calculate $(\overline{\boldsymbol{A}}\overline{\boldsymbol{s}} + \overline{\boldsymbol{e}})f^{-1}$. If the input is an NTRU distribution, this directly corresponds to h samples from an NTWE distribution. If instead the input is uniformly random, then so are the resulting h samples.

As such, any algorithm that solves the DNTWE(ψ, h) can be used to solve the decision NTRU problem by using $h + k$ NTRU samples. Similarly, any algorithm that solves the SNTWE(ψ, h) problem can be used to solve the search NTRU problem by using $h + k$ NTRU samples. □

Lemma 3.2 ensures that the NTWE problem in rank k modules is at least as hard as the rank 1 NTRU problem with multiple samples. We do, however, expect something significantly stronger to hold, namely that the rank k NTWE problem is at least as hard as the rank $k + 1$ module-NTRU problem. Lemma 3.3 below provides motivation for such a statement, as it shows that if we can solve a special version of the rank k NTWE problem, then we can also solve the rank $k + 1$ module NTRU problem.

This special version of the NTWE problem differs from an ordinary NTWE problem by using an $\overline{\boldsymbol{a}}$ that is not sampled uniformly at random and instead from some other distribution. The specific distribution for which $\overline{\boldsymbol{a}}$ is sampled from in these special NTWE instances is directly given by a rank $k + 1$ module-NTRU instance. As we do not have a good definition for this distribution besides for how it appears in the proof, we only define it as a part of the proof.

Although it is possible that the NTWE problem where $\overline{\boldsymbol{a}}$ is non-uniform is a harder problem than NTWE with uniformly random $\overline{\boldsymbol{a}}$, we have no reason to expect this to be the case. Instead, it seems more natural to assume the opposite, that samples with uniformly random $\overline{\boldsymbol{a}}$ are harder to distinguish from uniformly random than those with $\overline{\boldsymbol{a}}$ from some other distribution. As such, we consider this lemma to be a strong argument for why the concrete hardness of the rank k NTWE problem should be comparable to that of rank $k+1$ module-NTRU. However, this lemma does not actually prove that the rank k NTWE problem is at least as hard as a rank $k+1$ NTRU problem.

Lemma 3.3. *Assume that there is an algorithm W that solves a decision version of the NTWE problem where $\overline{\boldsymbol{a}}$ is not uniformly random and instead sampled from a special distribution, defined in the proof. If W achieves an advantage ε, then, using W once, with a negligible amount of additional computation, provides a solution to the decision rank $k+1$ module-NTRU problem with advantage ε. This is accomplished by using h module-NTRU samples.*

Proof. We claim that a sample from a rank $k+1$ module-NTRU instance $\overline{\boldsymbol{h}} = \overline{\boldsymbol{g}}\overline{\boldsymbol{F}}^{-1}$ corresponds to a sample from a rank k NTWE instance with a special structure on $\overline{\boldsymbol{a}}$. The $\overline{\boldsymbol{a}}$ for this sample is the negation of the first k elements of $\overline{\boldsymbol{h}}$, while the b part of the NTWE sample is the final element of $\overline{\boldsymbol{h}}$.

To see this, we write $\overline{\boldsymbol{e}} - \overline{\boldsymbol{h}}\overline{\boldsymbol{F}} = 0$ with $\overline{\boldsymbol{e}} = \overline{\boldsymbol{g}}$ and split $\overline{\boldsymbol{h}}$ into $(-\overline{\boldsymbol{a}}, b)$. Next, we rename the $k \times (k+1)$ dimensional submatrix of $\overline{\boldsymbol{F}}$ that we multiply with $\overline{\boldsymbol{a}}$ to $\overline{\boldsymbol{S}}$, while the remaining $k+1$ dimensional row we call $\overline{\boldsymbol{f}}$. Thus, $\overline{\boldsymbol{a}}\overline{\boldsymbol{S}} - b\overline{\boldsymbol{f}} + \overline{\boldsymbol{e}} = \overline{\boldsymbol{0}}$, which corresponds to a sample from $k+1$ different NTWE instance. These samples share the same $\overline{\boldsymbol{a}}$ and have the same resulting b, but each NTWE instance uses different secrets $\overline{\boldsymbol{s}}, f$ and errors e. This is seen by considering a single element, which is given by $\overline{\boldsymbol{a}}\overline{\boldsymbol{s}} - bf + e = 0$, or equivalently $(\overline{\boldsymbol{a}}\overline{\boldsymbol{s}} + e)f^{-1} = b$, if f is invertible.

Additional samples from the NTRU distribution also result in NTWE samples with the same $\overline{\boldsymbol{s}}$ and f, but with different $\overline{\boldsymbol{a}}$ and e. Furthermore, note that f, e and the elements in $\overline{\boldsymbol{s}}$ are sampled from ψ, as expected for the NTWE instance.

The distribution of $\overline{\boldsymbol{a}}$ in this constructed NTWE distribution is not uniformly random and instead given by the first k elements from a sample of the NTRU distribution. As such, each NTRU sample corresponds to a sample from an NTWE instance where the $\overline{\boldsymbol{A}}$ matrix is generated with rows given by samples from an NTRU distribution. Furthermore, in this NTWE instance, the secrets $\overline{\boldsymbol{s}}$ and f are part of the matrix $\overline{\boldsymbol{F}}$ used to define the NTRU distribution used to generate the $\overline{\boldsymbol{A}}$ matrix.

If instead given a sample from a uniformly random distribution, splitting the sample into $-\overline{\boldsymbol{a}}$ and b obviously results in $\overline{\boldsymbol{a}}$ and b that are uniformly random. Thus, if we are able to distinguish this special NTWE instance from uniformly random, then we are also able to distinguish a rank $k+1$ module-NTRU instance from uniformly random. □

4 The NTWE Lattice Problems

Both the NTRU problem and the module-LWE problem can be solved by considering the naturally corresponding lattice problems. Currently, this approach leads to the best performing algorithms for solving these problems, and we recall the techniques used below. We expect that the NTWE problem similarly is best solved by considering lattice problems that naturally correspond to the NTWE problem. We present these NTWE lattice problems in subsection 4.3 below and compare it to lattice problems in corresponding NTRU and LWE lattices.

We can not guarantee that there are no other, more efficient, attacks against the NTWE problem than the ones we consider here. However, its similarity to the LWE and NTRU problems motivates us to focus on attacks similar to the best performing attacks against these problems.

In some sense, the NTWE lattice is a mix between an NTRU lattice and a module-LWE lattice. We therefore believe that it is likely that any improved attacks against the NTWE problem would improve our understanding of the NTRU and LWE problems. In particular, we believe that a specialized attack against the NTWE problem would likely have interesting implications for both the NTRU and LWE problem, in some sense indicating that these problems are hard, but mixing them results in an easier problem.

4.1 NTRU Problem

In the module-NTRU problem, the input is a matrix $\overline{\boldsymbol{H}} \in R_q^{h\times k}$. In the search version of the problem, the task is to recover $\overline{\boldsymbol{G}} \in R_q^{h\times k}, \overline{\boldsymbol{F}} \in R_q^{k\times k}$ such that $\overline{\boldsymbol{G}\boldsymbol{F}}^{-1} = \overline{\boldsymbol{H}}$ and such that all elements in $\overline{\boldsymbol{G}}$ and $\overline{\boldsymbol{F}}$ are small. To state this as a lattice problem, we consider the integer matrices $\boldsymbol{F} \in \mathbb{Z}^{kn\times kn}$ and $\boldsymbol{G}, \boldsymbol{H} \in \mathbb{Z}^{hn\times kn}$ corresponding to $\overline{\boldsymbol{F}}, \overline{\boldsymbol{G}}, \overline{\boldsymbol{H}}$. Then, it can be seen that the $(h+k)n$-dimensional lattice spanned by the columns of

$$\begin{bmatrix} q\boldsymbol{I} & \boldsymbol{H} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

contains a dense kn-dimensional sublattice given by

$$\begin{bmatrix} q\boldsymbol{I} & \boldsymbol{H} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \star \\ \boldsymbol{F} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{F} \end{bmatrix}$$

where $\star$ represents the matrix corresponding to modular reduction. Using lattice reduction methods, this dense sublattice can be found, which solves both the search and decision NTRU problems.

