A Thorough Treatment of Highly-Efficient NTRU Instantiations

Julien Duman$^{[0000-0002-5195-1290]1}$, Kathrin Hövelmanns$^{[0000-0002-5478-0140]2}$, Eike Kiltz$^{[0000-0003-1178-048X]1}$, Vadim Lyubashevsky[3], Gregor Seiler[3], and Dominique Unruh$^{[0000-0001-8965-1931]4}$

[1] Ruhr-Universität Bochum
[2] TU Eindhoven
[3] IBM Research Europe, Zurich
[4] University of Tartu

Abstract. Cryptography based on the hardness of lattice problems over polynomial rings currently provides the most practical solution for public key encryption in the quantum era. Indeed, three of the four schemes chosen by NIST in the recently-concluded post-quantum standardization effort for encryption and signature schemes are based on the hardness of these problems. While the first encryption scheme utilizing properties of polynomial rings was NTRU (ANTS '98), the scheme that NIST chose for public key encryption (CRYSTALS-Kyber) is based on the hardness of the somewhat-related Module-LWE problem. One of the reasons for Kyber's selection was the fact that it is noticeably faster than NTRU and a little more compact. And indeed, the practical NTRU encryption schemes in the literature generally lag their Ring/Module-LWE counterparts in either compactness or speed, or both.

In this paper, we put the efficiency of NTRU-based schemes on equal (even slightly better, actually) footing with their Ring/Module-LWE counterparts. We provide several instantiations and transformations, with security given in the ROM and the QROM, that are on par, compactness-wise, with their counterparts based on Ring/Module-LWE. Performance-wise, the NTRU schemes instantiated in this paper over NTT-friendly rings of the form $\mathbb{Z}_q[X]/(X^d - X^{d/2} + 1)$ are the fastest of all public key encryption schemes, whether quantum-safe or not. When compared to the NIST finalist NTRU-HRSS-701, our scheme is 15% more compact and has a 15X improvement in the round-trip time of ephemeral key exchange, with key generation being 35X faster, encapsulation being 6X faster, and decapsulation enjoying a 9X speedup.

1 Introduction

The NTRU encryption scheme [19] was the first truly practical scheme based on the hardness of lattice problems over polynomial rings and, in many ways, the first really practical quantum-safe encryption scheme. The hardness of NTRU was originally stated as its own assumption, but as lattice cryptography evolved over the next few decades, the most natural way to view the hardness behind the NTRU encryption scheme was as a combination of two assumptions over a polynomial ring $R = \mathbb{Z}_q[X]/(f(X))$. The first assumption, which we call the NTRU assumption, is that the quotient of two polynomials $\mathbf{f}$ and $\mathbf{g}$, with coefficients chosen from some narrow distribution, looks uniform in R. The second assumption, which later became known as the Ring-LWE assumption [25, 30] states that given a uniformly random $\mathbf{h} \in R$, and $\mathbf{hr} + \mathbf{e}$, for polynomials $\mathbf{e}$ and $\mathbf{r}$ with coefficients from a narrow distribution, it is difficult to recover $\mathbf{e}$. One could eliminate the need for the first assumption by choosing a relatively wide distribution for $\mathbf{f}$ and $\mathbf{g}$ [29], but the resulting scheme becomes very inefficient; thus all practical instantiations of NTRU were based on these two assumptions.

Since Regev's seminal work constructing an encryption scheme based on the LWE problem over general lattices [28], and its subsequent porting to lattices over polynomial rings [23, 25, 30], most of the community effort of shifted to building encryption schemes that do not require the NTRU assumption, and are just based on the decisional version (which was shown to be equivalent to the search one in [25], and for which no faster practical algorithm is known) of the Ring/Module-LWE problems. Indeed, in the first round of the NIST call for quantum-safe encryption, only 3 out of 17 proposals for lattice-based encryption schemes over polynomial rings relied on the NTRU assumption, while the rest used just an LWE-type assumption.

There are a few reasons for avoiding the NTRU assumption. The first is that the additional NTRU assumption is known to be false in the regime where the modulus q of the ring is noticeably larger than the dimension [1, 8, 15, 22] (for the same parameters, the Ring-LWE problem is still believed to be hard). While the attacks against this parameter regime have not been extended to the one used for public key encryption, it does give some reason for concern. Secondly, in many rings, the division operation is significantly more expensive than multiplication, and so the assumption was also avoided for efficiency considerations. And third is that the NTRU assumption does not naturally lend itself to more flexible instantiations, such as Module-LWE. That is, it naturally operates over a module of dimension 1 (again, due to the division operation), whereas LWE-based schemes can be extended to work over modules of a larger dimension. This has the advantage that the underlying ring operations do not need to change as one increases the security parameter. In fact, all of the non-NTRU finalists in the NIST post-quantum standardization process use the module structure [5, 10]. These schemes are also significantly more efficient than the finalist NTRU-based proposal [21].[5]

There are, however, also several advantages to NTRU-based schemes. One real-world advantage that NTRU has is that all patents on it have expired, while there may still conceivably be some (possibly still hidden) intellectual property claims on the Ring/Module-LWE schemes. Also, NTRU may have practical advantages when used in certain scenarios involving zero-knowledge proofs, since

[5] The schemes [5, 10] can be made even more efficient by eliminating an unnecessary input to the random oracle (see [17]) which did not exist in [21].

the ciphertext has a simpler form and thus may require shorter proofs that it was correctly formed. In this paper, our goal is to put NTRU-based constructions on equal footing, performance-wise, as schemes based on Ring/Module-LWE.

1.1 Speed

The most efficient lattice-based schemes are those that natively work over rings $\mathbb{Z}_q[X]/(f(X))$ that support the Number Theory Transform (NTT). When the polynomial $f(X)$ factors into components having small degree, one can perform multiplication (and division) in the ring using the Chinese Remainder Theorem. That is, one evaluates the multiplicands modulo these factors, performs component-wise multiplication, and finally converts the product back into the original form. The process of efficiently doing these computations is the NTT and the inverse NTT.

The most commonly used NTT-friendly ring is of the form $\mathbb{Z}_q[X]/(X^d+1)$, where d is a power-of-2. For well-chosen q, the polynomial $X^d+1=(X^{d/2}-r)(X^{d/2}+r) \bmod q$, and the respective factors similarly split as $(X^{d/2}-r)=(X^{d/4}-\sqrt{r})(X^{d/4}+\sqrt{r}) \bmod q$, etc. until one reaches an irreducible polynomial of a small (usually 1 or 2) degree. Because of this very nice factorization (the "niceness" mainly rests in the fact that all factors have 2 non-zero coefficients, making reduction modulo them linear-time), evaluation of any polynomial modulo the irreducible factors can be done using approximately $2d\log d$ operations over $\mathbb{Z}_q$. These rings also have some very nice algebraic properties – in particular the expansion factor [24] controlling the growth of polynomial products in the ring is the minimal of all rings. The one disadvantage of these rings is that they are sparse and so one cannot always find one for an appropriate security level. The hardness of the NTRU and Ring-LWE problem directly depends on the degree of the polynomial $f(X)$. Based on the current state of knowledge, obtaining 128-256 bit hardness requires taking dimensions somewhere between 512 and 1024. Since there are no powers of 2 in between, and because one may need to go beyond 1024 in case somewhat better algorithms are discovered, the sparsity of these rings is an inconvenience. The Module-LWE problem overcomes this inconvenience because the problem instance can be made up of a matrix of smaller rings, but this does not work for NTRU because this approach would significantly increase the size of the public key.

One can overcome this issue in NTRU by using "NTT-friendly" rings $f(X)=X^d-X^{d/2}+1$ where $d=2^i3^j$.[6] The rings $\mathbb{Z}_q[X]/(X^d-X^{d/2}+1)$, for appropriately-chosen primes, also support efficient NTT because $X^d-X^{d/2}+1=(X^{d/2}+\zeta)(X^{d/2}-(\zeta+1)) \bmod q$, where ζ is a third root of unity in $\mathbb{Z}_q$ (not equal to 1). And after that, every term (X^k-r) factors into either $(X^{k/2}-\sqrt{r})(X^{k/2}+\sqrt{r})$ or into $(X^{k/3}-\sqrt[3]{r})(X^{k/3}-\zeta\sqrt[3]{r})(X^{k/3}-\zeta^2\sqrt[3]{r})$ modulo q. In both cases, one can efficiently proceed with the very efficient NTT because all factors have two non-zero coefficients. As can be seen from Table 1, there are many such polynomials of degree between 512 and 1024. In the work of [26], a version of NTRU

[6] The polynomial $f(X)$ is therefore the $3d$-th cyclotomic polynomial.

was implemented over the ring $\mathbb{Z}_{7681}[X]/(X^{768}-X^{384}+1)$, but due to the structure of the ring, no factorization into three terms was necessary. In this work we show that there aren't any efficiency issues when the latter does happen, and give an instantiation of a scheme over the ring $\mathbb{Z}_{2917}[X]/(X^{648}-X^{324}+1)$. The conclusion is that all of the schemes in Table 1 should have almost equally good instantiations.

One should also mention that Module and Ring-LWE schemes can be used in non-NTT-friendly rings [9], and the inefficiency of multiplication in these rings can be partially overcome by doing multiplication in a ring with a larger modulus and/or degree of $f(X)$ which supports NTT, and then reducing back into the original ring. This is, however not possible for NTRU-based schemes because NTRU requires polynomial *division*, and it is not known how to map this operation between rings. On the other hand, if a ring supports NTT, then division is essentially as fast as multiplication, with only the operation in the base ring (which is of a very low degree) being different. Thus any hope of having NTRU-based schemes being competitive with Ring/Module-LWE schemes seems to require defining the NTRU encryption scheme directly over NTT-friendly rings.

A reason that NTRU was traditionally not defined over NTT-friendly-rings was presumably due to an attack of Gentry [18] against a version of NTRU over the ring $\mathbb{Z}_q[X]/(X^d-1)$, where the polynomial X^d-1 could be factored as $(X^{d/2}-1)(X^{d/2}+1)$. The observation was that instead of working over the ring $\mathbb{Z}_q[X]/(X^d-1)$, one can reduce everything modulo $X^{d/2}-1$ and work over the ring $\mathbb{Z}_q[X]/(X^{d/2}-1)$. What makes the attack work is that reduction modulo $X^{d/2}-1$ is a ring homomorphism and that this reduction increases the size of the maximum coefficient by at most a factor of 2. Thus one can solve a shortest vector problem (upon which NTRU is based) in a lattice with a significantly smaller dimension, but whose norm increased by only a factor of 2. From this attack, one might infer that it's important to have the polynomial $f(X)$ be irreducible (or have a large component of it be irreducible). Interestingly, however, the theoretical works of [23–25] showed that in the reductions from worst-case lattice problems to average-case problems over polynomial rings (e.g. Ring/Module-SIS, Ring/Module-LWE), one needs the polynomial $f(X)$ to be *irreducible* in $\mathbb{Z}[X]$, but the polynomial $f(X)$ splitting in $\mathbb{Z}_q[X]$ does not seem to make the average-case problem easier.[7] And in fact, most practical lattice-based constructions work over the ring $\mathbb{Z}_q[X]/(X^d+1)$, where d is a power of 2. While polynomials X^d+1 are irreducible in $\mathbb{Z}[X]$, they are *always* reducible in $\mathbb{Z}_q[X]$; and consistent with the theoretical intuition, there have not been any

[7] As a sanity check, one can see that the attack in [18] does not work because it is impossible for a polynomial $f(X)$ that's irreducible over the integers to split modulo q into polynomials of large degree (e.g. $d/2$) whose coefficients are small. For example, it's trivial to see that X^d+1 cannot have factors $X^{d/2}\pm\beta$ with $\beta<\sqrt{q}$. For a more general result, one needs a little algebraic number theory (e.g. implicit in the proof of [27, Lemma 3.1] is that any factor of degree d/k of X^d+1 has ℓ_2-norm at least $p^{1/k}$, and this result extends in a similar way to other polynomials).

attacks exploiting the factorization of $X^d + 1$ modulo q. We therefore don't see any danger of using NTRU over NTT-friendly rings.

1.2 Decryption Error and Compactness

To make NTRU encryption work efficiently over NTT-friendly rings, one creates the public key as $\mathbf{h} = p\mathbf{g}/(p\mathbf{f} + 1)$, for a small prime p, and then the encryption function (which is one-way CPA secure – meaning that it is hard to decrypt for a random message) outputs $\mathbf{c} = \mathbf{hr} + \mathbf{m}$, where $\mathbf{r}, \mathbf{m}$ are polynomials with coefficients coming from a narrow distribution. The decryption algorithm computes $(p\mathbf{f} + 1)\mathbf{c} = p(\mathbf{gr} + \mathbf{fm}) + \mathbf{m}$. If the coefficients of the product $p(\mathbf{gr} + \mathbf{fm})$ are smaller than $q/2$, then one can recover $\mathbf{m}$ by taking the above value modulo p.

One important area of optimization (and what was already recognized in the original NTRU scheme [19]) is that the product $p(\mathbf{gr}+\mathbf{fm})$ does not *always* need to be less than $q/2$, but only with very high probability. On the one hand, this probability should be negligible, as obtaining decryption failures on honestly-generated ciphertexts is the folklore way of recovering the secret key in LWE-based schemes. On the other hand, the decryption error can be defined as an *information-theoretic* quantity. Unlike the security parameter, there is therefore no safety margin needed as there is no danger of a better algorithm being found to lower this quantity.

