# A Thorough Treatment of Highly-Efficient NTRU Instantiations

Julien Duman[1], Kathrin Hövelmanns[2], Eike Kiltz[1], Vadim Lyubashevsky[3], Gregor Seiler[3,4], and Dominique Unruh[5]

[1] Ruhr-Universität Bochum
[2] TU Eindhoven
[3] IBM Research Europe, Zurich
[4] ETH, Zurich
[5] 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. The first encryption scheme utilizing properties of polynomial rings was NTRU (ANTS '98), but in the recent decade, most research has focused on constructing schemes based on the hardness of the somewhat related Ring/Module-LWE problem. Indeed, 14 out of the 17 encryption schemes based on the hardness of lattice problems in polynomial rings submitted to the first round of the NIST standardization process used some version of Ring/Module-LWE, with the other three being based on NTRU.

The preference for using Ring/Module-LWE is due to the fact that this problem is at least as hard as NTRU, is more flexible in the algebraic structure due to the fact that no polynomial division is necessary, and that the decryption error is independent of the message. 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 detach the decryption error from the message, thus eliminating the adversary's power to have any effect on it, which ultimately allows us to decrease parameter sizes. The resulting schemes 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 [HPS98] 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 the 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 NTRU encryption 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 [SSTX09, LPR10] 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 $\mathbf{g}$ [SS11], the scheme becomes rather 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 [Reg09], and its subsequent porting to lattices over polynomial rings [SSTX09, LPR10, LS15], 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 in [LPR10], 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 as the modulus $q$ of the ring becomes larger than the dimension [ABD16, CJL16, KF17, DvW21] (the Ring-LWE problem is still believed to be hard for these parameters). 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 [BDK+18, DKRV18]. These schemes are also significantly more efficient than the finalist NTRU-based proposal [HRSS17].[6]

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 an advantage when used in certain scenarios involving zero-knowledge proofs, since 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

[6] The schemes [BDK+18, DKRV18] can be made even more efficient by eliminating an unnecessary input to the random oracle (see [DHK+21]) which did not exist in [HRSS17].

approximately $2d \log d$ operations over $\mathbb{Z}_q$. These rings also have some very nice algebraic properties – in particular the expansion factor [LM06] 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 use 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 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 by using "NTT-friendly" rings $f(X) = X^d - X^{d/2} + 1$ where $d = 2^i 3^j$.[7] 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 $\zeta$ 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 [LS19], a version of NTRU 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 [CHK+21], 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.

### 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 [HPS98]) 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

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

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 [HHK17]. 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 $\mathsf{ACWC}_0$ 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 13. 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 $\mathsf{ACWC}$ (Figure 6 on page 14) 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 [HRSS17] 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 a very interesting open problem as to whether our transformation can still be proved secure for a different, more easily sampled, distribution of the message (and with the resulting scheme being based on the same assumption).

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 $\mu$ and an error part $\epsilon$. One has this implicit split by defining a function $f(\mu, \epsilon) = \mathbf{m}$ in a particular way where $\epsilon$ and $\mu$ 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 $\mu$. If we want to choose $\mathbf{m}$ according to the binomial distribution (as in e.g. NewHope [ADPS16], Kyber [BDK+18], or Saber [DKRV18]), 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 $\mu$. 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 set the parameter $p$ 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 $\mathsf{ACWC}_0$ and $\mathsf{ACWC}$ are defined relative to random oracles, and have proofs in the ROM that are conceptually very simple. We show that $\mathsf{ACWC}_0$ transforms any one-way secure (OW-CPA) encryption scheme into one that is IND-CPA secure, and that $\mathsf{ACWC}$ transforms any OW-CPA secure encryption scheme into one that is also OW-CPA secure. Note that we cannot prove IND-CPA security of $\mathsf{ACWC}$ since there exist instantiations for which application of $\mathsf{ACWC}$ yields a scheme that simply isn't IND-CPA secure.[8] By working with $q$-OW-CPA security,[9] a slight generalisation of OW-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 $\mathsf{ACWC}_0$ and $\mathsf{ACWC}$. We show that $\mathsf{ACWC}_0$ transforms a one-way secure (OW-CPA) encryption scheme into an IND-CPA secure one. This proof is a reasonably straightforward application of the one-way to hiding theorem, O2H [Unr15] in the variant from [AHU19]. (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 $\varepsilon$ 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 $\mathsf{ACWC}$ does not have an obvious proof using O2H. Instead, we use the measure-and-reprogram technique (M&R) from [DFMS19, DFM20]. This technique was developed for proving the security of the Fiat-Shamir transform; it is a quantum alternative to the classical forking-lemma. 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 $\mathsf{ACWC}$ 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 $\Theta$

[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, $\mathsf{ACWC}$ 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 $q$-OW-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 OW-CPA security.