Depending on the specific parametrization, the lattice reduction algorithms may directly find vectors that directly corresponds to elements of $\overline{\boldsymbol{F}}$ and $\overline{\boldsymbol{G}}$. If the parameters are chosen in an overstretched regime, the lattice reduction may first find other vectors in the dense sublattice [9]. In either case, finding unusually short vectors in the lattice solves the decision NTRU problem and quickly leads to an attack against NTRU-based cryptosystems.

4.2 Module-LWE Problem

The Module-LWE problem is typically solved by using lattice reduction algorithms on one of two different types lattices, corresponding to the dual and primal lattice attacks.

A Module-LWE instance with h samples is given by a uniformly random $\overline{\boldsymbol{A}} \in R_q^{h\times k}$ and the vector $\overline{\boldsymbol{b}} = \overline{\boldsymbol{A}}\overline{\boldsymbol{s}} + \overline{\boldsymbol{e}} \in R_q^h$, where the elements of $\overline{\boldsymbol{e}}$ are sampled from the error distribution ψ. In the normal form version of the problem, the elements of the secret vector $\overline{\boldsymbol{s}}$ are also sampled from ψ. The lattices corresponding to this module-LWE instance are given by the integer matrices $\boldsymbol{A} \in \mathbb{Z}^{hn\times kn}$ and $\boldsymbol{B} \in \mathbb{Z}^{hn\times n}$ corresponding to $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{b}}$.

In the primal attack, the relevant lattice is spanned by the columns of

$$\begin{bmatrix} q\boldsymbol{I} & \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & t\boldsymbol{I} \end{bmatrix}$$

where t is a constant typically chosen to be 1. Furthermore, one typically do not consider this full $(h + k + 1)n$ dimensional lattice, instead only considering a single column $\boldsymbol{b}$ of the full matrix $\boldsymbol{B}$, giving a $(h + k)n + 1$ dimensional lattice. Using lattice reduction, we can find the short lattice vector

$$\begin{bmatrix} \boldsymbol{e} \\ -\boldsymbol{s} \\ t \end{bmatrix} = \begin{bmatrix} q\boldsymbol{I} & \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & t \end{bmatrix} \begin{bmatrix} \star \\ -\boldsymbol{s} \\ 1 \end{bmatrix}$$

where $\star$ gives the modular reductions and $\boldsymbol{s}$ and $\boldsymbol{e}$ are integer vectors representing $\overline{\boldsymbol{s}}$ and $\overline{\boldsymbol{e}}$ respectively. This short vector directly solves both the decision and search Module-LWE problems.

In the dual attack, the lattice given by

$$L^{\perp} = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{Z}^{(h+k)n} : \boldsymbol{x}\boldsymbol{A} = \boldsymbol{y} \mod q\}$$

is considered. The dual attack is based on using lattice reduction in order to find short vectors in $L^{\perp}$. Short vectors in $L^{\perp}$ can be used to distinguish between samples from a module-LWE instance and samples from a uniformly random distribution, solving the decision module-LWE problem.

Given a short vector $\boldsymbol{w}$ in $L^{\perp}$, the attack works by multiplying samples $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$ with $\boldsymbol{w}$, resulting in $\boldsymbol{w}\boldsymbol{b} = \boldsymbol{y} \cdot \boldsymbol{s} + \boldsymbol{x} \cdot \boldsymbol{e}$ which is small if $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}$ and $\boldsymbol{e}$ all are short. By using these short lattice vectors, samples from a module-LWE distribution are thus transformed into small integers. Meanwhile, multiplying a uniformly random $\boldsymbol{b}$ with such a short vector results in a uniformly random integer in $\mathbb{Z}_q$. As such, short vectors in $L^{\perp}$ multiplied with samples from a distribution behave noticeably differently depending on if the distribution is uniformly random or a module-LWE distribution.

4.3 NTWE Problem

We now present the natural lattices corresponding to the NTWE problem. These lattices correspond to the lattices used during primal and dual lattice attacks against the LWE problem. We therefore similarly denote our algorithms as the primal and dual attacks against the NTWE problem. Finding short vectors in these NTWE latices allows solving the NTWE problem.

4.3.1 Primal Attack

A primal attack against the NTWE problem use the same lattice construction as for the primal attack against the LWE problem. In a rank k NTWE instance with h samples, we are given

$$(\overline{\boldsymbol{A}}, \overline{\boldsymbol{b}} = (\overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{s}} + \overline{\boldsymbol{e}}) \cdot f^{-1}) \in R_q^{h \times k} \times R_q^k$$

and are either supposed to distinguish these samples from uniformly random or use the samples to recover $\overline{\boldsymbol{s}}$ and f. As with the primal attack against the LWE problem, this can be stated as finding a short vector in the lattice generated by the columns of

$$\begin{bmatrix} q\boldsymbol{I} & \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & t\boldsymbol{I} \end{bmatrix}$$

where $\boldsymbol{A} \in \mathbb{Z}^{hn \times kn}, \boldsymbol{B} \in \mathbb{Z}^{hn \times n}$ are the matrices corresponding to $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{b}}$ respectively, while t is some small constant. In the primal attack against the module-LWE problem, it is sufficient to consider only a single column of the matrix $\boldsymbol{B}$. However, when solving the NTWE problem, the full matrix $\boldsymbol{B}$ must be accounted for.

This NTWE lattice contains the $(h+k+1)n$ dimensional secret vector

$$\begin{bmatrix} \boldsymbol{e} \\ -\boldsymbol{s} \\ t\boldsymbol{f} \end{bmatrix} = \begin{bmatrix} q\boldsymbol{I} & \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & t\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \star \\ -\boldsymbol{s} \\ \boldsymbol{f} \end{bmatrix}$$

where $\star$ gives the modular reduction while $\boldsymbol{s}$ and $\boldsymbol{f}$ are the integer vectors representing $\overline{\boldsymbol{s}}$ and f.

Using $t = 1$, with an error distribution that has standard deviation σ, we expect the secret vector to have length approximately $\sigma \cdot \sqrt{(h+k+1)n}$. By the 2016 estimate, detailed in (3), we deem such a short lattice vector to be recoverable by BKZ with block size β if

$$\beta\sigma \leq \delta_\beta^{2\beta-(h'+k+1)n} q^{h'/(h'+k+1)}$$

for some $h' \leq h$ such that $h'n$ is an integer. Choosing $h' < h$ corresponds to ignoring rows of $\boldsymbol{A}$ and $\boldsymbol{B}$, which decreases both the lattice dimension and its determinant and sometimes leads to a more efficient attack.

Although the NTWE lattice at first glance may seem similar to the lattice given by a rank k module-LWE instance with the full $\boldsymbol{B}$ matrix, there are some significant differences. The known basis for these lattices are of exactly the same form and, when constructed using the same number of samples, the lattices have the same determinant. However, in the rank k module-LWE lattice, the target vectors are shorter than in the NTWE lattice. Furthermore, in the rank k module-LWE lattice, each of the secret vectors is known to lie in a specific $(k+h)n+1$ dimensional sublattice, which is not the case in the NTWE problem. Due to these factors, we believe that the hardness of the rank k NTWE problem is more comparable to that of the rank $k+1$ module-LWE problem.

The lattice constructed in the primal attack against the rank k NTWE problem is very similar to the lattice used in the primal attack against the rank $k+1$ module-LWE problem. Letting $t = 1$, and combining all but one column of $\boldsymbol{B}$ with $\boldsymbol{A}$ into $\tilde{\boldsymbol{A}} \in \mathbb{Z}^{hn \times ((k+1)n-1)}$, the lattice is given by

$$\begin{bmatrix} q\boldsymbol{I} & \tilde{\boldsymbol{A}} & \boldsymbol{b} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & 1 \end{bmatrix}$$

where $\boldsymbol{b}$ is the remaining column of $\boldsymbol{B}$. This lattice is one dimension smaller than the corresponding lattice for a rank $k+1$ module-LWE problem and the $\tilde{\boldsymbol{A}}$ matrix is not generated in the same way. However, with the same number of samples, the lattice determinant is the same as for the module-LWE lattice and the target vectors are of essentially the same length.