To make the decryption error an information-theoretic quantity, one should define it as being worst-case when the adversary is even given the secret key [20]. In LWE-based schemes, the message is an *additive* term in the decryption procedure, and since the message's coefficients are generally small (normally in $\{0, 1\}$), there is no difference between a worst-case and an "average-case" (or even best-case) message. In NTRU, however, as we saw from the decryption equation, we need the quantity $p(\mathbf{gr} + \mathbf{fm})$ to be smaller than $q/2$, and $\mathbf{m}$ is multiplied by $\mathbf{f}$. Purposefully choosing a "bad" $\mathbf{m}$ can, therefore, make a large difference (increasing the decryption error by factors larger than 2^{100} is normal for standard parameter choices). The naive way to keep the worst-case decryption error small is to increase the modulus q so that encryption errors do not occur. But increasing q weakens the security of the scheme by making the lattice-reduction algorithms more effective.

In this paper, we demonstrate three different ways of handling the decryption error. The first way is a generic transformation ACWC_0 from any scheme into one in which the message does not affect the decryption error. Hence the worst-case correctness error of the transformed scheme equals the average-case correctness error of the original scheme. This transformation is most likely folklore, and it is presented in Figure 5 on page 16. The downside of this transformation is that it increases the ciphertext size by the message length.

The two next manners in which a worst-case decryption error is handled preserves the ciphertext size of the underlying scheme. The transformation ACWC (Figure 6 on page 18) requires some specific properties of the distribution from which the message is generated. A natural distribution that satisfies this property is having coefficients uniformly-random modulo p. When p is not a power of 2, this distribution is not particularly pleasant to sample with AVX2 optimizations (due to the branching caused by rejection sampling), and so it was proposed in [21] to sample the distribution as a binomial distribution modulo p. Since the binomial distribution is very easy to sample by summing up and subtracting random bits, and because this value modulo p is pretty close to the uniform distribution, this is a more preferable way of sampling the secret coefficients. Still, being required to only sample the message $\mathbf{m}$ according to the uniform distribution could be an acceptable compromise. It is an interesting open problem as to whether our transformation can still be proved secure under the same assumptions for a different, more easily sampled, distribution of the message.

Our final way of handling adversarial-generated messages does not involve any transformation, but rather shows how for certain distributions of $\mathbf{m}$, the worst-case decryption error is not much worse than the average-case (or best-case), as in LWE-based schemes. Consider the coefficients of $\mathbf{m}$ as consisting of a message part μ and an error part ϵ. One has this implicit split by defining a function $f(\mu, \epsilon) = \mathbf{m}$ in a particular way where ϵ and μ are sampled independently. A property that we need from f is that $f(\mu, \epsilon) \bmod 2 = \mu$. Thus if one recovers $\mathbf{m}$, one can also recover μ. If we want to choose $\mathbf{m}$ according to the binomial distribution (as in e.g. NewHope [3], Kyber [5], or Saber [10]), then f can be a very simple function as described in Lemma 4.1. And of course, we also want the decryption error of this function to be approximately the same for all adversarially-chosen μ. It turns out that because the adversary only gets to set the residue modulo 2 in the binomial distribution, he has no control over the sign of the final output, nor the variance of the conditional distribution. And for this reason, the worst-case error distribution is close to the random one.

A further observation is that if we only need to recover $\mu = \mathbf{m} \bmod 2$, then there is no need to set the parameter p large enough so as to be able to recover the entire $\mathbf{m}$. In particular, we could just set $p = 2$ and the decryption procedure would still work. By decoupling the magnitude of p from the magnitude of the coefficients of $\mathbf{m}$, we can set $\mathbf{m}$ to be large (which increases the hardness of Ring-LWE), while keeping $p = 2$. The value of p has no effect on the hardness of any version of Ring-LWE (since $p\mathbf{h}$ is as uniform as just $\mathbf{h}$), and based on the state of affairs regarding solving Ring-LWE problems, finding $\mathbf{m} \bmod 2$ is as hard as finding $\mathbf{m}$. We discuss the complexity of this problem in Section 4.3 and present the scheme in Section 4.4.

1.3 Proofs in the (Q)ROM

Our two transformations ACWC_0 and ACWC are defined relative to random oracles, and have proofs in the ROM that are conceptually very simple. We show that ACWC_0 transforms any one-way secure ($\mathsf{OW\text{-}CPA}$) encryption scheme into one that is $\mathsf{IND\text{-}CPA}$ secure, and that ACWC transforms any $\mathsf{OW\text{-}CPA}$ secure encryption scheme into one that is also $\mathsf{OW\text{-}CPA}$ secure. Note that we cannot prove $\mathsf{IND\text{-}CPA}$ security of ACWC since there exist instantiations for which application of ACWC yields a scheme that simply isn't $\mathsf{IND\text{-}CPA}$ secure.[8] By working with q-$\mathsf{OW\text{-}CPA}$ security,[9] a slight generalisation of $\mathsf{OW\text{-}CPA}$ security, we can combine the aforementioned transformations with the well-known Fujisaki-Okamoto transformation $\mathsf{FO}^{\perp}$ in a way such that we obtain a tight proof for the resulting KEMs.

Since post-quantum security is a central goal of the constructions in this paper, we also prove all our results in the quantum random oracle model (QROM). That is, we show the security even if the adversary can perform queries to the random oracle in superposition between different inputs. The two constructions involving the random oracle are ACWC_0 and ACWC. We show that ACWC_0 transforms a one-way secure ($\mathsf{OW\text{-}CPA}$) encryption scheme into an IND-CPA secure one. This proof is a reasonably straightforward application of the one-way to hiding theorem, O2H [31] in the variant from [4]. (O2H is a common technique used in random oracle proofs for encryption schemes.) The drawback of the use of O2H is that it introduces a square-root in the adversary's advantage. (That is, if the adversary has ε advantage against the underlying scheme and it makes q random oracle queries, then it has advantage $O(\sqrt{q^2\varepsilon})$ against the result of the transformation.)

In contrast, security of ACWC does not have an obvious proof using O2H. Instead, we use the measure-and-reprogram technique (M&R) from [11, 13]. This technique was developed for proving the security of the Fiat-Shamir transform. The fact that this technique works here is unexpected for two reasons: First, it was designed specifically with transformations of sigma-protocols (or related structures) into signatures or non-interactive proof systems in mind; transformations of encryption schemes such as ACWC have a very different structure. Second, M&R is a technique for adaptive reprogramming of the random oracle: Its core feature is, on a high level, that we can measure a query that the adversary will use later for its attack (e.g., as part of a forged signature), and sneak in a value of interest Θ into the answer to exactly that query (e.g., the challenge in a sigma-protocol). But in our setting, there is no such value of interest Θ. (We use a random value Θ when invoking the M&R theorem because that is technically required, but we would be perfectly happy if the random oracle was not reprogrammed at all.) We thus "misuse" the M&R for a situation where reprogramming is not required in the first place. This raises the interesting open question whether there could be variants of the M&R theorem that only cover the measurement-part of it (without reprogramming) but have tighter parameters and could be used in situations such as ours to produce a tighter reduction.

[8] Say that PKE has message space $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$, and say that PKE's encryptions of messages $M_1||M_2$ leak M_1 and the first bit of M_2. When instantiated with the classical one-time-pad, ACWC encrypts a message m by sampling a message $M_1 \leftarrow \mathcal{M}_1$ and encrypting $M_1||m \oplus \mathsf{F}(M_1)$, thereby leaking the first bit of m.

[9] In q-$\mathsf{OW\text{-}CPA}$ security the adversary is given an encryption of a random plaintext and wins if it returns a set of cardinality at most q containing the plaintext. For $q = 1$ this is $\mathsf{OW\text{-}CPA}$ security.

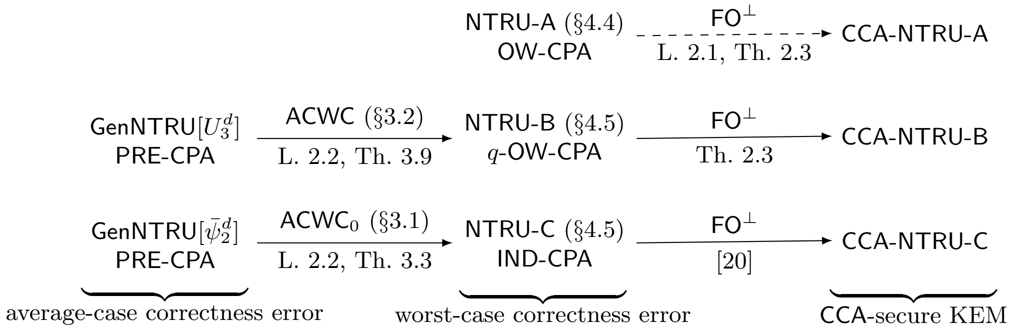

Fig. 1. Overview: How to obtain efficient IND-CCA-secure KEMs from our NTRU-based PKE schemes. Solid arrows indicate tight reductions in the ROM, dashed arrows indicate non-tight reductions. q-OW-CPA is a strengthening of standard OW-CPA security, where the adversary is allowed to return q many guesses (instead of just one). PRE-CPA security stands preimage resistance which in the setting of NTRU is essentially equivalent to OW-CPA security.

Furthermore, the use of M&R also leads to better parameters than we got using O2H: The advantage of the adversary against the result of the transformation ACWC is $O(q^2\varepsilon)$, i.e., no square-root is involved. (However, in contrast to ACWC$_0$, we only get one-way security. This is not a limitation of the proof technique, though, but stems from the fact that ACWC does not achieve IND-CPA security. But note that in a setting were we only need one-way security, we still do not have a better bound than $\sqrt{q^2\varepsilon}$ for ACWC$_0$; in this case, ACWC gives strictly better security.)

1.4 Concrete Results and Comparison to the State of the Art

We now describe the various ways that one can instantiate NTRU using the techniques described in this paper and compare it to other lattice-based schemes. We defined three different ways to instantiate NTRU, with all three approaches being in the same ring and only differing in the secret distributions and the manner in which it is transformed into a scheme with a small "worst-case" decryption error. When working over the ring $\mathbb{Z}_q[X]/(X^d - X^{d/2} + 1)$, we will write NTRU-A$_q^d$ to be the scheme in Figure 7 which did not require any transformation. By NTRU-B$_q^d$, we denote the scheme presented in Figure 9 which is derived from the generic NTRU scheme GenNTRU (Figure 8) by utilizing the size-preserving transformation from Figure 6. And by NTRU-C$_q^d$, we refer to the scheme in Figure 10 derived from the folklore transformation of the generic NTRU scheme GenNTRU (Figure 8) in Figure 5. All of the aforementioned schemes are CPA-secure, and we use the standard FO-transformation from Figure 4 to create a CCA-KEM. The above is summarized in the overview Figure 1.

In Table 1, we summarize the "interesting" instantiations of the schemes described in this paper having between 150 and 350 bits of security. We also compare these to other instantiations of NTRU and Module-LWE based schemes in Figure 2. For a consistent evaluation of security, we used the online LWE hardness estimator [2]. This estimator has undergone some updates since its initial release, but still does not (as of this writing) include some recent cryptanalytic techniques (e.g. [14]) which could lower the security a little bit. Nevertheless, it still provides very meaningful results for comparing between various schemes.

In comparison to NTRU-HRSS, which was a finalist in the NIST standardization process, $\mathsf{NTRU\text{-}C}_{2917}^{648}$ is based on an NTRU problem with the same error distribution, and has an approximately equal security level. But due to the fact that we show how to control the worst-case decryption error, the ciphertext/public key sizes are 15% smaller. If one looks at $\mathsf{NTRU\text{-}C}_{3457}^{768}$, which has a similar public key/ciphertext size as NTRU-HRSS-701, one sees that the tradeoff for no error vs. 2^{-252} error is 30 bits of security, and the difference in security is even larger if one considers the $\mathsf{NTRU\text{-}A}$ version. In our opinion, exchanging such a large security margin in return for reducing 2^{-250} to 0 in the information-theoretic decryption error term, is not a sensible trade-off. The comparison of our NTRU instantiations to Kyber shows that the two schemes are essentially on the same size/security curve.

We produced a sample implementation of $\mathsf{NTRU\text{-}A}_{2917}^{648}$, as it is most similar in security to NTRU-HRSS-701. In table 3, we compare this scheme to NTRU-HRSS and other highly-efficient lattice-based schemes such as Kyber and NTTRU. The efficiency of our implementation is similar to that of Kyber-512, even though the NTRU variant has about 30 extra bits of security. The efficiency improvement is due to the fact that there is no matrix sampling required in NTRU-based schemes. When compared to NTRU-HRSS-701, there is a clear difference in efficiency, with $\mathsf{NTRU\text{-}A}$ being over 15X faster for round-trip ephemeral key exchange. The running time of $\mathsf{NTRU\text{-}C}$ should be quite similar, and $\mathsf{NTRU\text{-}B}$ will be a little hampered by the more complicated (uniform vs. binomial) error distribution, but should also be close.