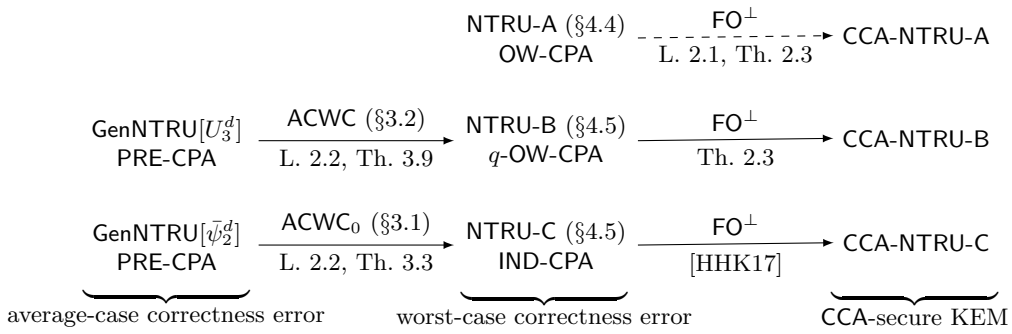


**Fig. 1.** Overview: How to obtain efficient IND-CCA-secure KEMs from our NTRU-based PKE schemes. Solid arrows indicate tight reductions in the ROM, dashed arrows indicate non-tight reductions. $q$-OW-CPA is a strengthening of standard OW-CPA security, where the adversary is allowed to return $q$ many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

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 $\Theta$. (We use a random value $\Theta$ 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.

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 "neutral" evaluation of security, we used the online LWE hardness estimator [APS15]. 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. [Duc18]) 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 is currently a finalist in the NIST standardization, $\mathsf{NTRU\text{-}C}^{648}_{2917}$ 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 $\mathsf{NTRU\text{-}C}^{768}_{3457}$, which has a similar public key/ciphertext size as NTRU-HRSS-701, one see 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 NTRU-A version. In our opinion, exchanging such a large security margin in return for, what is essentially a gimmick of reducing $2^{-250}$ to 0 in the information-theoretic decryption error value, 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 $\mathsf{NTRU\text{-}A}^{648}_{2917}$, 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 NTRU-A being over 15X faster for round-trip ephemeral key exchange. The running time of NTRU-C should be quite similar, and NTRU-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 NTRU-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.

## 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 Appendix A.

### 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

| $d$ (dim.) | $q$ (mod.) | inertia degree | $pk$ & ct (B) [a] | $\log_2(\delta)$ | security | $\log_2(\delta)$ | security | $\log_2(\delta)$ | security |
|---|---|---|---|---|---|---|---|---|---|
| | | | | 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 $\delta$ 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 [APS15].

| | 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. The NTTRU parameters are from [LS19], and the NTRU-HRSS-701 parameters are from [HRSS17], 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 [APS15].

| Scheme | Key Gen | Encaps | Decaps | Total Round-Trip |
|---|---|---|---|---|
| NTRU-$A^{648}_{2917}$ (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 | 6.2K | 7.9K | 9.2K | 23.3K |
| Kyber-768 | 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 [DHK+21, Table 3], which shows an improved implementation of Kyber90's when using prefix hashing and employing an explicit reject in the decapsulation procedure.