We can thus in some sense consider the NTWE problem as a more structured version of the rank $k+1$ module-LWE problem, with part of the $\tilde{\boldsymbol{A}}$ matrix not sampled uniformly at random. Furthermore, by standard estimates for hardness of lattice problems, the primal attack against the rank k NTWE problem and against rank $k+1$ module-LWE should require approximately as much work.

Note also that, unlike in the lattice given by a module-LWE instance, in the NTWE lattice there is not only a single short vector to be found. Instead the NTWE lattice contains several short vectors that span a dense n dimensional sublattice. The existence of such a dense sublattice seems to be the reason for attacks against an overstretched NTRU parameters [13], but the situation for NTWE is different from NTRU.

In a NTRU lattice, there is an n-dimensional dense sublattice in a $2n$-dimensional lattice. Meanwhile, in the NTWE lattice there is an n-dimensional dense sublattice in a $(hn + kn + n)$-dimensional lattice. In this NTWE lattice, the same analysis as for overstretched NTRU does not apply. It furthermore does not seem like there exist regime where it is significantly easier to find the dense sublattice in this NTWE lattice than it is to find the secret vector.

We also also note that we can construct the same lattice for the ordinary module-LWE problem, by including the full $\boldsymbol{B}$ matrix. As such, if there is some parameter regime where there is a behavior similar to overstretched NTRU, the same behavior also applies to LWE instances. This limits the potential impact of an overstretched parameter regime on the NTWE problem, unless there also exists overstretched parameters for the LWE problem.

Finally, we note that this lattice for the primal attack against the NTWE problem is very similar to the lattice used when attacking the NTRU problem. With $h = k + 1$, a combination of the $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{B}}$ matrices is a $(k + 1) \times (k + 1)$ dimensional matrix $\overline{\boldsymbol{H}}$. With $\boldsymbol{H}$ being the integer matrix representing $\overline{\boldsymbol{H}}$, a basis matrix for the primal attack is given by

$$\begin{bmatrix} q\boldsymbol{I} & \boldsymbol{H} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} .$$

This lattice is of exactly the same form as the lattice in a rank $k + 1$ module-NTRU instance, but with part of $\boldsymbol{H}$ uniformly random instead of given by $\boldsymbol{G}\boldsymbol{F}^{-1}$. As such, the natural primal lattice for the rank k NTWE problem is essentially a less structured version of the rank $k + 1$ module-NTRU lattice, as suggested by Lemma 3.3.

4.3.2 Dual Attack

A dual attack against the NTWE problem is performed in a similar way to how the dual attack is performed against the LWE problem. By using a short vector in the lattice

$$L^{\perp} = \{(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x}\boldsymbol{A} = \boldsymbol{y} \mod q\}$$

we are able to transform $\overline{\boldsymbol{b}}$ from an NTWE distribution into what essentially corresponds to an NTRU sample. This is the case as if $\boldsymbol{w}$ is a short vector in $L^{\perp}$ and $\overline{\boldsymbol{w}}$ is the corresponding vector in R^k then

$$\overline{\boldsymbol{w}} \cdot \overline{\boldsymbol{b}} = \overline{\boldsymbol{w}} \cdot (\overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{s}} + \overline{\boldsymbol{e}}) \cdot f^{-1} = (\overline{\boldsymbol{y}} \cdot \overline{\boldsymbol{s}} + \overline{\boldsymbol{x}} \cdot \overline{\boldsymbol{e}}) \cdot f^{-1} = gf^{-1}$$

where $g = \overline{\boldsymbol{y}} \cdot \overline{\boldsymbol{s}} + \overline{\boldsymbol{x}} \cdot \overline{\boldsymbol{e}}$ is a short element in R_q if $\overline{\boldsymbol{w}}$ is short. As such, $\overline{\boldsymbol{w}} \cdot \overline{\boldsymbol{b}} = gf^{-1}$ can be interpreted as an NTRU sample, although with different distributions for g and f.

Note that, as we consider the norm of the coefficient vector, the norm of g is dependent on the ring R. However, as we only consider the case where R is the ring of integers of a power of two cyclotomic field, we are guaranteed that the product $\overline{\boldsymbol{s}} \cdot \overline{\boldsymbol{y}}$ is a small element in R_q if both $\overline{\boldsymbol{y}}$ and $\overline{\boldsymbol{s}}$ are short.

In order for it to be reasonable to solve the constructed NTRU instance, we require that the vector given by (g, f) is significantly shorter than the shortest vector in a random q-ary $2n$-dimensional lattice without the NTRU structure. Such a lattice has determinant q^n and, by the Gaussian Heurisitic, is expected to contain a vector of length $\sqrt{(qn)/(\pi e)}$.

On the other hand, the lattice corresponding to the constructed NTRU instance contains a short vector (g, f) where $g = \overline{\boldsymbol{y}} \cdot \overline{\boldsymbol{s}} + \overline{\boldsymbol{x}} \cdot \overline{\boldsymbol{e}}$ is expected to have length $\|(x, y)\| \cdot \sigma\sqrt{(h + k)n}$. Furthermore, we have that f is of expected length $\sigma\sqrt{n}$, and we can thus argue that in order for the constructed NTRU instance to actually contain an unusually short vector, we have the requirement that

$$\sigma^2 n \left((h + k) \|(x, y)\|^2 + 1\right) \leq \frac{nq}{\pi e}$$

or equivalently

$$\|(x, y)\| \leq \sqrt{\frac{q}{(h + k)\pi e\sigma^2} - \frac{1}{h + k}} \ .$$

We can improve this attack by rebalancing the NTRU lattice so that we have to find a vector of length $L = \|(x, y)\| \cdot \sigma\sqrt{2(h + k)n}$, corresponding to $(g, f \cdot \sqrt{h + k}\, \|(x, y)\|)$. Although this vector is longer, the corresponding NTRU lattice also has a determinant that is $(\sqrt{h + k}\, \|(x, y)\|)^n$ times larger, meaning that it is not expected to contain as short vectors. This leads to the requirement that

$$2(h + k)n\sigma^2 \|(x, y)\|^2 \leq \frac{nq\sqrt{h + k}\, \|(x, y)\|}{\pi e} \tag{4}$$

in order for the dual attack to succeed, which corresponds to a requirement that

$$\sigma^2 \geq \frac{q}{2\pi e \|(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}})\| \sqrt{h + k}}$$

for the problem to not be solvable by a given $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}})$ in $L^\perp$.

The lattice $L^\perp$ is $h + k$ dimensional and has determinant q^k. As such, BKZ with block-size β should be able to find a vector $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}})$ of length $\delta_\beta^{(h+k)n} q^{k/(h+k)}$ in $L^\perp$. This means that, for the problem to be hard to solve, we require that

$$\sigma^2 \geq \frac{q^{h/(h+k)}}{2\pi e \cdot \delta_\beta^{(h+k)n} \cdot \sqrt{h + k}} \tag{5}$$

for every β that an adversary can afford to use as block-size.

While (4) is a necessary requirement for the constructed NTRU instance to be solvable via lattice reduction, a bounded adversary may still not be able to solve this NTRU instance. Therefore, we also estimate how short vectors can be found in the resulting $2n$-dimensional NTRU lattice.