While all the parameters in Table 1 are over rings of the form $\mathbb{Z}_q[X]/(X^d - X^{d/2} + 1)$, we mention that another interesting instantiation would be a version of $\mathsf{NTRU\text{-}A}$ from Figure 7 with $\eta = \psi_3^d$ over the ring $\mathbb{Z}_{3329}[X]/(X^{512} + 1)$. This would have exactly the security of Level 1 Kyber, a decryption error of 2^{-197}, and public key / ciphertext sizes of 768 bytes. The parameters make it an attractive NIST level 1 candidate. The one difference is that the inertia degree would be 4, which requires one to do inversions and multiplications in degree 4 rings, but we don't believe that this should cause a noticeable slowdown.

d	q	inertia	pk &	$\log_2(\delta)$	security	$\log_2(\delta)$	security	$\log_2(\delta)$	security
(dim.)	(mod.)	degree	ct (B) [a]	CCA-NTRU-A		CCA-NTRU-B		CCA-NTRU-C	
576	2593	2	864	-150	162	-165	155	-187	153
576	3457	1	864	-257	157	-297	150	-333	149
648	2917	2	972	-170	180	-187	172	-211	171
648	3889	1	972	-289	175	-335	166	-376	165
768	3457	2	1152	-202	210	-222	201	-252	199
864	3457	3	1296	-182	238	-197	227	-224	225
972	3889	3	1458	-206	265	-223	253	-253	251
1152	3457	1	1728	-140	321	-147	306	-167	304
1296	3889	1	1944	-158	358	-166	342	-189	339
1296	6481	3	2106	-420	339	-471	324	-530	322

[a] The ciphertext size for NTRU-C is 32 bytes larger.

Table 1. Parameters for the NTRU schemes CCA-NTRU-A, CCA-NTRU-B, and CCA-NTRU-C from this paper. All of the variants of the NTRU schemes work over the same ring, with the only difference being the underlying distributions of the secrets and messages, as well as the transformation (if one is necessary) from an instance with worst-case decryption error to one with average-case. The public key and ciphertext are of the same length (except for the ciphertext of CCA-NTRU-C, which is 32 bytes larger) and it is reported in bytes. The inertia degree is the smallest degree of the polynomial ring over which one has to perform operations at the bottom of the NTT tree (for efficiency, one may not always want to split down to the smallest possible degree, though). The parameter δ is the decryption error for a worst-case message (computed via a Pari script), and the security (in the ROM) is obtained using the LWE estimator script [2].

	dimension	modulus	pk (B)	ct (B)	$\log_2(\delta)$	security
Kyber-512	512	3329	800	768	-139	148
Kyber-768	768	3329	1184	1088	-164	212
Kyber-1024	1024	3329	1568	1568	-174	286
NTTRU	768	7681	1248	1248	-1217	183
NTRU-HRSS-701	701	8192	1138	1138	$-\infty$	166
NTRU-HRSS-1373	1373	16384	2401	2401	$-\infty$	314

Table 2. Comparison to Existing Work. The Kyber parameters are taken from the Round 3 submission to the NIST PQC Standardization Process. The NTTRU parameters are from [26], and the NTRU-HRSS-701 parameters are from [21], and the NTRU-HRSS-1373 instantiation is from the comments to the NIST PQC mailing list. For consistency of comparing these schemes to those in Table 1, the security of the schemes are computed using the LWE estimator script [2].

Scheme	Key Gen	Encaps	Decaps	Total Round-Trip
$\mathsf{CCA\text{-}NTRU\text{-}A}^{648}_{2917}$ (This Paper)	6.2K	5.6K	7.3K	19.1K
NTRU-HRSS-701	220.3K	34.6K	65K	319.9K
NTTRU	6.4K	6.1K	7.9K	20.4K
Kyber-512 (90's)	6.2K	7.9K	9.2K	23.3K
Kyber-768 (90's)	11K	13.1K	14.8K	38.9K

Table 3. Number of cycles (on a Skylake machine) for various operations of a CCA-secure KEM. The numbers for Kyber-512 and Kyber-768 are taken from [17, Table 3], which shows an improved implementation of Kyber90's (i.e. the version using AES and SHA-256 instead of SHAKE) when using prefix hashing and employing an explicit reject in the decapsulation procedure.

2 Preliminaries

2.1 Notation

If $\mathcal{M}$ is a finite set and $\psi_{\mathcal{M}}$ is a distribution on $\mathcal{M}$, then $m \leftarrow \psi_{\mathcal{M}}$ samples m from $\mathcal{M}$ according to $\psi_{\mathcal{M}}$. We write $m \leftarrow \mathcal{M}$ to denote sampling according to the uniform distribution. For a random variable X, $H_\infty(X)$ denotes its min-entropy.

For the sake of completeness, we summarise all relevant quantum preliminaries in the full version [16].

2.2 Cryptographic Definitions

Public-Key Encryption. A public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ consists of three algorithms, a probability distribution $\psi_{\mathcal{M}}$ on a finite message space $\mathcal{M}$. If no probability distribution is specified we assume $\psi_{\mathcal{M}}$ to be the uniform distribution. The key generation algorithm KeyGen outputs a key pair (pk, sk), where pk also defines a finite randomness space $\mathcal{R} = \mathcal{R}(pk)$. The encryption algorithm Enc, on input pk and a message $m \in \mathcal{M}$, outputs an encryption $c \leftarrow \mathsf{Enc}(pk, m)$ of m under the public key pk. If necessary, we make the used randomness of encryption explicit by writing $c := \mathsf{Enc}(pk, m; r)$, where $r \in \mathcal{R}$. By $\psi_{\mathcal{R}}$ we denote be the distribution of r in Enc, which we require to be independent of m. The decryption algorithm Dec, on input sk and a ciphertext c, outputs either a message $m = \mathsf{Dec}(sk, c) \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$ to indicate that c is not a valid ciphertext.

Randomness Recoverability. PKE is randomness recoverable (RR) if there exists an algorithm $\mathsf{Recover}$ such that for all $(pk, sk) \in \mathsf{supp}(\mathsf{Gen})$ and $m \in \mathcal{M}$, we have that

$$\Pr\left[\forall m' \in \mathsf{Preimg}(pk, c) \colon \mathsf{Enc}(pk, m'; \mathsf{Recover}(pk, m', c)) \neq c \mid c \leftarrow \mathsf{Enc}(pk, m)\right] = 0\,,$$

where the probability is taken over $c \leftarrow \mathsf{Enc}(pk, m)$ and $\mathsf{Preimg}(pk, c) := \{m \in \mathcal{M} \mid \exists r \in \mathcal{R} \colon \mathsf{Enc}(pk, m; r) = c\}$. Additionally, we will require that $\mathsf{Recover}$ returns $\perp$ if it is run with input $m \notin \mathsf{Preimg}(pk, c)$.

Correctness Error. PKE has (worst-case) correctness error δ [20] if

$$\mathbb{E}\left[\max_{m\in\mathcal{M}} \Pr\left[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) \neq m\right]\right] \leq \delta\,,$$

where the expectation is taken over $(pk, sk) \leftarrow \mathsf{Gen}$ and the choice of the random oracles involved (if any). PKE has average-case correctness error δ relative to distribution $\psi_{\mathcal{M}}$ over $\mathcal{M}$ if

$$\Pr\left[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) \neq m\right] \leq \delta\,,$$

where the probability is taken over $(pk, sk) \leftarrow \mathsf{Gen}$, $m \leftarrow \psi_{\mathcal{M}}$ and the randomness of Enc. This condition is equivalent to

$$\mathbb{E}\left[\Pr\left[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) \neq m\right]\right] \leq \delta\,,$$

where the expectation is taken over $(pk, sk) \leftarrow \mathsf{Gen}$, the choice of the random oracles involved (if any), and $m \leftarrow \psi_{\mathcal{M}}$.

Spreadness. PKE is weakly γ-spread [12] if

$$\mathbb{E}\left[\max_{m\in\mathcal{M}, c\in\mathcal{C}} \Pr\left[\mathsf{Enc}(pk, m) = c\right]\right] \leq 2^{-\gamma}\,,$$

where the probability is taken over the random coins of encryptions and the expectation is taken over $(pk, sk) \leftarrow \mathsf{Gen}$.

Security. In the usual one-way game $\mathsf{OW\text{-}CPA}$ for PKE, the adversary has to decrypt a ciphertext c^* of a random plaintext $m^* \leftarrow \psi_{\mathcal{M}}$ by sending *one* candidate m' back to the challenger, and wins if $m' = m^*$. In the generalized q-$\mathsf{OW\text{-}CPA}$ game, the adversary gets to send a set $\mathcal{Q}$ of size at most q and wins if $m^* \in \mathcal{Q}$. The formal definition of q-$\mathsf{OW\text{-}CPA}$ is given in Fig. 2 and the advantage function of an adversary $\mathcal{A}$ is

```
Game $q$-OW-CPA
$(pk, sk) \leftarrow \mathsf{KeyGen}$
$m^* \leftarrow \psi_{\mathcal{M}}$
$c^* \leftarrow \mathsf{Enc}(pk, m^*)$
$\mathcal{Q} \leftarrow \mathcal{A}(pk, c^*)$
return $[\![m^* \in \mathcal{Q} \wedge |\mathcal{Q}| \leq q]\!]$
```

```
Game PRE-CPA
$(pk, sk) \leftarrow \mathsf{KeyGen}$
$m^* \leftarrow \psi_{\mathcal{M}}$
$c^* \leftarrow \mathsf{Enc}(pk, m^*)$
$(m, r) \leftarrow \mathcal{A}(pk, c^*)$
return $[\![\mathsf{Enc}(pk, m; r) = c^*]\!]$
```

```
Game IND-CPA
$(pk, sk) \leftarrow \mathsf{Gen}$
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk)$
$b \leftarrow \{0, 1\}$
$c^* \leftarrow \mathsf{Enc}(pk, m_b)$
$b' \leftarrow \mathcal{A}_2(pk, c^*)$
return $[\![b = b']\!]$
```

Fig. 2. Left: q-Set One-Wayness game q-$\mathsf{OW\text{-}CPA}$ for PKE, where $q = 1$ is standard $\mathsf{OW\text{-}CPA}$. Middle: Preimage resistance game $\mathsf{PRE\text{-}CPA}$ for PKE. Right: game $\mathsf{IND\text{-}CPA}$ for PKE and adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

$$\mathsf{Adv}^{q\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) := \Pr\left[q\text{-}\mathsf{OW\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1\right]\,.$$

For $q = 1$ one recovers standard $\mathsf{OW\text{-}CPA}$ security, i.e., $\mathsf{OW\text{-}CPA} := 1\text{-}\mathsf{OW\text{-}CPA}$. We also introduce preimage resistance of PKE by the defining the advantage function of an adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{PRE\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) := \Pr\left[\mathsf{PRE\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1\right] ,$$

where game $\mathsf{PRE\text{-}CPA}$ is given in Fig. 2.

Finally, we define the $\mathsf{IND\text{-}CPA}$ advantage for an adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) := \left|\Pr\left[\mathsf{IND\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1\right] - \frac{1}{2}\right| ,$$

where the game $\mathsf{IND\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}}$ is defined in Fig. 2.

Lemma 2.1 **(PKE $\mathsf{OW\text{-}CPA}$ $\implies$ PKE q-$\mathsf{OW\text{-}CPA}$).** *For any adversary $\mathcal{A}$ against the q-$\mathsf{OW\text{-}CPA}$ security of PKE, there exists an $\mathsf{OW\text{-}CPA}$ adversary against PKE with*

$$\mathsf{Adv}^{q\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) \leq q \cdot \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}) .$$

where the running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

Proof. Sketch. The reduction $\mathcal{B}$ runs the adversary $\mathcal{A}$ on the inputs it got from its $\mathsf{OW\text{-}CPA}$ challenger and obtains the set $\mathcal{Q}$ of size q. It samples $m \leftarrow \mathcal{Q}$ uniformly at random and forwards m to the $\mathsf{OW\text{-}CPA}$ challenger, with probability $1/q$ it guessed the right one when the solution is contained in $\mathcal{Q}$, thus, the claim follows.

Lemma 2.2 **(PKE $\mathsf{PRE\text{-}CPA}$ and RR $\overset{\textbf{tightly}}{\implies}$ PKE q-$\mathsf{OW\text{-}CPA}$).** *If PKE is randomness recoverable, then for any adversary $\mathcal{A}$ against the q-$\mathsf{OW\text{-}CPA}$ security of PKE, there exists an $\mathsf{PRE\text{-}CPA}$ adversary $\mathcal{B}$ against PKE with*

$$\mathsf{Adv}^{q\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{PRE\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}) .$$

where the running time of $\mathcal{B}$ is about $\mathbf{Time}(\mathcal{A}) + q \cdot (\mathbf{Time}(\mathsf{Recover}) + \mathbf{Time}(\mathsf{Enc}))$.

Proof. The reduction $\mathcal{B}$ forwards to $\mathcal{A}$ the challenge public-key and ciphertext c^* and obtains a set $\mathcal{Q}$. For every $m \in \mathcal{Q}$ it runs $r := \mathsf{Recover}(pk, m, c)$ and then runs $\mathsf{Enc}(pk, m; r)$ to obtain c. If c equals c^* it returns (m, r) as the solution, otherwise it continues with the search. If no element is found it can return a random $m \leftarrow \mathcal{M}$. Clearly, if $\mathcal{A}$ wins, then so does $\mathcal{B}$. Since the reduction $\mathcal{B}$ runs $\mathcal{A}$ once, and algorithms $\mathsf{Recover}$ and Enc at most q many times, the claim follows.