$\mathsf{KeyGen}$ outputs a key pair $(pk, sk)$, where $pk$ also defines a finite randomness space $\mathcal{R} = \mathcal{R}(pk)$. The encryption algorithm $\mathsf{Enc}$, 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 $\mathsf{Enc}$, which we require to be independent of $m$. The decryption algorithm $\mathsf{Dec}$, 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. $\mathsf{PKE}$ 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. $\mathsf{PKE}$ has (worst-case) correctness error $\delta$ [HHK17] 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). $\mathsf{PKE}$ has average-case correctness error $\delta$ 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 $\mathsf{Enc}$. 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. $\mathsf{PKE}$ is weakly $\gamma$-spread [DFMS21] 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 $\mathsf{PKE}$, 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

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

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 $\mathsf{PKE}$ by the defining the advantage function of an adversary $\mathcal{A}$ as

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

| Game $q$-OW-CPA | Game PRE-CPA | Game IND-CPA |
|---|---|---|
| $(pk, sk) \leftarrow \mathsf{KeyGen}$ | $(pk, sk) \leftarrow \mathsf{KeyGen}$ | $(pk, sk) \leftarrow \mathsf{Gen}$ |
| $m^* \leftarrow \psi_{\mathcal{M}}$ | $m^* \leftarrow \psi_{\mathcal{M}}$ | $(m_0, m_1) \leftarrow \mathcal{A}_1(pk)$ |
| $c^* \leftarrow \mathsf{Enc}(pk, m^*)$ | $c^* \leftarrow \mathsf{Enc}(pk, m^*)$ | $b \leftarrow \{0, 1\}$ |
| $\mathcal{Q} \leftarrow \mathcal{A}(pk, c^*)$ | $(m, r) \leftarrow \mathcal{A}(pk, c^*)$ | $c^* \leftarrow \mathsf{Enc}(pk, m_b)$ |
| **return** $[\![m^* \in \mathcal{Q} \wedge \lvert\mathcal{Q}\rvert \leq q]\!]$ | **return** $[\![\mathsf{Enc}(pk, m; r) = c^*]\!]$ | $b' \leftarrow \mathcal{A}_2(pk, c^*)$ |
| | | **return** $[\![b = b']\!]$ |

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

where game PRE-CPA is given in Fig. 2.

Finally, we define the IND-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 OW-CPA $\implies$ PKE $q$-OW-CPA).** *For any adversary $\mathcal{A}$ against the $q$-OW-CPA security of PKE, there exists an OW-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 OW-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 OW-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 PRE-CPA and RR $\overset{\text{tightly}}{\Longrightarrow}$ PKE $q$-OW-CPA).** *If PKE is randomness recoverable, then for any adversary $\mathcal{A}$ against the $q$-OW-CPA security of PKE, there exists an PRE-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 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 $\mathsf{Encaps}$ 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 $\mathsf{Decaps}$, 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 $\mathsf{KEM}$ has correctness error $\delta$ 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 $\mathsf{Encaps}$ and $(pk, sk) \leftarrow \mathsf{Gen}$. In terms of $\mathsf{KEM}$'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 | Decaps($c \neq c^*$) |
|---|---|
| 01 $(pk, sk) \leftarrow \mathsf{Gen}$ | 07 **return** $\mathsf{Decaps}(sk, c)$ |
| 02 $(K_0, c^*) \leftarrow \mathsf{Encaps}(pk)$ | |
| 03 $K_1 \leftarrow \mathcal{K}$ | |
| 04 $b \leftarrow \{0, 1\}$ | |
| 05 $b' \leftarrow \mathcal{A}^{\mathsf{Decaps}}(pk, c^*, K_b)$ | |
| 06 **return** $[\![b = b']\!]$ | |

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

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 [DHK+21] 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 ($q_{\mathsf{H}}$-OW-CPA of PKE $\overset{\mathsf{ROM}}{\Longrightarrow}$ IND-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* $\mathsf{KEM}$*, there exists an adversary $\mathcal{B}$ against the $q_H$-*$\mathsf{OW\text{-}CPA}$ *security of* $\mathsf{PKE}$ *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 $\mathsf{Decaps}$, we argue that the challenge key cannot be distinguished from random unless the adversary $\mathcal{A}$ queries $\mathsf{H}$ on the challenge plaintext. When reducing to plain $\mathsf{OW\text{-}CPA}$ security, a reduction would have to guess, but a reduction to $q_{\mathsf{H}}$-$\mathsf{OW\text{-}CPA}$ security can simply keep a list of all of $\mathcal{A}$ queries to $\mathsf{H}$ and return this list as the list of plaintext guesses. For the sake of completeness, a full proof is given in Appendix B.