In the resulting NTRU instance, there is an unusually short vector, with expected length $L = \sigma\sqrt{2(h + k)n}\delta_\beta^{(h+k)n} q^{k/(h+k)}$. By the 2016 estimate, detailed in (3), this vector is expected to be found by BKZ with block-size β in the $2n$-dimensional NTRU lattice with determinant $(qL)^n$ if

$$L \cdot \sqrt{\beta/(2n)} \leq \delta_\beta^{2\beta - 2n} \sqrt{qL} \ .$$

This gives that an adversary succeeds if

$$L\beta = \sigma\sqrt{2(h + k)n} \cdot \delta_\beta^{(h+k)n} \cdot q^{k/(h+k)} \cdot \beta \leq 2nq\delta_\beta^{4\beta - 4n}$$

which corresponds to the requirement that

$$\sigma \geq \delta_\beta^{4\beta-(4+h+k)n} \frac{q^{k/(h+k)}\sqrt{2n}}{\beta \cdot \sqrt{h+k}} \tag{6}$$

in order for the problem to not be solvable by an adversary that is able to run BKZ with block size β. However, this analysis is only applicable if the secret vector actually is unusually short in the NTRU lattice, and thus, for hardness of the NTWE problem, it is sufficient that either (5) or (6) is fulfilled.

It is not directly clear how this dual attack compares to the primal attack against the NTWE problem or to attacks against the NTRU and LWE problems. For the concrete parametrizations we present in Section 6, it seems like this approach for a dual-lattice attack against the NTWE problem is significantly less efficient than the primal attack. However, for certain choices of parameters, it seems like this dual attack is more efficient than the primal attack. In particular, by our estimates, the NTWE problem parametrized with $h = k = 1$ is often more efficiently solved with this dual attack than with the primal attack.

5 Our Cryptosystem

The procedures for key generation, encryption and decryption in our cryptosystem are detailed in Algorithm 1. The system follows essentially the same idea as the LWE-based Lindner-Peikert scheme [14] but with decryption requiring using f in a similar way to how it is used in an NTRU based cryptosystem.

For our cryptosystem, we only consider the case where R is the ring of integers of a power of two cyclotomic field, but the system parametrized with other rings R could potentially also be interesting to investigate. We also only consider the system using $p = 2$, encoding a single bit of the message into the relevant coefficients of the ciphertext.

To improve the efficiency of our scheme, we prefer rings R and modulos q such that operations can be performed efficiently using the Number Theoretic Transform (NTT). With such a parametrization, it is very efficient to compute the inverse f^{-1} in R_q. However, computing the inverse in R_p is a less efficient operation. If the public key is reused multiple times, this additional time for key generation can be acceptable.

During decryption, a product with f_p must be computed, which can not be done efficiently by using the NTT. As such, this multiplication has a significant impact on the decryption efficiency in our PKE. For our PKE scheme, a large majority of the decryption time is spent on this multiplication. However, if some version of the Fujisaki-Okamoto transform is used to construct an IND-CCA secure KEM from this IND-CPA secure PKE, decryption in the resulting scheme also performs encryption of a message. Therefore, in this KEM, although the multiplication by f_p^{-1} still has a significant performance impact, it no longer constitutes a large majority of the decryption time.

We consider two different versions of ψ_{f} in our parametrizations, both defined in terms of ψ_{gen}. In both versions, samples from this distribution, are always invertible in both R_p and R_q. In the first version of this distribution, we let a sample from ψ_{f} be sampled from ψ_{gen} with rejection sampling ensuring that the result is invertible in both R_p and R_q. In the second version, we let a sample be given by $f = pf' + 1$, with $f' \leftarrow \psi_{\mathrm{gen}}$

ensuring that f is the identity in R_p. By using rejection sampling, it is also ensured that samples from this version of ψ_{f} are invertible in R_q.

Selecting $f = pf' + 1$ ensures that $f_p^{-1} = 1$ and therefore no expensive inverse has to be computed during key generation. Furthermore, this choice of f ensures that multiplication with f_p^{-1} is trivial, resulting in more efficient decryption. However, this comes at the cost of using a larger f, which results in a larger failure probability for the scheme. As the structure of how f is sampled is known, we can not argue that this larger f results in harder instances of the corresponding lattice problem.

For implementations, the matrix $\overline{\boldsymbol{A}}$ may be sampled from a pseudorandom number generator. This allows a much more compact public key, as it only has to include the seed used to derive $\overline{\boldsymbol{A}}$ instead of the full $\overline{\boldsymbol{A}}$ matrix. Using a short seed to represent the public matrix $\overline{\boldsymbol{A}}$ in this way is standard for LWE-based cryptosystems and is for example used in Kyber [22].

Algorithm 1 Procedures for key generation, encryption and decryption for our cryptosystem

```
procedure Key generation
    $\overline{\boldsymbol{A}} \leftarrow \mathcal{U}(R_q)^{h \times k}$
    $\overline{\boldsymbol{s}} \leftarrow \psi_{\mathrm{gen}}^k$
    $\overline{\boldsymbol{e}} \leftarrow \psi_{\mathrm{gen}}^h$
    $f \leftarrow \psi_{\mathrm{f}}$
    $b = \left(\overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{s}} + \overline{\boldsymbol{e}}\right) \cdot f^{-1} \in R_q^h$
    Let $f_p^{-1}$ be the inverse of $f$ in $R_p$
    return $(pk = (\overline{\boldsymbol{A}}, \overline{\boldsymbol{b}}), sk = (\overline{\boldsymbol{s}}, f, f_p^{-1}))$
end procedure
procedure Encryption($(\overline{\boldsymbol{A}}, \overline{\boldsymbol{b}}) = pk$, $m \in \mathbb{R}_p$)
    $\overline{\boldsymbol{s}}' \leftarrow \psi_{\mathrm{enc}}^h$
    $e' \leftarrow \psi_{\mathrm{enc}}$
    $\overline{\boldsymbol{e}}'' \leftarrow \psi_{\mathrm{enc}}^k$
    $c_1 = \overline{\boldsymbol{s}}' \cdot \overline{\boldsymbol{b}} + e' + \lfloor mq/p \rfloor$        ▷ With $m$ interpreted as element in $R_q$
    $\overline{\boldsymbol{c}}_2 = \overline{\boldsymbol{s}}' \cdot \overline{\boldsymbol{A}} + \overline{\boldsymbol{e}}''$
    return $ct = (c_1, \overline{\boldsymbol{c}}_2) \in R_q \times R_q^k$
end procedure
procedure Decryption($(\overline{\boldsymbol{s}}, f, f_p^{-1}) = sk$, $(c_1, \overline{\boldsymbol{c}}_2) = ct$)
    Let $v = c_1 \cdot f - \overline{\boldsymbol{c}}_2 \cdot \overline{\boldsymbol{s}} \mod q$
    Interpret $v$ as element in $R$ with coefficients in $[0, q)$
    Let $u = \lfloor v \cdot p/q \rceil$ interpreted as element in $R_p$
    return $vf_p^{-1} \in R_p$
end procedure
```

5.1 Security

The security of our cryptosystem relies on the hardness of both the NTWE problem and the module-LWE problem. Based on the assumed hardness of the decision NTWE problem, the public key in our cryptosystem is indistinguishable from uniformly random. Meanwhile, assuming the hardness of the decision module-LWE problem, the ciphertext completely masks the encrypted message. This is similar to a typical NTRU-based cryptosystem, where the public key is pseudorandomly generated as an NTRU instance, while the security of the ciphertexts relies on the hardness of a problem that can be seen as a variant of the ring-LWE problem.

In the following lemmas, we formalize how, assuming the computational hardness of the NTWE and module-LWE problems, the security of our cryptosystem is guaranteed. First we note that, assuming the hardness of the decision NTWE problem, the public key of our cryptosystem is indistinguishable from uniformly random.

Lemma 5.1. *Let W be an algorithm that, with advantage ε, can distinguish the public keys from our cryptosystem from uniformly random. Then, using W once with a negligible amount of additional computations provides a solution to the $DNTWE(\psi_{gen}, h)$ problem with advantage ε.*

Proof. The public key in our cryptosystem consists of h NTWE samples. Thus, given h samples from an instance of the DNTWE(ψ_{gen}, h) problem, we can consider these as a public key for our system. If the samples are from an NTWE distribution, the public key is exactly distributed as for our actual cryptosystem. Thus, using W with these samples as the public key gives an algorithm that, with advantage ε, solves the DNTWE(ψ_{gen}, h) problem. □

The next lemma shows that, assuming the hardness of the rank h module-LWE problem, a version of our cryptosystem where a uniformly random public key is used is IND-CPA secure.