Key-Encapsulation Mechanism. A key encapsulation mechanism $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ consists of three algorithms and a finite key space $\mathcal{K}$ similar to a PKE scheme, but Encaps does not take a message as input. The key generation algorithm Gen outputs a key pair (pk, sk), where pk also defines a finite randomness space $\mathcal{R} = \mathcal{R}(pk)$ as well as a ciphertext space $\mathcal{C}$. The encapsulation algorithm Encaps takes as input a public-key pk and outputs a key encapsulation ciphertext c and a key K, that is $(c, K) \leftarrow \mathsf{Encaps}(pk)$. The decapsulation algorithm Decaps, on input sk and a ciphertext c, outputs either a key $K = \mathsf{Decaps}(sk, c) \in \mathcal{K}$ or a special symbol $\perp \notin \mathcal{K}$ to indicate that c is not a valid ciphertext. We say KEM has correctness error δ if

$$\Pr\left[\mathsf{Decaps}(sk, c) = K \mid (c, K) \leftarrow \mathsf{Encaps}(pk)\right] \leq \delta \ ,$$

where the probability is taken over the randomness in Encaps and $(pk, sk) \leftarrow \mathsf{Gen}$. In terms of KEM's security, we consider the $\mathsf{IND\text{-}CCA}$ advantage function of an adversary $\mathcal{A}$:

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}) := \Pr\left[\mathsf{IND\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}} \Rightarrow 1\right] - \frac{1}{2}$$

where game $\mathsf{IND\text{-}CCA}$ is defined in Fig. 3.

```
IND-CCA
    $(pk, sk) \leftarrow \mathsf{Gen}$
    $(K_0, c^*) \leftarrow \mathsf{Encaps}(pk)$
    $K_1 \leftarrow \mathcal{K}, b \leftarrow \{0, 1\}$
    $b' \leftarrow \mathcal{A}^{\mathsf{Decaps}}(pk, c^*, K_b)$
    return $[\![ b = b' ]\!]$

Decaps($c \neq c^*$)
    return $\mathsf{Decaps}(sk, c)$
```

Fig. 3. Game $\mathsf{IND\text{-}CCA}$ for KEM

The Fujisaki-Okamoto transformation with explicit reject. To a public-key encryption scheme $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and associated uniform distribution over $\mathcal{M}$, randomness space $\mathcal{R}$, and hash functions $\mathsf{H} : \{0, 1\}^* \to \mathcal{R} \times \mathcal{K}$, we associate $\mathsf{KEM} := \mathsf{FO}^{\perp}[\mathsf{PKE}, \mathsf{H}] := (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$. Its constituting algorithms are given in Fig. 4. In [17] it was formally shown that including a *short prefix* of the public-key into the hash function provably improves the multi-user security of the Fujisaki-Okamoto transform. In this work, for simplicity, we will omit this inclusion and analyze the security in the single-user setting.

Theorem 2.3 (q_{H}-$\mathsf{OW\text{-}CPA}$ of PKE $\overset{\mathsf{ROM}}{\Longrightarrow}$ $\mathsf{IND\text{-}CCA}$ of KEM). *For any adversary $\mathcal{A}$, making at most q_D decapsulation, q_H hash queries, against the* $\mathsf{IND\text{-}CCA}$ *security of* KEM, *there exists an adversary $\mathcal{B}$ against the q_H-*$\mathsf{OW\text{-}CPA}$ *security of* PKE *with*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}) \leq \mathsf{Adv}^{q_H\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}) + q_D 2^{-\gamma} + q_H \delta \ ,$$

where the running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

The proof is very similar to formerly known proofs for FO - after showing how to simulate oracle Decaps, we argue that the challenge key cannot be distinguished

```
$\mathsf{Encaps}(pk)$
$m \leftarrow \mathcal{M}$
$(r, K) := \mathsf{H}(m)$
$c := \mathsf{Enc}(pk, m; r)$
return $(K, c)$

$\mathsf{Decaps}^{\perp}(sk, c)$
$m' := \mathsf{Dec}(sk, c)$
$(r', K') := \mathsf{H}(m')$
if $m' = \perp$ or $c \neq \mathsf{Enc}(pk, m'; r')$
    return $\perp$
return $K'$
```

Fig. 4. Key encapsulation mechanism $\mathsf{KEM} = \mathsf{FO}^{\perp}[\mathsf{PKE}, \mathsf{H}]$, obtained from $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with worst-case correctness error.

from random unless the adversary $\mathcal{A}$ queries H on the challenge plaintext. When reducing to plain OW-CPA security, a reduction would have to guess, but a reduction to q_{H}-OW-CPA security can simply keep a list of all of $\mathcal{A}$ queries to H and return this list as the list of plaintext guesses. For the sake of completeness, a full proof is given in in the full version [16].

Theorem 2.4 (IND-CPA of PKE $\overset{\mathsf{ROM}}{\Longrightarrow}$ IND-CCA of KEM [20]). *For any adversary $\mathcal{A}$, making at most q_D decapsulation, q_H hash queries, against the* IND-CCA *security of* KEM*, there exists an adversary $\mathcal{B}$ against the* IND-CPA *security of* PKE *with*

$$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) \leq 2\big(\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}) + q_H / \left|\mathcal{M}\right|\big) + q_D 2^{-\gamma} + q_H \delta \,,$$

where the running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

Theorem 2.5 (OW-CPA of PKE $\overset{\mathsf{QROM}}{\Longrightarrow}$ IND-CCA of KEM [12]). *For any quantum adversary $\mathcal{A}$, making at most q_D decapsulation, q_H (quantum) hash queries, against the* IND-CCA *security of* KEM*, there exists a quantum adversary $\mathcal{B}$ against the* OW-CPA *security of* PKE *with*

$$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) \leq 2q\sqrt{\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B})} + 24q^2\sqrt{\delta} + 24q\sqrt{qq_D} \cdot 2^{-\gamma/4} \,.$$

where $q := 2(q_H + q_D)$ and $\mathbf{Time}\,(\mathcal{B}) \approx \mathbf{Time}\,(\mathcal{A}) + \mathsf{O}(q_H \cdot q_D \cdot \mathbf{Time}\,(\mathsf{Enc}) + q^2)$.

3 Worst-Case to Average-Case Decryption Error

In this section we introduce two worst-case to average case correctness transform for public-key encryption.

3.1 Simple transformation ACWC_0 with redundancy

Let PKE be an encryption scheme with small average-case correctness error and F be a random oracle. We first introduce a simple transformation ACWC_0 by describing $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$ in Fig. 5 which adds λ bits of redundancy to the ciphertexts, where λ is the size of the message space. The resulting scheme has small worst-case correctness error.

```
$\mathsf{Enc}'(pk, m \in \{0,1\}^\lambda)$
$r \leftarrow \psi_\mathcal{R}$
$\mathbf{return}\ (\mathsf{Enc}(pk, r), \mathsf{F}(r) \oplus m)$

$\mathsf{Dec}'(sk, (c, u))$
$r := \mathsf{Dec}(sk, c)$
$\mathbf{return}\ \mathsf{F}(r) \oplus u$
```

Fig. 5. $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$ transforms PKE with small average-case correctness error, with message space $\mathcal{R}$ and associated distribution $\psi_\mathcal{R}$, into PKE' with small worst-case correctness error. The resulting scheme is λ bits longer.

Lemma 3.1. *If* PKE *is δ-average-case-correct, then* $\mathsf{PKE}' := \mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$ *is δ-worst-case-correct.*

Proof. We need to upper bound $\delta' = \mathbb{E} \max_{m \in \{0,1\}^\lambda} \Pr[\mathsf{Dec}'(\mathsf{Enc}'(m)) \neq m]$, where the expectation is taken over the internal randomness of KeyGen and the choice of random oracle F, and the probability is taken over the internal randomness of Enc'. Since a ciphertext $(\mathsf{Enc}(pk, r), \mathsf{F}(r) \oplus m)$ fails to decrypt iff $\mathsf{Enc}(pk, r)$ fails to decrypt, and since message r is drawn according to the distribution $\psi_\mathcal{R}$ on the message space of PKE,

$$\mathbb{E} \max_{m \in \{0,1\}^\lambda} \Pr[\mathsf{Dec}'(sk, \mathsf{Enc}'(pk, m)) \neq m] = \mathbb{E} \Pr_{r \leftarrow \psi_\mathcal{R}} [\mathsf{Dec}(sk, \mathsf{Enc}(pk, r)) \neq r] = \delta \ .$$

Lemma 3.2. *If* PKE *is weakly γ-spread, then so is* $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$.

Proof. Follows directly by how PKE is used, since the ciphertext of $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$ consists of the ciphertext of PKE, plus the message blinding part.

Theorem 3.3 (q_F-OW-CPA of PKE $\overset{\text{ROM}}{\Longrightarrow}$ IND-CPA of $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$). *For any adversary $\mathcal{A}$ against the* IND-CPA *security of* $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$*, issuing at most q_F queries to* F*, there exists an adversary $\mathcal{B}$ against the* OW-CPA *security of* PKE *with*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ACWC[PKE,F]}}(\mathcal{A}) \leq \mathsf{Adv}^{q_\mathsf{F}\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}) \ ,$$

and the running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

In the IND-CPA game for $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$, the challenge ciphertext $c^* \leftarrow \big(\mathsf{Enc}(pk, r), \mathsf{F}(r) \oplus m_b\big)$ perfectly hides m_b unless the adversary queries F on r, thus breaking OW-CPA security of PKE. A reduction to q_F-OW-CPA security can simply keep a list of all of $\mathcal{A}$ queries to F and return this list as the list of plaintext guesses. For the sake of completeness, a full proof of Theorem 3.3 is given in the full version [16].

Theorem 3.4 (p_F-OW-CPA of PKE $\overset{\text{QROM}}{\Longrightarrow}$ IND-CPA of $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$). *For any quantum adversary $\mathcal{A}$ against the* IND-CPA *security of* $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$*, with query depth at most d_F and query parallelism at most p_F, there exists a quantum adversary $\mathcal{B}$ against the* OW-CPA *security of* PKE *with*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ACWC[PKE,F]}}(\mathcal{A}) \leq 2 d_\mathsf{F} \sqrt{\mathsf{Adv}^{p_\mathsf{F}\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B})}.$$

and the running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

Since the random oracle is now quantum-accessible, we will use the O2H lemma to argue that we can reprogramm F on r, again with the consequence that c^* now perfectly hides b. In accordance with the definition of the O2H extractor, our reduction will pick one of $\mathcal{A}$'s queries at random, measure this query, and return the measured plaintexts as its guess list. Since the query has query parallelism at most p_{F}, the list has at most p_{F} many elements. For the sake of completeness, a full proof of Theorem 3.4 is given in the full version [16].

3.2 Transformation ACWC without redundancy

Let PKE be an encryption scheme with small average-case correctness error, and let F be a random oracle. We will now introduce our second transformation ACWC by describing $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$ in Fig. 6. Again, the resulting scheme has a small worst-case correctness error. Instead of adding redundancy to the ciphertexts, however, the scheme makes use of a generalised One-Time Pad GOTP.

Definition 3.5. *Function* $\mathsf{GOTP} : \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$ *is called generalized one-time pad (for distributions* $\psi_{\mathcal{X}}, \psi_{\mathcal{Y}}, \psi_{\mathcal{U}}$*) if*

1. Decoding: *There exists an efficient inversion algorithm* Inv *such that for all* $x \in \mathcal{X}$, $u \in \mathcal{U}$, $\mathsf{Inv}(\mathsf{GOTP}(x, u), u) = x$.
2. Message-hiding: *For all* $x \in \mathcal{X}$*, the random variable* $\mathsf{GOTP}(x, u)$*, for* $u \leftarrow \psi_{\mathcal{U}}$*, has the same distribution as* $\psi_{\mathcal{Y}}$
3. Randomness-hiding: *For all* $u \in \mathcal{U}$*, the random variable* $\mathsf{GOTP}(x, u)$*, for* $x \leftarrow \psi_{\mathcal{X}}$*, has the same distribution as* $\psi_{\mathcal{Y}}$

A simple example of the generalized one-time pad $\mathsf{GOTP} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ for the uniform distributions is $\mathsf{GOTP}(x, u) := x \oplus u$ with inversion algorithm $\mathsf{Inv}(y, u) := y \oplus u$. The second and third properties are obviously satisfied since the XOR operation is a one-time pad.

Let PKE be a public-key encryption scheme with $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$, where $\psi_{\mathcal{M}} = \psi_{\mathcal{M}_1} \times \psi_{\mathcal{M}_2}$ is a product distribution. Let $\mathsf{GOTP} : \mathcal{M}' \times \mathcal{U} \to \mathcal{M}_2$ be a generalized one-time pad for distribution $\psi_{\mathcal{M}_2}$ and $\mathsf{F} : \mathcal{M}_1 \to \mathcal{U}$ be a random oracle. The associated distributions $\psi_{\mathcal{M}_1}, \psi_{\mathcal{M}_2}, \psi_{\mathcal{M}'}, \psi_{\mathcal{U}}$ do not necessarily have to be uniform. (If $\psi_{\mathcal{U}}$ is not uniform, then the distribution of the random oracle F is such that every output is independently $\psi_{\mathcal{U}}$-distributed.) PKE' obtained by transformation $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$ is described in Fig. 6.

Our first theorem relates the average-case correctness of PKE to the worst-case correctness of $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$.