Lemma 5.2. *Assume that there is an adversary $\mathcal{A}$ that is able to achieve an advantage ε against the IND-CPA security of a version of our cryptosystem that uses uniformly random public keys. Then, using $\mathcal{A}$ once, with a negligible amount of additional computations, provides a solution to the rank h decision module-LWE problem with advantage ε.*

Proof. In a version of our system where the public key is uniformly random, the ciphertext is directly given by $k + 1$ samples from a rank h module-LWE distribution. The public key in this case is given by the $\overline{\boldsymbol{a}}$ part of these module-LWE samples, while the ciphertext is constructed from the b part of the samples.

To encrypt the message encoded as $m \in R_p$, the ciphertext is constructed with $\overline{\boldsymbol{c}}_2$ being the b part of k module-LWE samples. The corresponding $\overline{\boldsymbol{a}}$ part of the module-LWE samples are used as the $\overline{\boldsymbol{A}}$ matrix for the public key. The $\overline{\boldsymbol{a}}$ part of the final module-LWE sample gives the $\overline{\boldsymbol{b}}$ part of the public key for our cryptosystem with uniformly random public key. Meanwhile, the c_1 part of the ciphertext is given by the b part of this final module-LWE sample plus $\lfloor mq/p \rfloor$. This exactly corresponds to the ciphertext that encrypts m in a version of our cryptosystem that uses a uniformly random public key.

If instead given $k + 1$ samples from a uniformly random distribution, the ciphertext constructed in this way is uniformly random. As such, using $\mathcal{A}$ against these ciphertexts gives an advantage ε in distinguishing between the uniform distribution and a module-LWE distribution. □

Finally, combining these lemmas shows that, assuming the hardness of both the NTWE problem and the rank h module-LWE problem, our cryptosystem is IND-CPA secure.

Lemma 5.3. *Assume that there is an adversary $\mathcal{A}$ that achieves an advantage 2ε against the IND-CPA security of our cryptosystem. Then, using $\mathcal{A}$ once, with a negligible amount of additional computations, provides a solution, with advantage ε, to either a rank h module-LWE problem or the rank k DNTWE(ψ_{gen}, h) problem.*

Proof. An adversary $\mathcal{A}$ achieving advantage 2ε against the IND-CPA security of our cryptosystem could be used in order to solve either the relevant decision NTWE problem or the relevant decision ring-LWE problem with advantage ε. This follows from a simple hybrid argument and using Lemmas 5.1 and 5.2.

If $\mathcal{A}$ has advantage at least ε against a version of our cryptosystem with uniformly random public key, Lemma 5.2 provides an efficient algorithm for the decision module-LWE problem.

Otherwise, $\mathcal{A}$ has advantage 2ε against our cryptosystem but advantage less than ε against a version of our cryptosystem where the public key is uniformly random. This provides an ε distinguisher between our cryptosystem and a version of the system with uniformly random public key. Thus, by Lemma 5.1, we can use $\mathcal{A}$ to solve the DNTWE(ψ_{gen}, h) problem with advantage ε. □

Our scheme only claims to be IND-CPA secure and, as with LWE- and NTRU-based schemes, it is vulnerable to a trivial chosen-ciphertext attack where the decryption oracle is used with the target ciphertext plus some small noise. Our PKE can, however, be used to construct an IND-CCA secure KEM by using some variant of the Fujisaki-Okamoto transform [10]. This approach to achieving IND-CCA security is also used in many of the submissions to the NIST post-quantum standardization process.

5.2 Correctness of Decryption

In the decryption algorithm, the value of v is given by

$$\begin{aligned} v &= c_1 \cdot f - \overline{\boldsymbol{c}}_2 \cdot \overline{\boldsymbol{s}} \\ &= ((\overline{\boldsymbol{s}}' \cdot (\overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{s}} + \overline{\boldsymbol{e}}) f^{-1} + e') + \lfloor mq/p \rfloor) \cdot f - (\overline{\boldsymbol{s}}' \cdot \overline{\boldsymbol{A}} + \overline{\boldsymbol{e}}'') \cdot \overline{\boldsymbol{s}} \\ &= \overline{\boldsymbol{s}}' \cdot \overline{\boldsymbol{e}} - \overline{\boldsymbol{e}}'' \cdot \overline{\boldsymbol{s}} + (\lfloor mq/p \rfloor + e') \cdot f \end{aligned}$$

and we want $\lfloor vp/q \rceil$ to equal mf, when both are interpreted as elements in R_p. This is the case if every coefficient of v is less than a distance $q/(2p)$ from the corresponding coefficient in $(mq/p) \cdot f$ and we therefore want to bound this distance.

To this end, we first note that the most $\lfloor mq/p \rfloor \cdot f$ can differ from $(mq/p) \cdot f$ is if maximal rounding occurs for all coefficients. In this case, the difference is rf, where r is $1/2$ in all coefficients. Thus, the decryption is correct if each coefficient of

$$(r + e')f - \overline{\boldsymbol{e}}'' \cdot \overline{\boldsymbol{s}} + \overline{\boldsymbol{s}}' \cdot \overline{\boldsymbol{e}}$$

is smaller than $q/(2p)$.

As we consider power of two cyclotomic fields, the distribution of every coefficient of the resulting product is the same. We can therefore consider the corresponding integer vectors and bound the probability that

$$\frac{q}{2p} < \left| \boldsymbol{s}' \cdot \boldsymbol{e} + (\boldsymbol{r} + \boldsymbol{e}') \cdot \boldsymbol{f} - \boldsymbol{e}'' \cdot \boldsymbol{s} \right| \ .$$

To get a rough idea of the decryption failure probability of our system, we consider the case where ψ_{gen} and ψ_{enc} are discrete Gaussian distributions with standard deviations σ_{gen} and σ_{enc} respectively and where $\psi_{\mathrm{f}} = \psi_{\mathrm{gen}}$. Furthermore, we consider the case where we encrypt the all 0 message, meaning that $mq/p = \lfloor mq/p \rceil$ and we therefore do not have to consider contribution of r. This allows the following minor alteration of Lemma 3.1 from [14] to be used to bound the decryption failure probability.

Lemma 5.4. *The error probability per symbol (over the choice of secret key) when decrypting the all* 0 *message, is bounded from above by any desired* δ *as long as*

$$\sigma_{gen} \cdot \sigma_{enc} \leq \frac{1}{4p} \frac{q}{\sqrt{2(h+k+1) \cdot n \cdot \ln(2/\delta)}}$$

except for with a probability less than 2^{-n} *over the randomness in the ciphertext.*

For a more precise bound on the failure probability applicable for the different types of error distributions used in our concrete parametrizations, we numerically calculate the failure probability in the same way as done for the Kyber submission [22], by estimating the actual probability distribution for the error terms.

5.3 Comparison to a Other Cryptosystems

5.3.1 Security

Based on the analysis in Section 4, the NTWE problem in rank k seems to be as hard as the rank $k+1$ module-LWE problem. As such, our cryptosystem parametrized with $h = k+1$ should have security comparable to a rank $k+1$ module-LWE-based system that uses the same error distributions. For general h, our system corresponds to a module-LWE-based system where the public key is given by h samples from a rank $k+1$ module-LWE distribution while the ciphertext corresponds to $k+1$ samples from a rank h module-LWE distribution.

In our system, the public key is a sample from an NTWE distribution. Similarly, the public key in an NTRU-based system is given by an instance of the NTRU problem. As such, the security against key-recovery attacks in these systems is based on the hardness of the search-NTWE problem and the search-NTRU problems respectively. Furthermore, the NTWE problem should be essentially as hard as the rank $k+1$ module-NTRU problem, as indicated by Lemma 3.3 and analyzed in Section 4.

Compared to a rank $k+1$ module-LWE-based system, our system exposes fewer module-LWE samples as part of the ciphertext. Whereas the ciphertext in our system consists of $k+1$ module-LWE samples, the ciphertext in the module-LWE-based system consists of $k+2$ such samples. This could potentially be a reason for our system to be more secure than the corresponding module-LWE-based system. However, based on current understanding, the number of available module-LWE samples does not significantly impact the hardness of this problem.

The ciphertext for our system with $h = k+1$ is also very similar to the ciphertext in a rank $k+1$ module-NTRU-based cryptosystem. In the module-NTRU-based cryptosystem, the ciphertext consists of $k+1$ noisy inner products of public data with a secret vector. Similarly our system exposes $k+1$ such noisy inner products.

Module-NTRU based cryptosystem are typically not considered due to their large public keys. Therefore, a more fair comparison in the efficiency and compactness of our system is to that of an NTRU-based system that uses a ring of degree $(k+1)n$. The natural lattice problem for this NTRU problem is essentially the same as for the rank $k+1$ module-NTRU problem and based on current understanding, the security of these systems should be essentially the same.

5.3.2 Efficiency

Besides calculation of inverses and an additional multiplication by f^{-1}, our cryptosystem performs the same operations as in a typical LWE-based cryptosystem and should thus have similar efficiency. When using $f = pf' + 1$, all multiplications are efficiently computable in the NTT domain. Furthermore, calculating the inverse f^{-1} in R_q is also efficient in the NTT domain, meaning that these additional steps barely affect the performance of the scheme. Furthermore, as we deem the rank k NTWE problem to be as hard as a rank $k+1$ module-LWE problem, for the same security level our system should actually be more efficient than a comparable module-LWE-based system.

Compared to a rank $k+1$ module-LWE-based system, our cryptosystem does not need as much uniformly random data for $\overline{\boldsymbol{A}}$. Furthermore, in our system, we perform fewer additions and multiplications in key generation and encryption, and essentially the same number of these operations during decryption.

With $\overline{\boldsymbol{s}}$ and f combined, the key generation samples as much data from error distributions as in the key generation of a rank $k+1$ module-LWE-based system with h samples in the public key. With $f = pf' + 1$, we are guaranteed that f_p^{-1} is trivial. However, we are not guaranteed that f is invertible in R_q and may therefore have to sample multiple f' from ψ_{gen}. However, for the rings we consider, the probability that f is not invertible is small enough that this resampling has a negligible impact on the average running time of key generation.

Another reason our scheme can be more efficiently implemented than a module-LWE-based scheme is that we do not perform any ciphertext compression. This allows the ciphertexts in our system to be transmitted in NTT-form, decreasing the number of times we have to perform the transform and its inverse. This results in a speed-up both during the encryption of messages and the decryption of ciphertexts.

Compared to an NTRU-based scheme, it is easier to parametrize our scheme with rings that support efficient operations by using the NTT. This is the case as we may select a fixed base ring which supports efficient NTT operations and target different security levels by altering h and k. As module-NTRU-based systems are typically not considered due to their large public keys, having an NTRU-based system support efficient NTT operations imposes a restrictive condition on the possible rings that can be used. It is possible to construct NTRU-based systems that support NTT operations, as done in a paper by Lyubashevsky and Seiler [16]. However, the NTRU-based submissions for public key-encryption and key establishment in the third round of the NIST post-quantum standardization process did not support efficient NTT operations [3, 5].

Compared to an NTRU-based scheme that does not use NTT, our key-generation should be significantly more efficient, at least if using an f that given by $pf' + 1$. In this case, we have no expensive operations during key-generation, as f_p^{-1} is trivial while the inverse in R_q is efficiently computed by using the NTT. Meanwhile, in an NTRU-based scheme that does not support the NTT, at least one expensive inverse must be computed. However, when our system samples f directly from ψ_{gen}, the inverse in f_p^{-1} must be computed. Therefore, this version of our scheme has a key-generation time that is more similar to the one of an NTRU-based system.

It is harder to compare the performance of encryption and decryption for our system to that of an NTRU-based system. By using the NTT, general multiplications in our system are computed more efficiently than possible without using the NTT. However, NTRU-based systems typically only consider multiplications with certain classes of polynomials, which allows multiplications that have similar efficiency to that of our scheme. Furthermore, if our scheme is used with a seed for $\overline{\boldsymbol{A}}$ instead of the full matrix in the public key, the full matrix must be generated both during encryption and decryption. This can be a somewhat costly operation that is not necessary in the NTRU-based system, where the public key is a single ring element.

5.3.3 Compactness

In a typical rank $k+1$ module-LWE-based system, the ciphertext consists of a heavily compressed ring element and a somewhat compressed module element. In total, these are represented by using essentially as much space as it takes to represent a single rank $k+1$ module element. The public-key in such a system is $k+1$ ring elements and a uniformly random matrix that is typically represented by a small seed.

In our cryptosystem, the ciphertext consists of an uncompressed rank k module element and an uncompressed ring element. This is represented in the same amount of space as a single rank $k+1$ module element, and our ciphertext size is therefore essentially the same as for a comparable module-LWE-based system. In our system, the public key consists of h ring elements and a uniformly random matrix. Thus, by using $h = k+1$ and representing the uniformly random matrix by a small seed, our public key is of the same size as in the rank $k+1$ module-LWE-based system.

In an NTRU-based system, both the public-key and the ciphertext consists of a single ring element. Representing an element of a degree $(k+1)n$ ring takes as much space as representing $k+1$ elements from a degree n ring. Thus, our cryptosystem has the same sized ciphertext as an NTRU-based system in a ring of degree $(k+1)n$. Furthermore, our system with $h = k+1$ and with $\overline{\boldsymbol{A}}$ represented by a small seed has essentially the same sized public-key as this NTRU-based system, with the only difference being this small seed that is used to derive the $\overline{\boldsymbol{A}}$ matrix.

6 Example Parametrizations

For our cryptosystem, described in Algorithm 1, we are able to choose ring R, integers p, q and error distributions $\psi_{\text{gen}}, \psi_{\text{enc}}, \psi_{\text{f}}$ relatively freely. To use Lemma 5.3 to argue for the security of our system, we require both that the relevant NTWE instance is hard and that a rank h module-LWE problem is hard.

One way to parametrize our system is to use $h = k$ and selecting ψ_{gen} and ψ_{enc} differently. By balancing the standard deviation on ψ_{gen} and ψ_{enc}, we can ensure that both problems seem to be equally hard to solve, and that the scheme achieves an acceptable decryption failure probability.

This results in parametrizations that are similar to a rank $k+1$ module-LWE-based scheme but where only k module-LWE samples are included in the public key.

By instead using $h = k+1$ and $\psi_{\text{gen}} = \psi_{\text{enc}}$, the resulting parametrizations are more comparable to typical module-LWE-based schemes. We propose such parametrizations that support NTT calculations, using the same ring R and modulos q as in Kyber. This results in parametrizations with performance quite similar to Kyber, but which should allow for more efficient implementations and which do not have to use any ciphertext compression.

Another approach is to instantiate the system to only rely on the concrete hardness of the NTRU and ring-LWE problems, as these problems have already been thoroughly investigated. While there are relations between the NTRU and ring-LWE problems [18, 23, 24], these do not directly show that specific instantiations of a cryptosystem based on the NTRU problem or on some version of the LWE problem is more secure than another. However, our cryptosystem parametrized like this is essentially guaranteed to remain secure as long as either the corresponding NTRU- or module-LWE-based cryptosystem is secure.

For the concrete parametrization we propose, we let the $\overline{\boldsymbol{A}}$ part of the public key be derived from a 256 bit seed using some cryptographically secure pseudorandom number generator. This significantly decreases the size of the public key as this 32 byte seed is sufficient to represent the full $\overline{\boldsymbol{A}}$ matrix in the public key.

For all our parametrizations we use $p = 2$, encoding a single bit in each of the coefficients of the element c_1 of the ciphertext. Using a larger p allows the same sized ciphertext to include more data, but comes at the cost of a larger decryption failure probability.