Lemma 3.6. *Let* PKE *be a public-key encryption scheme with* $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$*, where* $\psi_{\mathcal{M}} = \psi_{\mathcal{M}_1} \times \psi_{\mathcal{M}_2}$ *is a product distribution, and let* $\|\psi_{\mathcal{M}_1}\| := \sqrt{\sum_{M_1} \psi_1(M_1)^2}$*. Let* $\mathsf{GOTP} : \mathcal{M}' \times \mathcal{U} \to \mathcal{M}_2$ *be a generalized one-time pad (for*

```
$\mathsf{Enc}'(pk, m \in \mathcal{M}')$
$M_1 \leftarrow \psi_{\mathcal{M}_1}$
$M_2 := \mathsf{GOTP}(m, \mathsf{F}(M_1))$
return $\mathsf{Enc}(pk, M_1||M_2)$

$\mathsf{Dec}'(sk, c)$
$M_1||M_2 := \mathsf{Dec}(sk, c)$
$m := \mathsf{Inv}(M_2, \mathsf{F}(M_1))$
return $m$
```

Fig. 6. $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$ transforms PKE with small average-case correctness error into PKE' with small worst-case correctness error. The output length of the two schemes is the same.

distributions $\psi_{\mathcal{M}'}, \psi_{\mathcal{U}}, \psi_{\mathcal{M}_2}$) and $\mathsf{F} : \mathcal{M}_1 \to \mathcal{U}$ be a random oracle. If PKE is δ-average-case-correct then $\mathsf{PKE}' := \mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$ is δ' worst-case-correct for

$$\delta' = \delta + \|\psi_{\mathcal{M}_1}\| \cdot \left(1 + \sqrt{(\ln|\mathcal{M}'| - \ln\|\psi_{\mathcal{M}_1}\|)/2}\right).$$ [10]

Proof. For any fixed key pair, $\delta'(pk, sk)$ can be bounded by an arbitrary $t \in \mathbb{R}^+$, plus the probability that $\delta'(pk, sk)$ exceeds t. To bound the latter, we set as t fixed-pair average-case correctness $\delta(pk, sk)$, plus $\|\psi_{\mathcal{M}_1}\| \cdot \sqrt{(c + \ln|\mathcal{M}'|)/2}$, and use helper Lemma 3.7 below. A full proof is given in the full version [16].

Lemma 3.7. *Let g be some function and B be some set such that*

$$\forall m \in \mathcal{M}, \Pr_{r_1 \leftarrow \psi_1, r_2 \leftarrow \psi_2, u \leftarrow U}[g(m, r_1, r_2, u) \in B] \leq \mu, \tag{1}$$

where ψ_1 and ψ_2 are independent. Let F be a random function mapping onto U. Define $\|\psi_1\| = \sqrt{\sum_{r_1} \psi_1(r_1)^2}$. Then for all but an e^{-c} fraction of random functions F, we have that $\forall m \in \mathcal{M}$,

$$\Pr_{r_1 \leftarrow \psi_1, r_2 \leftarrow \psi_2}[g(m, r_1, r_2, \mathsf{F}(r_1)) \in B] \leq \mu + \|\psi_1\| \cdot \sqrt{(c + \ln|\mathcal{M}|)/2} \tag{2}$$

Proof. We show that for any fixed $m \in \mathcal{M}$, the probability in (2) holds for all but a $e^{-c} \cdot |\mathcal{M}|^{-1}$-fraction of random functions F. The claim then follows by the union bound. The full proof is provided in the full version [16].

Lemma 3.8. *If PKE is weakly γ-spread, then so is $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$.*

Proof. Follows directly, since the ciphertext consists of the ciphertext of PKE.

Theorem 3.9 ($(q{\cdot}q_{\mathsf{F}})$-OW-CPA of $\mathsf{PKE} \overset{\text{ROM}}{\Longrightarrow}$ q-OW-CPA of $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$). *Let $q \in \mathbb{N}$. For any adversary $\mathcal{A}$ against the q-OW-CPA security of $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$, making at most q_{F} random oracle queries, there exists an adversary $\mathcal{B}$ against the $(q{\cdot}q_{\mathsf{F}})$-OW-CPA security of $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$ with*

$$\mathsf{Adv}^{q\text{-OW-CPA}}_{\mathsf{ACWC}[\mathsf{PKE},\mathsf{GOTP},\mathsf{F}]}(\mathcal{A}) \leq \mathsf{Adv}^{(q \cdot q_{\mathsf{F}})\text{-OW-CPA}}_{\mathsf{PKE}}(\mathcal{B}) + q \cdot 2^{-H_\infty(\psi_{\mathcal{M}'})},$$

where the running time of $\mathcal{B}$ is about $\mathbf{Time}(\mathcal{A}) + \mathcal{O}(q \cdot q_{\mathsf{F}})$.

[10] In cases where the support of $\psi_{\mathcal{M}_1}$ is some finite set R, it may be sometimes convenient to upper bound $\|\psi_{\mathcal{M}_1}\|$ by $\|\psi_{\mathcal{M}_1}\|_\infty \cdot \sqrt{|R|}$, where $\|\psi_{\mathcal{M}_1}\|_\infty$ is the maximum probability for any element in R.

In the q-$\mathsf{OW\text{-}CPA}$ game for $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$, the adversary is presented with an encryption $c^* \leftarrow \mathsf{Enc}(pk, M_1^* \| \mathsf{GOTP}(m^*, \mathsf{F}(M_1^*)))$ of a message pair $(M_1^*, m^*) \leftarrow \psi_{\mathcal{M}_1} \times \psi_{\mathcal{M}'}$, and has to return a list $\mathcal{Q}$ such that $m^* \in \mathcal{Q}$. Unless $\mathcal{A}$ queries F on M_1^*, m^* is perfectly hidden from $\mathcal{A}$ and $\mathcal{A}$ cannot win with probability better than $q \cdot 2^{-H_\infty(\psi_{\mathcal{M}'})}$. If $\mathcal{A}$ queries F on M_1^* and wins, a reduction can again record $\mathcal{A}$'s oracle queries, and then use the query list $\mathcal{L}_\mathsf{F}$ and $\mathcal{A}$'s one-way guessing list $\mathcal{Q}_\mathcal{A}$ to construct its set $\mathcal{Q}$ by going over all possible combinations $M' = M_1' || M_2'$, where $M_1' \in \mathcal{L}_\mathsf{F}$ and $M_2' := \mathsf{GOTP}(m', \mathsf{F}(M_1'))$ for $m' \in \mathcal{Q}_\mathcal{A}$. If $\mathcal{A}$ queries F on M_1^* and wins, then $\mathcal{L}_\mathsf{F}$ will contain the right M_1^*, meaning that $\mathcal{B}$'s list $\mathcal{Q}$ will contain the challenge plaintext. Note that the ciphertext for $\mathcal{B}$ would be defined relative to $M_2^* \leftarrow \psi_{\mathcal{M}_2}$, but due to the properties of GOTP, $\mathcal{A}$'s one-way game can be conceptually changed such that its ciphertext is also defined relative to $M_2^* \leftarrow \psi_{\mathcal{M}_2}$, and $\mathcal{A}$ wins if it returns a list $\mathcal{Q}$ containing $m := \mathsf{Inv}(M_2^*, \mathsf{F}(M_1^*))$. For the sake of completeness, a full proof of Theorem 3.9 is given in the full version [16].

Theorem 3.10 **($\mathsf{OW\text{-}CPA}$ of $\mathsf{PKE} \overset{\text{QROM}}{\Longrightarrow}$ $\mathsf{OW\text{-}CPA}$ of $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$).** *For any quantum adversary $\mathcal{A}$ against the $\mathsf{OW\text{-}CPA}$ security of $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$, making at most q_F random oracle queries, there exists a quantum adversary $\mathcal{B}$ against the $\mathsf{OW\text{-}CPA}$ security of PKE with*

$$\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{ACWC}[\mathsf{PKE},\mathsf{GOTP},\mathsf{F}]}(\mathcal{A}) \leq (2q_\mathsf{F} + 1)^2 \ \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}),$$

where the running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

Intuitively, the proof follows the same idea as its classical counterpart. In contrast to the security proof for ACWC_0, however, we can not simply apply the O2H lemma, as a reduction needs both a query to F from which it can extract M_1^* *and* its final output m, and an O2H extractor would simply abort $\mathcal{A}$ once that $\mathcal{A}$ has issued the query to be extracted. We will therefore use the measure-and-reprogram technique (M&R) from [11, 13], arguing that we can run the adversary, measure a random query, and continue running it afterwards to obtain its final output m. For the sake of completeness, a full proof of Theorem 3.10 is given in the full version [16].

4 NTRU Encryption over NTT Friendly Rings

In this section we present three instantiations of the NTRU encryption scheme in polynomial rings of the form $\mathbb{Z}_q[X]/(X^d - X^{d/2} + 1)$, where $d = 2^i 3^j$, where the parameters are set such that multiplication and inversion can be performed very efficiently using the NTT.

4.1 Notation

We denote by $\mathcal{R}$ the polynomial ring $\mathbb{Z}_q[X]/(X^d - X^{d/2} + 1)$, where the positive integer d (of the form $2^i 3^j$) and the prime q are implicit from context. Elements in $\mathcal{R}$ will be represented by polynomials of degree less than d, and we will denote these polynomials by bold lower-case letters. That is, all elements of $\mathcal{R}$ are of the form $\mathbf{h} = \sum_{i=0}^{d-1} \mathbf{h}_i X^i \in R$, where $\mathbf{h}_i \in \mathbb{Z}_q$. There is a natural correspondence between elements in $\mathcal{R}$ and vectors in $\mathbb{Z}_q^d$, where one simply writes the coefficients of a polynomial in vector form. As additive groups, the two are trivially isomorphic. We will thus sometimes abuse notation and for a vector $\vec{v}$, write $\mathbf{r} := \vec{v}$ to mean that the coefficients of the polynomial $\mathbf{r}$ are assigned the coefficients of the vector $\vec{v}$.

For an integer $h \in \mathbb{Z}_q$, we write $h \bmod {}^{\pm} q$ to mean the integer from the set $\left\{-\frac{q-1}{2}, \ldots, \frac{q-1}{2}\right\}$ which is congruent to h modulo q. Reducing an integer modulo 2 always maps it to a bit. These functions naturally extend to vectors and polynomials, where one applies the function individually to each coefficient. For a set S, the function $\mathsf{H}_S : \{0,1\}^* \to S$ denotes a hash function modeled as a random oracle that outputs a uniform distribution on S. Similarly, for a distribution ψ (over some implicit set S), we will write $\mathsf{H}_\psi : \{0,1\}^* \to S$ to denote a hash function modeled as a random oracle that outputs a distribution ψ. The function $\mathsf{pref}(\cdot)$ extracts a short (around 32-64 byte) prefix from an element of $\mathcal{R}$.

4.2 The Binomial Distribution

For an even k, we define the distribution ψ_k^d over $\mathbb{Z}^d$ to be the distribution

$$\sum_{i=1}^{k} \vec{a}_i - \sum_{i=1}^{k} \vec{b}_i, \quad \vec{a}_i, \vec{b}_i \leftarrow \{0,1\}^d. \tag{3}$$

The distribution $\bar{\psi}_k^d$ is the distribution over the set $\{-1,0,1\}^d$ defined as ψ_k^d reduced modulo 3. We will mostly be working with $\bar{\psi}_k^d$ and ψ_k^d for $k = 2$, which are, by definition, generated as $\vec{b} = \vec{b}_1 + \vec{b}_2 - \vec{b}_3 - \vec{b}_4$ and $\vec{b} \bmod {}^{\pm} 3$, where $\vec{b}_i \leftarrow \{0,1\}^d$. Each coefficient of $\vec{b}$ and $\vec{b} \bmod {}^{\pm} 3$ is distributed as

$\psi_2 =$ (4)

Output	-2	-1	0	1	2
Prob.	1/16	4/16	6/16	4/16	1/16

$\bar{\psi}_2 =$ (5)

Output	-1	0	1
Prob.	5/16	6/16	5/16

We now state a lemma, which is used for the construction of $\mathsf{NTRU\text{-}A}$ in Figure 7 that shows that by creating the distribution ψ_2 in a special way, one of the components of the distribution can be completely recovered when having access to whole sample. Note that this cannot be done if each coefficient is generated as $b = b_1 + b_2 - b_3 - b_4$. For example, if $b = 0$, then every b_i has conditional probability of 1/2 of being 0 or 1. If, on the other hand, we generate the distribution as $b = (b_1 - 2b_2b_3)(1 - 2b_4)$, where $b_i \leftarrow \{0,1\}$, then one can see that b_1 can be recovered by computing $b \mod 2$.

Lemma 4.1. *The distribution ψ_2^d can be generated as*

$$\vec{b} = (\vec{b}_1 - 2\vec{b}_2 \odot \vec{b}_3) \odot (1 - 2\vec{b}_4),$$

where $\vec{b}_i \leftarrow \{0,1\}^d$ and $\odot$ denotes component-wise multiplication. Furthermore, $\vec{b} \bmod 2 = \vec{b}_1$.

4.3 The NTRU Problem and Variants

In the framework for the NTRU trap-door function [19], the secret key consists of two polynomials $\mathbf{f}$ and $\mathbf{g}$ with small coefficients in a polynomial ring (e.g. $\mathcal{R}$) and the public key if the quotient $\mathbf{h} = \mathbf{g}\mathbf{f}^{-1}$. The hardness assumption states that given $(\mathbf{h}, \mathbf{h}\mathbf{r} + \mathbf{e})$, where $\mathbf{r}, \mathbf{e}$ are sampled from some distribution with support of elements in $\mathcal{R}$ with small coefficients, it is hard to recover $\mathbf{e}$. For appropriately-set parameters, one can recover $\mathbf{e}$ when knowing $\mathbf{f}$, and we will discuss this when presenting the full encryption scheme later in the section. For now, we are mainly interested in the security of NTRU.

The security of the NTRU function described above is naturally broken down into two assumptions. The first is that the distribution of $\mathbf{h} = \mathbf{g}\mathbf{f}^{-1}$ is indistinguishable from a random element in $\mathcal{R}$. And the second assumption is essentially the Ring-LWE assumption which states that given $(\mathbf{h}, \mathbf{h}\mathbf{r} + \mathbf{e})$, where $\mathbf{h}$ is uniform in $\mathcal{R}$ and $\mathbf{r}, \mathbf{e}$ are chosen from some distribution with small coefficients, it is hard to find $\mathbf{e}$ (and thus also $\mathbf{r}$). We point out that one can eliminate the need for the first assumption by choosing polynomials with coefficients that are small, but large enough, so that the quotient is statistically-close to uniform [29], but the resulting scheme ends up being significantly less efficient because the coefficients in the polynomials of the second (Ring-LWE) problem need to be rather small to compensate; and this in turn requires the dimension of the ring to be increased in order for the Ring-LWE problem to remain hard. The below definition formally states the first assumption for the distributions used in this paper.