We consider two different versions for the distribution ψ_{f}, corresponding either directly to a sample from ψ_{gen} or from $p\psi_{\text{gen}} + 1$, as described in Section 5. The first version results in a less efficient key generation, as inverses in R_p must be calculated, but with a smaller decryption failure probability δ. The second version ensures that key generation is efficient, but results in a larger decryption failure probability δ_p. In the parametrizations in Tables 1 and 2, we present both the decryption failure probabilities δ and δ_p for these different choices of ψ_{f}.

The decryption failure probabilities δ, δ_p and the core SVP security of the presented parametrizations have been calculated using a modified version of the script used to calculate the corresponding parameters for the Kyber specification.

6.1 Skewed Parameters

Here we consider parametrizations of the cryptosystem that use $h = k$, resulting in a public key that is significantly smaller than the ciphertext. This results in a system where the security of the public-key is based on the NTWE problem in rank k while the message security is based on the hardness of the rank k module-LWE problem. With the same error distribution, the rank k NTWE problem seems to be significantly harder than the rank k module-LWE problem. For these parametrizations, we therefore use error distributions with the standard deviation for ψ_{gen} significantly smaller than the standard deviation for ψ_{enc}.

We use the same NTT friendly ring $R = \mathbb{Z}[X]/(X^{256}+1)$ with $q = 3329$ as in Kyber. The error distributions are discrete Gaussian distributions

Version	512	1024
Core SVP PK	144	280
Core SVP CT	140	276
$h = k$	2	4
q	3329	3329
σ_{gen}	0.49	0.42
σ_{enc}	9.62	8.04
PK size (bytes)	800	1568
CT size (bytes)	1152	1920
δ	$< 2^{-300}$	$< 2^{-300}$
δ_p	2^{-201}	2^{-272}

Table 1: Some different parametrizations of our scheme with $h = k$. The table details the size of the public-key (PK) and ciphertext (CT). It also details the estimated Core SVP hardness, as described in Section 2.5, of the lattice problems underlying the public-key and ciphertext respectively.

with standard deviations σ_{gen} and σ_{enc} for key generation and encryption respectively. As we use $p = 2$ and a ring with degree 256, each ciphertext encrypts a 256-bit message. These parametrizations are detailed in Table 1.

A module-LWE based cryptosystem can also be parametrized with comparable parameters. This is achieved by letting the public key consist of k samples from a rank $k+1$ module-LWE instance, while the ciphertext is given by samples from a rank k module-LWE instance. These skewed parametrizations of a module-LWE-based cryptosystem are, however, not typically considered. We also believe that other parametrizations of our NTWE-based cryptosystem are more interesting than these that use $h = k$.

6.2 Parameters Similar to Kyber

In Table 2 we present parametrizations of our scheme that have been selected to be similar to parametrizations of Kyber [22]. As the NTWE problem is used for key generation in our scheme, our parametrizations use a k that is one rank smaller than the module rank used in corresponding Kyber parametrizations, while still claiming that the problem is essentially as hard. This allows our parametrizations to have essentially the same public key and ciphertext size as the corresponding Kyber implementations, even though our scheme does not include any ciphertext compression.

In comparison to Kyber, encryption for these parametrizations is more efficient, as we use a smaller module-rank for an equivalent security level. Furthermore, if ψ_{f} is given by $2\psi_{\text{gen}} + 1$, key generation in our scheme is also more efficient than in Kyber, as we use a smaller module-rank and the inverse f^{-1} in R_q is efficiently computable via the NTT.

We do not have an optimized implementation of our scheme and we have not performed any extensive profiling in order to compare the performance of Kyber and our scheme. However, we have implemented our scheme by modifying an implementation of Kyber. With ψ_{f} given by $2\psi_{\text{gen}} + 1$, the combination of key generation, encryption and decryption

Version	512-3	512-4	768-2	768-3	1024
Core SVP PK	118	123	182	193	256
Core SVP CT	118	124	183	191	253
k	1	1	2	2	3
m	2	2	3	3	4
$\psi = \psi_{\text{enc}} = \psi_{\text{gen}}$	$\mathcal{B}_3$	$\mathcal{B}_4$	$\mathcal{B}_2$	$\mathcal{B}_3$	$\mathcal{B}_2$
PK size (bytes)	800	800	1184	1184	1568
CT size (bytes)	768	768	1152	1152	1536
δ	2^{-190}	2^{-108}	2^{-291}	2^{-131}	2^{-224}
δ_p	2^{-102}	2^{-58}	2^{-182}	2^{-82}	2^{-153}

Table 2: Parametrizations of our scheme comparable to Kyber. The table details the size of the public-key (PK) and ciphertext (CT). It also details the estimated Core SVP hardness, as described in Section 2.5, of the lattice problems underlying the public-key and ciphertext respectively.

runs in around 10% less time than for the original Kyber implementation.

The decryption failure probability of our schemes with ψ_{f} directly given by ψ_{gen} is somewhat smaller than for the corresponding Kyber parametrizations. For our schemes, we recover a noisy version of the encoded message, with noise corresponding to the sum of $k + h + 1 = 2(k + 1)$ products of two small polynomials. In the corresponding parametrization of Kyber, the noise is the sum of $2(k+1)$ products of two small polynomials plus another small polynomial.

The contribution to the decryption failure probability of a single small polynomial is typically small. However, the ciphertext compression performed in Kyber increases the size of this small error polynomial, causing it to have a significant impact on the decryption failure probability. This means that, for a comparable decryption error probability, our scheme can be parametrized with a larger standard deviation for the error distributions than in Kyber. This allows us to parametrize our scheme to target somewhat higher security levels than in Kyber, at least when using ψ_{f} directly given by ψ_{gen}.

The error distribution used in these parametrizations is a centered binomial distribution $\mathcal{B}_k$, as in Kyber. A sample from this distribution is given by $\sum_{i=1}^{k}(x_i - y_i)$, where x_i and y_i are sampled from a Bernoulli distributed with equal probability for 0 and 1. As we are able to achieve a smaller decryption failure probability than in Kyber, we also include additional parametrizations that use larger error distributions than the ones used in Kyber.

All of these parametrizations use the same ring as in Kyber, namely $R = \mathbb{Z}[X]/(X^{256} + 1)$ and with the same modulos $q = 3329$. This allows efficient NTT operations in R_q. Furthermore, as we use $p = 2$, each ciphertext encrypts a 256-bit message.

6.3 Parameters Combining NTRU and LWE

A conservative approach for parametrizing our cryptosystem is to use $k = h = 1$ and only rely on the hardness given by Lemmas 3.1 and 3.2. Although we believe this to be overly conservative, the resulting cryptosystem serves as an efficient hybrid between cryptosystems based on the

NTRU and ring-LWE problems. The security of this scheme against attacks that recover the secret key of the system is guaranteed if either the corresponding NTRU-based or ring-LWE-based cryptosystem is secure against such attacks.

Our resulting system has the same public key size as the corresponding ring-LWE-based system. As we use a small seed to represent the $\overline{\boldsymbol{A}}$ matrix, the public key is also only 32-bytes larger than for the corresponding NTRU-based systems. Meanwhile, the ciphertexts in this system are as large as in the corresponding ring-LWE-based system without ciphertext compression.

As an example, we can choose a parametrization similar to one of the parametrizations of the New Hope [1] system. Thus, we use the ring $\mathbb{Z}[X]/(X^{1024}+1)$ and $q = 12289$ which allow efficient computations via the NTT. Using the same error distribution, we can argue that against key recovery attacks, our system is at least as secure as New Hope, while also being at least as secure as a corresponding NTRU-based cryptosystem. However, as we do not perform any ciphertext compression, this comes at the cost of a significantly larger ciphertext than the one for New Hope. Furthermore, if we sample f so that f_p^{-1} is trivial, the resulting scheme has significantly larger failure probability than the New Hope scheme. If we instead sample f directly from ψ_{gen}, the resulting scheme has significantly less efficient key generation than the New Hope scheme. As such, this scheme does not really compare favorably to New Hope by itself.

Compared to a system where NTRU and a ring-LWE-based system are used in parallel, our scheme does however have several advantages. Our scheme is more efficient than a hybrid of ring-LWE and NTRU and also has significantly smaller public key than that of a combined NTRU-based and ring-LWE-based scheme. The ciphertext is the same size as in a ring-LWE-based system without ciphertext compression, which is smaller than the size of the combined ciphertexts of an NTRU-based system and the New Hope cryptosystem.

While the security against key-recovery attacks is guaranteed if either of the corresponding NTRU or ring-LWE-based cryptosystems is secure, we do not have the same guarantee for message security. The ciphertext is computed in essentially the same way as in the New Hope cryptosystem, and we can more or less guarantee the IND-CPA security of our system if the New Hope system is secure.

The ciphertext of our system is also very similar to that of a comparable NTRU-based system. However, in the NTRU-based system, the ciphertext is a single noisy inner product while our ciphertext consists of two such products. Based on current understanding, the number of such noisy products, corresponding to ring-LWE samples, should not significantly impact how hard the products are to distinguish from uniformly random. As such, based on our current understanding, our system parametrized in this way is IND-CPA secure if New Hope is IND-CPA secure or if the corresponding NTRU-based system is IND-CPA secure.

7 Final Remarks

Based on our concrete hardness estimates for the NTWE problem, we parametrize our NTWE-based cryptosystem to have performance that is competitive to that of highly efficient module-LWE-based schemes. While the concrete hardness of the NTWE problem has not been analyzed before, we argue that its similarity to the NTRU and LWE problems provides some confidence in the security of these parametrizations.

As with the NTRU and LWE problems, the NTWE problem also naturally corresponds to a problem in a q-ary lattice. This NTWE lattice can be seen as a mix between an LWE lattice and a NTRU lattice, which motivates our belief in the hardness of the NTWE problem. We believe that any improved algorithms against the NTWE problem are likely to have interesting consequences for the NTRU and LWE problems as well. One possibility is that any such algorithm is directly applicable to the NTRU and LWE problems, which is of obvious of interest. However, a specialized algorithm that is only applicable to the NTWE problem would also be interesting, in some sense indicating that the NTRU and LWE problems are hard, but a mix between them is easier than we expect.

Although not as suitable for a public-key cryptosystem, a generalization of the NTWE problem seems to even better capture this mix between the NTRU and LWE problems. An instance from this generalized problem is given by $(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} = (\overline{\boldsymbol{A}\boldsymbol{S}} + \overline{\boldsymbol{E}})\overline{\boldsymbol{F}}^{-1})$ for $\overline{\boldsymbol{F}} \leftarrow (\psi^{t\times t})^*$, $\overline{\boldsymbol{A}} \leftarrow \mathcal{U}(R_q^{h\times k})$, $\overline{\boldsymbol{S}} \leftarrow \psi^{k\times t}$ and $\overline{\boldsymbol{E}} \leftarrow \psi^{k\times t}$. With $k = 0$, this is exactly a rank t module NTRU instance while the problem with $t = 1$ is a rank k NTWE instance. By instead considering the problem with $n = 1$ and $t = 1$, this problem is essentially the same as an unstructured LWE problem with secret dimension k.

The natural lattice for all of these instances is spanned by the columns of

$$\begin{bmatrix} q\boldsymbol{I} & \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

where $\boldsymbol{A}, \boldsymbol{B}$ are the integer matrix corresponding to $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{B}}$. In this full class of problems, the solution is given by a short vector in this lattice, with the length of the target vector only dependent on the lattice dimension. As such, by current understanding, there should be no significant difference in how hard these lattice problems are to solve if hn and $(k + t)n$ are constant.

We believe that any algorithm against some version of this problem may provide interesting insights for other versions of this problem. In particular, an algorithm that is relevant against either the NTRU or the LWE problem, but not the other, will be applicable to some versions of this generalized NTWE problem. Investigating which versions such an algorithm is applicable to could potentially give a better understanding of the limitations and possibilities of such an algorithm. For example, it may be interesting to investigate how attacks against overstretched NTRU parameters fare against this larger class of problems and if such an attack can be used against versions of this problem that are more similar to the LWE problem.

Acknowledgments

This research has been supported in part by the Swedish Armed Forces and was conducted at KTH Center for Cyber Defense and Information Security (CDIS). The author would like to thank Johan Håstad and Martin Ekerå for their helpful feedback and comments.

References

[1] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016: 25th USENIX Security Symposium*, pages 327–343, Austin, TX, USA, August 10–12, 2016. USENIX Association.

[2] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert Krauthgamer, editor, *27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 10–24, Arlington, VA, USA, January 10–12, 2016. ACM-SIAM.

[3] Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang. NTRU Prime. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`.

[4] Sanjit Chatterjee, Neal Koblitz, Alfred Menezes, and Palash Sarkar. Another look at tightness II: Practical issues in cryptography. Cryptology ePrint Archive, Report 2016/360, 2016. `https://eprint.iacr.org/2016/360`.

[5] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa. NTRU. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`.

[6] Yuanmi Chen. *Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe.* PhD thesis, Université Paris Diderot, 2013. 2013PA077242.

[7] Chitchanok Chuengsatiansup, Thomas Prest, Damien Stehlé, Alexandre Wallet, and Keita Xagawa. ModFalcon: Compact signatures based on module-NTRU lattices. In Hung-Min Sun, Shiuh-Pyng Shieh, Guofei Gu, and Giuseppe Ateniese, editors, *ASIACCS 20: 15th ACM Symposium on Information, Computer and Communications Security*, pages 853–866, Taipei, Taiwan, October 5–9, 2020. ACM Press.

[8] Léo Ducas and Ludo Pulles. Does the dual-sieve attack on learning with errors even work? Cryptology ePrint Archive, Report 2023/302, 2023. `https://eprint.iacr.org/2023/302`.

[9] Léo Ducas and Wessel P. J. van Woerden. NTRU fatigue: How stretched is overstretched? In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part IV*, volume 13093 of *Lecture Notes in Computer Science*, pages 3–32, Singapore, December 6–10, 2021. Springer, Heidelberg, Germany.

[10] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener,

editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Heidelberg, Germany.

[11] Qian Guo and Thomas Johansson. Faster dual lattice attacks for solving LWE with applications to CRYSTALS. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part IV*, volume 13093 of *Lecture Notes in Computer Science*, pages 33–62, Singapore, December 6–10, 2021. Springer, Heidelberg, Germany.

[12] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, Heidelberg, Germany, June 1998.

[13] Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched NTRU parameters. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 3–26, Paris, France, April 30 – May 4, 2017. Springer, Heidelberg, Germany.

[14] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339, San Francisco, CA, USA, February 14–18, 2011. Springer, Heidelberg, Germany.

[15] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. Cryptology ePrint Archive, Report 2012/230, 2012. `https://eprint.iacr.org/2012/230`.

[16] Vadim Lyubashevsky and Gregor Seiler. NTTRU: Truly fast NTRU using NTT. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(3):180–201, 2019. `https://tches.iacr.org/index.php/TCHES/article/view/8293`.

[17] MATZOV. Report on the security of lwe: Improved dual lattice attack. Technical report, MATZOV, 2022.

[18] Chris Peikert. A decade of lattice cryptography. Cryptology ePrint Archive, Report 2015/939, 2015. `https://eprint.iacr.org/2015/939`.

[19] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press.

[20] C. P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66(1):181–199, 1994.

[21] C.P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2):201–224, 1987.

[22] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, Damien Stehlé, and Jintai Ding. CRYSTALS-KYBER. Technical report, National Institute of Standards and

Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[23] Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 27–47, Tallinn, Estonia, May 15–19, 2011. Springer, Heidelberg, Germany.

[24] Yang Wang and Mingqiang Wang. Provably secure NTRUEncrypt over any cyclotomic field. In Carlos Cid and Michael J. Jacobson Jr:, editors, *SAC 2018: 25th Annual International Workshop on Selected Areas in Cryptography*, volume 11349 of *Lecture Notes in Computer Science*, pages 391–417, Calgary, AB, Canada, August 15–17, 2019. Springer, Heidelberg, Germany.