Definition 4.2 (The $\mathcal{R}$-NTRU_η assumption). *For a distribution η over the ring $\mathcal{R}$ and an integer p relatively-prime to q, the $\mathcal{R}$-NTRU_η assumption states that $\mathbf{g} \cdot (p\mathbf{f} + 1)^{-1}$ is indistinguishable from a uniformly-random element in $\mathcal{R}$ when $\mathbf{g}$ and $\mathbf{f}$ are chosen from the distribution η, and $p\mathbf{f} + 1$ is invertible in $\mathcal{R}$.*

Another common version of the assumption simply states that $\mathbf{g} \cdot \mathbf{f}^{-1}$ is indistinguishable from random, and it doesn't appear that there is any difference in the hardness between the two. The reason that multiplication of $\mathbf{f}$ by p is useful is because it eliminates the need for an inversion (which cannot be done using NTT) during the decryption process; and so we use this version of the problem in the paper. The downside of this multiplication by p is that half of the "noise terms" in the decrypted ciphertext increase by a factor of p. We now define the Ring-LWE problem that is specific to our instantiation, and which forms the second assumption needed for the NTRU cryptosystem.

Definition 4.3 ($\mathcal{R}$-LWE_η). *Let η be some distribution over $\mathcal{R}$. In the $\mathcal{R}$-LWE problem, one is given $(\mathbf{h}, \mathbf{hr}+\mathbf{e})$, where $\mathbf{h} \leftarrow \mathcal{R}$ and $\mathbf{r}, \mathbf{e} \leftarrow \eta$, and is asked to recover $\mathbf{e}$.*

One can also define the decision version of the above assumption as

Definition 4.4 (Decision $\mathcal{R}$-LWE_η). *Let η be a distribution over $\mathcal{R}$. The decision $\mathcal{R}$-LWE assumption states that $(\mathbf{h}, \mathbf{hr}+\mathbf{e})$, where $\mathbf{h} \leftarrow \mathcal{R}$ and $\mathbf{r}, \mathbf{e} \leftarrow \eta$, is indistinguishable from $(\mathbf{h}, \mathbf{u})$, where $\mathbf{h}, \mathbf{u} \leftarrow \mathcal{R}$.*

In the original LWE definition of Regev [28], the distribution η was a rounded continuous Gaussian, as this was the distribution most convenient for achieving a worst-case to average-case reduction from certain lattice problems over solving $\mathcal{R}$-LWE_η. When implementing cryptographic primitives based on the hardness of $\mathcal{R}$-LWE_η, it is more convenient to take η to be a distribution that can be easily sampled. Some common distributions include uniform (although sometimes it is not that simple to sample) and those that can be generated as sums of Bernoulli random variables such as ψ_k and $\bar{\psi}_k$ from (4) and (5).

The most efficient known attack against the $\mathcal{R}$-NTRU and $\mathcal{R}$-LWE problems are lattice attacks. They work by defining a set

$$\mathcal{L}_{\mathbf{c}}^{\perp}(\mathbf{h}) = \{(\mathbf{v}, \mathbf{w}) \in \mathbb{Z}[X]/(X^d - X^{d/2} + 1) \ : \ \mathbf{hv} + \mathbf{w} \equiv \mathbf{c} \pmod{q}\}.$$

When $\mathbf{c} = \mathbf{0}$, the above set is closed under addition, and therefore forms a lattice. To distinguish the quotient $\mathbf{h} = \mathbf{g}/\mathbf{f}$, where $\mathbf{f}, \mathbf{g}$ have small coefficients, from a uniformly-random $\mathbf{h} \in \mathcal{R}$, one can try to find the shortest vector in $\mathcal{L}_{\mathbf{0}}^{\perp}(\mathbf{h})$. If $\mathbf{h}$ is random, then a vector of ℓ_2-norm less than $\Omega(\sqrt{qd})$ is very unlikely to exist in $\mathcal{L}_{\mathbf{c}}^{\perp}(\mathbf{h})$. On the other hand, if the coefficients of $\mathbf{f}, \mathbf{g}$ are noticeably less than $\sqrt{q}$, then $(\mathbf{f}, -\mathbf{g}) \in \mathcal{L}_{\mathbf{c}}^{\perp}(\mathbf{h})$, and so an algorithm that can find a good approximation to the shortest vector should find something of length significantly less than $\Omega(\sqrt{qd})$.

When $\mathbf{c} \neq 0$, $\mathcal{L}_{\mathbf{c}}^{\perp}(\mathbf{h})$ is a *shifted lattice* and finding the shortest vector in it is known as the Bounded Distance Decoding (BDD) problem. For practical parameters, the complexity of the two problems is identical. Interestingly, when q is very large with respect to the size of the secret coefficients, finding a short vector in $\mathcal{L}_{\mathbf{c}}^{\perp}(\mathbf{h})$ is significantly easier when $\mathbf{c} = 0$, as opposed to when $\mathbf{c}$ is random [1, 8, 15, 22]. This phenomenon prevents the NTRU assumption from being used in scenarios requiring such a large gap (and so one uses Ring-LWE and Module-LWE schemes in those scenarios), such as in Fully-Homomorphic Encryption schemes. This security issue, however, does not seem to extend to the NTRU parameters that are used in practice for public key encryption and signature schemes.

We now define a version of the $\mathcal{R}$-LWE problem in which the adversary is not asked to recover the entire vector $\mathbf{e}$, but just $\mathbf{e}$ mod 2.

Definition 4.5 ($\mathcal{R}$-$\mathsf{LWE2}_\eta$). *Let η be a distribution over $\mathcal{R}$. In the $\mathcal{R}$-LWE problem, one is given $(\mathbf{h}, \mathbf{hr}+\mathbf{e})$, where $\mathbf{h} \leftarrow \mathcal{R}$ and $\mathbf{r}, \mathbf{e} \leftarrow \eta$, and is asked to recover $\mathbf{e}$ mod 2.*

While we do not have a formal reduction from $\mathcal{R}$-LWE to $\mathcal{R}$-$\mathsf{LWE2}$, based on the state of the art of how Ring-LWE problems are solved, the two are essentially equivalent. We now present two heuristic arguments for the equivalence of $\mathcal{R}$-LWE and $\mathcal{R}$-$\mathsf{LWE2}$.

Suppose that there is an algorithm that solves $\mathcal{R}$-$\mathsf{LWE2}_\eta$ and we feed it an instance $(\mathbf{h}, \mathbf{hr} + \mathbf{e})$ of $\mathcal{R}$-LWE_η. If the $\mathcal{R}$-$\mathsf{LWE2}_\eta$ solver returns a correct $\mathbf{f} \equiv \mathbf{e} \pmod 2$, then we can create another instance

$$(2^{-1} \cdot \mathbf{h}, 2^{-1}\mathbf{hr} + 2^{-1}(\mathbf{e} - \mathbf{f})) = (\mathbf{h}'\mathbf{r} + \mathbf{e}').$$

Note that $\mathbf{h}'$ is still uniformly random and the distribution of $\mathbf{e}'$ is now "narrower" than that of the original $\mathbf{e}$ – if the coefficients of $\mathbf{e}$ were distributed as ψ_2, then each coefficient of $\mathbf{e}'$ has a probability $3/16$ of being ± 1 and $10/16$ of being 0. Based on the state of the art, a $\mathcal{R}$-LWE-type problem should be easier with this narrower distribution. So one should be able to call the $\mathcal{R}$-$\mathsf{LWE2}_\eta$ oracle again, even though the distribution of $\mathbf{e}'$ is now different. It's easy to see now that this procedure will eventually recover the entire polynomial $\mathbf{e}$.

Another heuristic argument is based on a slightly-modified version of decision $\mathcal{R}$-LWE. In particular, if we assume that the decision $\mathcal{R}$-LWE problem, in which just the first polynomial coefficient in $\mathbb{Z}_q$ is noiseless, then there is a simple reduction from this problem to $\mathcal{R}$-$\mathsf{LWE2}_\eta$. In the reduction, we simply add a noise with distribution η to the first coefficient, and we decide whether the decision $\mathcal{R}$-LWE instance is real or random based on whether or not the answer returned by the $\mathcal{R}$-$\mathsf{LWE2}_\eta$ oracle matches our added error modulo 2. While the version of the decision $\mathcal{R}$-LWE problem where the first integer coefficient has no error is slightly different than usual, the current best-known algorithms would solve the decision problem by solving the search version. And in the search case, the two versions of the problem are equally hard.

The work of Brakerski et al [7] considers this "First-is-Errorless" version of LWE and shows that it is essentially as hard as the usual version. Boudgoust et al. [6] extend this problem to it's Module-LWE variant and showed that an even stronger assumption has a (non-tight) reduction from the usual Module-LWE problem. In short, it is very reasonable to assume that the concrete hardness of the $\mathcal{R}$-$\mathsf{LWE2}_\eta$ problem is the same as that of $\mathcal{R}$-LWE_η.

4.4 NTRU-A: Encryption Based on $\mathcal{R}$-NTRU + $\mathcal{R}$-LWE2 for $\eta = \psi_2^d$

We now give a construction of our first $\mathsf{OW\text{-}CPA}$-secure encryption scheme, $\mathsf{NTRU\text{-}A}$, whose hardness is based on the combination of the $\mathcal{R}$-NTRU_η + $\mathcal{R}$-$\mathsf{LWE2}_\eta$ problems for $\eta = \psi_2$. The way that this scheme differs from the more usual NTRU constructions is that the secret key does let one recover the entire $\mathbf{e}$. This can pose a problem because generally $\mathbf{e}$ the message in the $\mathsf{OW\text{-}CPA}$ NTRU scheme, and yet we can only recover a part of it. This is not a $\mathsf{OW\text{-}CPA}$ scheme and we will not be able to obtain a CCA-secure KEM using generic transformations.

We remedy this issue by only making the value $\mathbf{e} \bmod 2$ be the message. This requires that for a given random message $\mathbf{m}$, the $\mathbf{e}$ is generated from the correct distribution (i.e. ψ_2) with the additional restriction that $\mathbf{m} = \mathbf{e} \bmod 2$. An interesting aspect of this scheme is that because the message is not the entire $\mathbf{e}$, the adversary does not have as much freedom to pick it so as to maximize the decryption error. If the adversary can only pick $\mathbf{e} \bmod 2$, it turns out that the worst-case decryption error is quite close to the "best case". We now proceed to describe the OW-CPA scheme in Figure 7.

```
Gen1()
    $\vec{b}_1, \vec{b}_2, \vec{b}_3, \vec{b}_4 \leftarrow \{0,1\}^d$
    return $\vec{b}_1 + \vec{b}_2 - \vec{b}_3 - \vec{b}_4$

Gen2($\vec{b}_1 \in \{0,1\}^d$)
    $\vec{b}_2, \vec{b}_3, \vec{b}_4 \leftarrow \{0,1\}^d$
    return $(\vec{b}_1 - 2\vec{b}_2 \odot \vec{b}_3) \odot (1 - 2\vec{b}_4)$
```

```
KeyGen()
    $\mathbf{f}' :=$ Gen1()
    $\mathbf{f} := 2\mathbf{f}' + 1$
    if $\mathbf{f}$ is not invertible in $\mathcal{R}$, restart
    $\mathbf{g} :=$ Gen1()
    $(pk, sk) = (2\mathbf{g}\mathbf{f}^{-1}, \mathbf{f})$
    return $(pk, sk)$
```

```
Enc($\mathbf{h} \in \mathcal{R}, \vec{m} \in \{0,1\}^d, \rho \in \{0,1\}^{7d}$)
    Parse $\rho$ as $(\rho_1, \rho_2)$
    $\mathbf{r} :=$ Gen1($;\rho_1$), $\mathbf{e} :=$ Gen2($\vec{m};\rho_2$)
    return $\mathbf{h}\mathbf{r} + \mathbf{e}$

Dec($\mathbf{f} \in \mathcal{R}, \mathbf{c} \in \mathcal{R}$)
    $\mathbf{u} := (\mathbf{c}\mathbf{f} \bmod^{\pm} q) \bmod 2$
    $\vec{m} := \mathbf{u}$
    return $\vec{m}$
```

Fig. 7. OW-CPA Encryption Scheme NTRU-A based on the $\mathcal{R}$-NTRU$_{\psi_2}$ + $\mathcal{R}$-LWE2$_{\psi_2}$ problems. Only the procedures Gen1 and Gen2 are randomized. We include the coins ρ as input for the Encryption algorithm (which will be passed to Gen1 and Gen2) because these are explicitly used in the CCA transformation. The coins used in the key generation are implicit.

OW-CPA Scheme. The distribution of the coefficients of the secret polynomials used in key generation and encryption ψ_2 (see (4)) and is produced by the Gen1() algorithm in Figure 7. As per Lemma 4.1, this distribution can be generated as $b_1 + b_2 - b_3 - b_4$ or, equivalently, as $(b_1 - 2b_2b_3)(1 - 2b_4)$, where all the b_i are Bernoulli random variables. The reason the latter distribution is interesting to us is that modulo 2, it is one of the variables that creates it – b_1.

The secret key is generated by choosing polynomials $\mathbf{f}', \mathbf{g} \leftarrow \psi_2^d$ and computing $\mathbf{f} = 2\mathbf{f}' + 1$. If $\mathbf{f}$ is not invertible in $\mathcal{R}$, we restart. Otherwise, the public key is $\mathbf{h} = 2\mathbf{g}\mathbf{f}^{-1}$ and the secret key is $\mathbf{f}$.

To encrypt a message $\vec{m} \in \{0,1\}^d$, the encryptor first generates a random polynomial $\mathbf{r} \leftarrow \psi_2^d$ using the Gen1() procedure. He then needs to choose a polynomial $\mathbf{e}$ such that $\mathbf{e} \bmod 2$ (as a vector) is $\vec{m}$. Furthermore, when $\vec{m}$ is chosen uniformly at random from $\{0,1\}^d$, the distribution of $\mathbf{e}$ should be ψ_2^d. To create such a distribution, we define $\mathbf{e} =$ Gen2($\vec{m}$). By Lemma 4.1, $\mathbf{e}$ is distributed according to ψ_2^d. The ciphertext is $\mathbf{c} = \mathbf{h}\mathbf{r} + \mathbf{e}$.

To decrypt the ciphertext $\mathbf{c} = \mathbf{h}\mathbf{r} + \mathbf{e} = 2\mathbf{g}\mathbf{r}/\mathbf{f} + \mathbf{e}$, we multiply it by $\mathbf{f}$, centralize it mod q, and then reduce modulo 2 to obtain

$$(\mathbf{c}\mathbf{f} \bmod^{\pm} q) \bmod 2 = 2\mathbf{g}\mathbf{r} + \mathbf{e}\mathbf{f} \bmod 2 = 2\mathbf{g}\mathbf{r} + 2\mathbf{e}\mathbf{f}' + \mathbf{e} \bmod 2 \qquad (6)$$

If all the coefficients of $2\mathbf{gr} + 2\mathbf{ef}' + \mathbf{e}$ (as integers) are smaller than $q/2$, then modulo 2, this value will be exactly $\mathbf{e} \bmod 2$, which is $\vec{m}$. Since the coefficients of $\mathbf{e}$ have absolute value at most 2, in order to have decryption be correct, we need the coefficients of $\mathbf{gr} + \mathbf{ef}'$ to be less than $q/4 - 1$. We will now move on to show how to compute this probability.

Decryption Error for a Worst-Case Message. The decryption error of NTRU-A can be computed following the template given in [26, Section 3.2]. As discussed above, if a coefficient of $\mathbf{gr} + \mathbf{ef}'$ (as an integer) has absolute value less than $q/4 - 1$, then the output of that coefficient in (6) will be $\mathbf{e} \bmod 2$, as desired. So we now need to understand what each coefficient of $\mathbf{gr} + \mathbf{ef}'$ looks like. This is easiest to see via an example of how polynomial multiplication in the ring $\mathcal{R}$ can be represented by a matrix-vector product. If we, for example, want to multiply two polynomials $\mathbf{ab}$ in the ring $\mathbb{Z}_q[X]/(X^6 - X^3 + 1)$, where $\mathbf{a} = \sum\limits_{i=0}^{5} \mathbf{a}_i$ and $\mathbf{b} = \sum\limits_{i=0}^{5} \mathbf{b}_i$ then their product $\mathbf{c} = \sum\limits_{i=0}^{5} \mathbf{c}_i$ can be written as in (7).

$$\begin{bmatrix} \mathbf{a}_0 & -\mathbf{a}_5 & -\mathbf{a}_4 & -\mathbf{a}_3 & -\mathbf{a}_2 - \mathbf{a}_5 & -\mathbf{a}_1 - \mathbf{a}_4 \\ \mathbf{a}_1 & \mathbf{a}_0 & -\mathbf{a}_5 & -\mathbf{a}_4 & -\mathbf{a}_3 & -\mathbf{a}_2 - \mathbf{a}_5 \\ \mathbf{a}_2 & \mathbf{a}_1 & \mathbf{a}_0 & -\mathbf{a}_5 & -\mathbf{a}_4 & -\mathbf{a}_3 \\ \mathbf{a}_3 & \mathbf{a}_2 + \mathbf{a}_5 & \mathbf{a}_1 + \mathbf{a}_4 & \mathbf{a}_0 + \mathbf{a}_3 & \mathbf{a}_2 & \mathbf{a}_1 \\ \mathbf{a}_4 & \mathbf{a}_3 & \mathbf{a}_2 + \mathbf{a}_5 & \mathbf{a}_1 + \mathbf{a}_4 & \mathbf{a}_0 + \mathbf{a}_3 & \mathbf{a}_2 \\ \mathbf{a}_5 & \mathbf{a}_4 & \mathbf{a}_3 & \mathbf{a}_2 + \mathbf{a}_5 & \mathbf{a}_1 + \mathbf{a}_4 & \mathbf{a}_0 + \mathbf{a}_3 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{b}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \mathbf{c}_4 \\ \mathbf{c}_5 \end{bmatrix} \tag{7}$$

Notice that $\mathbf{c}_3, \mathbf{c}_4$, and $\mathbf{c}_5$ are a sum of three independently-generated integers of the form

$$c = ba + b'(a + a'). \tag{8}$$

The coefficient $\mathbf{c}_2$, however, is simply a sum of 6 independent random variables of the form ab. Or to make it look similar to (8), we can think of it as the sum of three random variables of the form

$$c = ba + b'a'. \tag{9}$$

It should be clear that the distribution of (8) is wider than that of (9), and so the probability that the coefficients which follow the former distribution will be outside of the "safe zone" is larger. The coefficients $\mathbf{c}_0$ and $\mathbf{c}_1$ are a hybrid of these two distributions. For example, $\mathbf{c}_1$ is the sum of one coefficient from (8) and two from (9); while $\mathbf{c}_2$ is the sum of two from (8) and one from (9).

To bound the probability that decryption will be correct, we should therefore bound the distribution of $\mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5$, or in the general case, a coefficient in the bottom half of $\mathbf{c}$ and then apply the union bound. So the widest distribution will consist of sums of $d/2$ random variables having the distribution as in (8). The term $\mathbf{gr}$ in (6) has this exact distribution, where each coefficient of $\mathbf{g}, \mathbf{r}$ is distributed according to Gen1().

The term $\mathbf{f}'\mathbf{e}$ is distributed differently because in our security proof we need to consider an adversarially-chosen message $\vec{m}$, after the adversary sees the public key. Because the adversary does not get to choose the whole message, but just the modulo 2 residue, it turns out that the failure probability for a worst-case message is not too different than for a uniformly random one. In (10), we give the distribution of a particular coefficient of $\mathbf{e}_i$ conditioned on the message bit being either 0 or 1.

$$\mathsf{Gen2}(0) = \begin{array}{|c||c|c|c|}\hline \text{Output} & \text{-2} & 0 & 2 \\ \hline \text{Probability} & 0.125 & 0.75 & 0.125 \\ \hline \end{array} \quad \mathsf{Gen2}(1) = \begin{array}{|c||c|c|}\hline \text{Output} & \text{-1} & 1 \\ \hline \text{Probability} & 0.5 & 0.5 \\ \hline \end{array} \tag{10}$$

One can see that in both cases the distribution is centered around 0 and has variance 1, and so one should not expect a very large difference in the decryption error. Experimentally, it turns out that the worst-case messages occur when choosing $\vec{m} = \vec{0}$. Furthermore, the worst-case message is the same for any secret key.[11] This implies that the worst-case correctness error is the average-case one where the distribution over the coefficients of $\mathbf{e}$ is as in $\mathsf{Gen2}(0)$ of (10). As in [3, 5, 26], the error probability reported in Table 1 is computed via polynomial multiplications which represent convolutions of random variables.

IND-CCA-secure KEM. One can apply the Fujisaki-Okamoto transformation $\mathsf{FO}^{\perp}$ from Fig. 4 to obtain the IND-CCA secure version $\mathsf{CCA\text{-}NTRU\text{-}A} := \mathsf{FO}^{\perp}[\mathsf{NTRU\text{-}A}, \mathsf{H}]$ of $\mathsf{NTRU\text{-}A}$. The concrete security bounds on the $\mathsf{IND\text{-}CCA}$ security of $\mathsf{CCA\text{-}NTRU\text{-}A}$ from Table 4 can be derived in the ROM using Lemma 2.1 and Theorem 2.3 and in the QROM using Theorem 2.5.

IND-CCA secure KEM	ROM	QROM
CCA-NTRU-A	$q(\varepsilon_A + \delta)$	$q\sqrt{\varepsilon_A} + q^2\sqrt{\delta}$
CCA-NTRU-B	$\varepsilon_B + q(3^{-\lambda} + \delta)$	$q^2(\sqrt{\varepsilon_B} + \sqrt{\delta})$
CCA-NTRU-C	$\varepsilon_C + q(2^{-\lambda} + \delta)$	$q^{1.5}(\sqrt[4]{\varepsilon_C} + \sqrt{\delta})$

Table 4. Bounds on the $\mathsf{IND\text{-}CCA}$ secure NTRU-variants $\mathsf{CCA\text{-}NTRU\text{-}A}$, $\mathsf{CCA\text{-}NTRU\text{-}B}$, and $\mathsf{CCA\text{-}NTRU\text{-}C}$. Constants and negligible terms are suppressed for simplicity. The value q is the sum of all adversarial (random oracle and decryption) queries, i.e., $q = q_{\mathsf{H}} + q_{\mathsf{D}} + q_{\mathsf{F}}$. The ε values are the advantage functions of the underlying NTRU assumptions: $\varepsilon_A = \mathsf{Adv}^{\mathcal{R}\text{-}\mathsf{NTRU}_\eta} + \mathsf{Adv}^{\mathcal{R}\text{-}\mathsf{LWE2}_\eta}$ for $\eta = \psi_2^d$; $\varepsilon_B = \mathsf{Adv}^{\mathcal{R}\text{-}\mathsf{NTRU}_\eta} + \mathsf{Adv}^{\mathcal{R}\text{-}\mathsf{LWE}_\eta}$ for $\eta = U_3^d$ and $\varepsilon_C = \mathsf{Adv}^{\mathcal{R}\text{-}\mathsf{NTRU}_\eta} + \mathsf{Adv}^{\mathcal{R}\text{-}\mathsf{LWE}_\eta}$ for $\eta = \bar{\psi}_2^d$.

4.5 Generic NTRU encryption and Error-Reducing Transformations

Fig. 8 defines $\mathsf{GenNTRU}[\eta]$ relative to distribution η over $\mathcal{R}$. Note that $\mathsf{GenNTRU}[\eta]$ is randomness-recoverable (RR) because once we have $\mathbf{e}$ and $\mathbf{c} = \mathbf{hr} + \mathbf{e}$, we can

[11] This was verified experimentally by fixing the a, a' in (8) to all valid values and computing the probability of failure assuming that all the secret keys have this value.

compute $\mathbf{r} = (\mathbf{c} - \mathbf{e}) \cdot \mathbf{h}^{-1}$. Because we checked that $\mathbf{g}$ is invertible, it holds that $\mathbf{h} = 3\mathbf{g}\mathbf{f}^{-1}$ also has an inverse.

```
KeyGen()
    $\mathbf{f}', \mathbf{g} \leftarrow \eta$
    $\mathbf{f} := 3\mathbf{f}' + 1$
    if $\mathbf{f}$ or $\mathbf{g}$ is not invertible in $\mathcal{R}$, restart
    return $(pk, sk) = (3\mathbf{g}\mathbf{f}^{-1}, \mathbf{f})$

Enc($\mathbf{h} \in \mathcal{R}, \vec{m} \in \{-1,0,1\}^d$)
    $\mathbf{r} \leftarrow \eta$
    return $\mathbf{c} := \mathbf{h}\mathbf{r} + \vec{m}$

Dec($\mathbf{f} \in \mathcal{R}, \mathbf{c} \in \mathcal{R}$)
    return $\vec{m} := (\mathbf{c}\mathbf{f} \bmod {}^{\pm}q) \bmod {}^{\pm}3$
```

Fig. 8. Generic NTRU $\mathsf{GenNTRU}[\eta]$ relative to distribution ψ over ring $\mathcal{R}$ with average-case correctness error. During key-generation, we need to check that $\mathbf{g}$ is invertible in order to have the randomness recovery property. It seems doubtful that this check adds any actual security in practice, but for all parameter sets, it only adds less than 0.01% chance to a restart, so it does not make much difference either way.

By the definition, the $\mathsf{OW\text{-}CPA}$ security of $\mathsf{GenNTRU}[\eta]$ is implied by the $\mathcal{R}\text{-}\mathsf{NTRU}_\eta{+}\mathcal{R}\text{-}\mathsf{LWE}_\eta$ assumptions. In this subsection, we will consider two concrete instantiations of $\mathsf{GenNTRU}$, namely $\mathsf{GenNTRU}[U_3]$, where U_3 is the uniform distribution over $\{-1,0,1\}^d$, and $\mathsf{GenNTRU}[\bar{\psi}_2^d]$, where $\bar{\psi}_2^d$ was defined in Section 4.2. Both schemes do not have sufficiently small worst-case correctness error, which is the reason why we will first apply one of our average-case to worst-case correctness error transformations from the last section.

$\mathsf{NTRU\text{-}B}$: Encryption Based on $\mathcal{R}\text{-}\mathsf{NTRU}_\eta{+}\mathcal{R}\text{-}\mathsf{LWE}_\eta$ for $\eta = U_3^d$. We define the generalized one-time pad $\mathsf{GOTP} : \mathcal{R} \times \mathcal{R} \to \mathcal{R}$ relative to distributions U_3^d as $\mathsf{GOTP}(\vec{m}, u) := \vec{m} + u \bmod {}^{\pm}3$. Then $\mathsf{NTRU\text{-}B} := \mathsf{ACWC}[\mathsf{GenNTRU}[U_3^d], \mathsf{GOTP}, \mathsf{F}]$, obtained by applying the ACWC transformation from Section 3.2 to $\mathsf{GenNTRU}[U_3^d]$, is described in Fig. 9. Its message space is $\mathcal{M}' = \{-1,0,1\}^\lambda$ with distribution U_3^d, where $\mathcal{M}_1 = \{-1,0,1\}^{d-\lambda}$ and $\mathcal{M}_2 = \{-1,0,1\}^\lambda$.

By Lemma 3.6, the average-case correctness error of $\mathsf{GenNTRU}[U_3^d]$ and the worst-case correctness error of $\mathsf{NTRU\text{-}B}$ are off by an additive factor of

$$\Delta = \|U_3^{d-\lambda}\| \cdot \left(1 + \sqrt{(\ln|\mathcal{M}'| - \ln\|U_3^{d-\lambda}\|)/2}\right) \approx \|U_3^{d-\lambda}\| = 3^{-(d-\lambda)/2} \approx 2^{-0.8\times(d-\lambda)}$$

which can be neglected for $\lambda = 256$ and $d \geq 576$. Hence, for all practical parameters considered in Table 1, worst-case and average-case correctness errors are equal. Using the techniques Section 4.4 it can be verified that the error probabilities reported in Table 1 are correct for $\mathsf{NTRU\text{-}B}$.

Finally, one can apply the Fujisaki-Okamoto transformation $\mathsf{FO}^\perp$ from Fig. 4 to obtain the IND-CCA secure version $\mathsf{CCA\text{-}NTRU\text{-}B} := \mathsf{FO}^\perp[\mathsf{NTRU\text{-}B}, \mathsf{H}]$ of $\mathsf{NTRU\text{-}B}$. In the ROM, the concrete security bound on the $\mathsf{IND\text{-}CCA}$ security of $\mathsf{CCA\text{-}NTRU\text{-}B}$ from Table 4 can be derived by combining Lemma 2.2 with

```
KeyGen()
    $\mathbf{f}', \mathbf{g} \leftarrow \{-1,0,1\}^d$
    $\mathbf{f} := 3\mathbf{f}' + 1$
    if $\mathbf{f}$ or $\mathbf{g}$ is not invertible
        in $\mathcal{R}$, restart
    $(pk, sk) = (3\mathbf{g}\mathbf{f}^{-1}, \mathbf{f})$
    return $(pk, sk)$

Enc($\mathbf{h} \in \mathcal{R}, \vec{m} \in \{-1,0,1\}^\lambda, \rho$)
    (use the randomness $\rho$ for
        creating $\vec{m}'$ and $\mathbf{r}$)
    $\vec{m}' \leftarrow \{-1,0,1\}^{d-\lambda}$
    $\vec{u} := \mathsf{F}_{\{-1,0,1\}^\lambda}(\vec{m}')$
    $\vec{m}'' := \vec{m} + u \bmod^{\pm} 3$
    $\mathbf{r} \leftarrow \{-1,0,1\}^d$
    $\mathbf{e} := \vec{m}' || \vec{m}''$
    return $\mathbf{h}\mathbf{r} + \mathbf{e}$

Dec($\mathbf{f} \in \mathcal{R}, \mathbf{c} \in \mathcal{R}$)
    $\vec{m}' || \vec{m}'' := (\mathbf{c}\mathbf{f} \bmod^{\pm} q)$
        $\bmod^{\pm} 3$
    $u := \mathsf{F}_{\{-1,0,1\}^\lambda}(\vec{m}')$
    $\vec{m} := \vec{m}'' - \vec{u} \bmod^{\pm} 3$
    return $\vec{m}$
```

Fig. 9. Randomness-recoverable OW-CPA encryption scheme NTRU-B with worst-case correctness error based on the $\mathcal{R}$-$\mathsf{NTRU}_{U_3^d}$ + $\mathcal{R}$-$\mathsf{LWE}_{U_3^d}$ problems for U_3^d being uniform over $\{-1,0,1\}^d$.

Theorems 3.9 and 2.3. We refer to Fig. 1 for an overview of the implications. In the QROM, the bound can be derived by combining Theorem 3.10 with Theorem 2.5.

NTRU-C: Encryption Based on $\mathcal{R}$-NTRU$_\eta$+$\mathcal{R}$-LWE$_\eta$ for $\eta = \bar{\psi}_2^d$. We define NTRU-C := $\mathsf{ACWC}_0[\mathsf{GenNTRU}[\bar{\psi}_2^d], \mathsf{F}]$ with uniform message space $\mathcal{M}' = \{0,1\}^\lambda$, obtained by applying the ACWC_0 transformation with redundancy from Section 3.1 to $\mathsf{GenNTRU}[\bar{\psi}_2^d]$ is described in Fig. 10. By Lemma 3.1, the average-case correctness error of $\mathsf{GenNTRU}[\bar{\psi}_2^d]$ and the worst-case correctness error of NTRU-C are identical. Using the techniques Section 4.4 it can be verified that the error probabilities reported in Table 1 are correct for NTRU-C. Finally, one can apply the Fujisaki-Okamoto transformation $\mathsf{FO}^\perp$ from Fig. 4 to obtain the IND-CCA secure version CCA-NTRU-C := $\mathsf{FO}^\perp[\mathsf{NTRU\text{-}C}, \mathsf{H}]$ of NTRU-C. In the ROM, the concrete security bound on the IND-CCA security of CCA-NTRU-C from Table 4 can be derived by combining Lemma 2.2 with Theorems 3.3 and 2.4. In the QROM, the bound can be derived by combining Lemma 2.2 with Theorem 3.4 and Theorem 2.5.

```
KeyGen()
    $\mathbf{f}', \mathbf{g} \leftarrow \bar{\psi}_2^d$
    $\mathbf{f} := 3\mathbf{f}' + 1$
    if $\mathbf{f}$ or $\mathbf{g}$ is not invertible
        in $\mathcal{R}$, restart
    return $(pk, sk) = (3\mathbf{g}\mathbf{f}^{-1}, \mathbf{f})$

Enc($\mathbf{h} \in \mathcal{R}, \vec{m} \in \{0,1\}^\lambda, \rho \in \{0,1\}^{8d}$)
    (use the randomness $\rho$ for
        creating $\mathbf{e}$ and $\mathbf{r}$)
    $\mathbf{e}, \mathbf{r} \leftarrow \bar{\psi}_2^d$
    $\vec{u} := \vec{m} \oplus \mathsf{F}_{\{0,1\}^\lambda}(\mathbf{e})$
    return $(\mathbf{h}\mathbf{r} + \mathbf{e}, \vec{u})$

Dec($\mathbf{f} \in \mathcal{R}, (\mathbf{c} \in \mathcal{R}, \vec{u} \in \{0,1\}^\lambda)$)
    $\mathbf{e} := (\mathbf{c}\mathbf{f} \bmod^{\pm} q) \bmod^{\pm} 3$
    $\vec{m} := \vec{u} \oplus \mathsf{F}_{\{0,1\}^\lambda}(\mathbf{e})$
    return $\vec{m}$
```

Fig. 10. NTRU-C: a randomness-recoverable OW-CPA encryption scheme with worst-case correctness error based on the $\mathcal{R}$-NTRU_η + $\mathcal{R}$-LWE_η problems for $\eta = \bar{\psi}_2^d$.

Acknowledgements

The work of Julien Duman was supported by the German Federal Ministry of Education and Research (BMBF) in the course of the 6GEM Research Hub under Grant 16KISK037. Eike Kiltz was supported by the Deutsche Forschungsgemeinschaft (DFG, German research Foundation) as part of the Excellence Strategy of the German Federal and State Governments – EXC 2092 CASA - 390781972, and by the European Union (ERC AdG REWORC - 101054911). Dominique Unruh was supported by the ERC consolidator grant CerQuS (819317), by the Estonian Centre of Exellence in IT (EXCITE) funded by ERDF, by PUT team grant PRG946 from the Estonian Research Council. Vadim Lyubashevsky and Gregor Seiler were supported by the ERC Consolidator grant PLAZA (101002845).

References

[1] M. R. Albrecht, S. Bai, and L. Ducas. "A Subfield Lattice Attack on Overstretched NTRU Assumptions - Cryptanalysis of Some FHE and Graded Encoding Schemes". In: *CRYPTO (1)*. Vol. 9814. Springer, 2016, pp. 153–178.

[2] M. R. Albrecht, R. Player, and S. Scott. "On the concrete hardness of Learning with Errors". In: *J. Math. Cryptol.* 9.3 (2015), pp. 169–203.

[3] E. Alkim et al. "Post-quantum Key Exchange - A New Hope". In: *USENIX Security Symposium*. USENIX Association, 2016, pp. 327–343.

[4] A. Ambainis, M. Hamburg, and D. Unruh. "Quantum Security Proofs Using Semi-classical Oracles". In: *Advances in Cryptology – CRYPTO 2019*. Vol. 11693. Springer, 2019, pp. 269–295.

[5] J. W. Bos et al. "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM". In: *EuroS&P*. IEEE, 2018, pp. 353–367.

[6] K. Boudgoust et al. "On the Hardness of Module-LWE with Binary Secret". In: *CT-RSA*. Vol. 12704. Springer, 2021, pp. 503–526.

[7] Z. Brakerski et al. "Classical hardness of learning with errors". In: *STOC*. 2013, pp. 575–584.

[8] J. H. Cheon, J. Jeong, and C. Lee. "An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low-level encoding of zero". In: *LMS Journal of Computation and Mathematics* 19.A (2016), 255–266.

[9] C. M. Chung et al. "NTT Multiplication for NTT-unfriendly Rings New Speed Records for Saber and NTRU on Cortex-M4 and AVX2". In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2021.2 (2021), pp. 159–188.

[10] J. D'Anvers et al. "Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM". In: *AFRICACRYPT*. 2018, pp. 282–305.

[11] J. Don, S. Fehr, and C. Majenz. "The Measure-and-Reprogram Technique 2.0: Multi-round Fiat-Shamir and More". In: *CRYPTO (3)*. Ed. by D. Micciancio and T. Ristenpart. Vol. 12172. Springer, 2020, pp. 602–631.

[12] J. Don et al. "Online-extractability in the quantum random-oracle model". In: *EUROCRYPT (3)*. Springer. 2022, pp. 677–706.

[13] J. Don et al. "Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model". In: *CRYPTO 2019*. Ed. by A. Boldyreva and D. Micciancio. Vol. 11693. Springer, 2019, pp. 356–383.

[14] L. Ducas. "Shortest Vector from Lattice Sieving: A Few Dimensions for Free". In: *EUROCRYPT (1)*. Vol. 10820. Springer, 2018, pp. 125–145.

[15] L. Ducas and W. van Woerden. "NTRU fatigue: how stretched is overstretched?" In: *ASIACRYPT (4)*. Springer. 2021, pp. 3–32.

[16] J. Duman et al. "A thorough treatment of highly-efficient NTRU instantiations". In: *Cryptology ePrint Archive* (2021).

[17] J. Duman et al. "Faster Lattice-Based KEMs via a Generic Fujisaki-Okamoto Transform Using Prefix Hashing". In: *CCS*. 2021.

[18] C. Gentry. "Key Recovery and Message Attacks on NTRU-Composite". In: *EUROCRYPT*. Vol. 2045. Springer, 2001, pp. 182–194.

[19] J. Hoffstein, J. Pipher, and J. H. Silverman. "NTRU: A Ring-Based Public Key Cryptosystem". In: *ANTS*. 1998, pp. 267–288.

[20] D. Hofheinz, K. Hövelmanns, and E. Kiltz. "A Modular Analysis of the Fujisaki-Okamoto Transformation". In: *TCC*. 2017, pp. 341–371.

[21] A. Hülsing et al. "High-Speed Key Encapsulation from NTRU". In: *CHES*. Vol. 10529. Springer, 2017, pp. 232–252.

[22] P. Kirchner and P. Fouque. "Revisiting Lattice Attacks on Overstretched NTRU Parameters". In: *EUROCRYPT (1)*. Vol. 10210. 2017, pp. 3–26.

[23] A. Langlois and D. Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptography* 75.3 (2015), pp. 565–599.

[24] V. Lyubashevsky and D. Micciancio. "Generalized Compact Knapsacks Are Collision Resistant". In: *ICALP (2)*. 2006, pp. 144–155.

[25] V. Lyubashevsky, C. Peikert, and O. Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *EUROCRYPT*. 2010, pp. 1–23.

[26] V. Lyubashevsky and G. Seiler. "NTTRU: Truly Fast NTRU Using NTT". In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2019.3 (2019), pp. 180–201.

[27] V. Lyubashevsky and G. Seiler. "Short, Invertible Elements in Partially Splitting Cyclotomic Rings and Applications to Lattice-Based Zero-Knowledge Proofs". In: *EUROCRYPT (1)*. Vol. 10820. Springer, 2018, pp. 204–224.

[28] O. Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *J. ACM* 56.6 (2009).

[29] D. Stehlé and R. Steinfeld. "Making NTRU as Secure as Worst-Case Problems over Ideal Lattices". In: *EUROCRYPT*. 2011, pp. 27–47.

[30] D. Stehlé et al. "Efficient Public Key Encryption Based on Ideal Lattices". In: *ASIACRYPT*. 2009, pp. 617–635.

[31] D. Unruh. "Revocable quantum timed-release encryption". In: *J ACM* 62.6 (2015), 49:1–49:76.