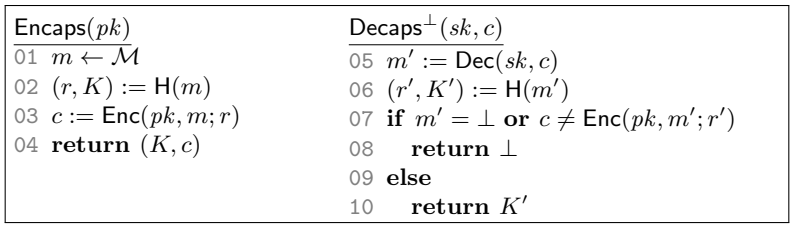


**Fig. 4.** Key encapsulation mechanism $\mathsf{KEM} = \mathsf{FO}^{\perp}[\mathsf{PKE}, \mathsf{H}]$, obtained from $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with worst-case correctness error.

**Theorem 2.4** (IND-CPA of PKE $\overset{\mathsf{ROM}}{\Longrightarrow}$ IND-CCA of KEM [**HHK17**])**.** *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*

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

*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 [**DFMS21**])**.** *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*

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

*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 $\mathsf{ACWC}_0$ 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 $\mathsf{ACWC}_0$ by describing $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$ in Fig. 5 which adds $\lambda$ bits of redundancy to the ciphertexts, where $\lambda$ is the size of the message space. The resulting scheme has small worst-case correctness error.

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

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

| $\mathsf{Enc}'(pk, m \in \{0,1\}^\lambda)$ | $\mathsf{Dec}'(sk, (c, u))$ |
|---|---|
| 01 $r \leftarrow \psi_{\mathcal{R}}$ | 03 $r := \mathsf{Dec}(sk, c)$ |
| 02 **return** $(\mathsf{Enc}(pk, r), \mathsf{F}(r) \oplus m)$ | 04 **return** $\mathsf{F}(r) \oplus u$ |

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

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

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

**Theorem 3.3 ($q_{\mathsf{F}}$-OW-CPA of $\mathsf{PKE} \overset{\text{ROM}}{\Longrightarrow}$ 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* $\mathsf{F}$*, there exists an adversary* $\mathcal{B}$ *against the* OW-CPA *security of* $\mathsf{PKE}$ *with*

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

*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 $\mathsf{F}$ on $r$, thus breaking OW-CPA security of $\mathsf{PKE}$. A reduction to $q_{\mathsf{F}}$-OW-CPA security can simply keep a list of all of $\mathcal{A}$ queries to $\mathsf{F}$ 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 Appendix C.1.

**Theorem 3.4 ($p_{\mathsf{F}}$-OW-CPA of $\mathsf{PKE} \overset{\text{QROM}}{\Longrightarrow}$ 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* $\mathsf{PKE}$ *with*

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

*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 $\mathsf{F}$ 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 $p_{\mathsf{F}}$, the list has at most $p_{\mathsf{F}}$ many elements. For the sake of completeness, a full proof of Theorem 3.4 is given in Appendix C.2.

### 3.2 Transformation ACWC without redundancy

Let $\mathsf{PKE}$ be an encryption scheme with small average-case correctness error, and let $\mathsf{F}$ 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 $\mathsf{GOTP}$.

**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* $\mathsf{Inv}$ *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 $\mathsf{PKE}$ 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 $\mathsf{F}$ is such that every output is independently $\psi_{\mathcal{U}}$-distributed.) $\mathsf{PKE}'$ obtained by transformation $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$ is described in Fig. 6.

| $\underline{\mathsf{Enc}'(pk, m \in \mathcal{M}')}$ | $\underline{\mathsf{Dec}'(sk, c)}$ |
|---|---|
| 01 $M_1 \leftarrow \psi_{\mathcal{M}_1}$ | 04 $M_1 \| M_2 := \mathsf{Dec}(sk, c)$ |
| 02 $M_2 := \mathsf{GOTP}(m, \mathsf{F}(M_1))$ | 05 $m := \mathsf{Inv}(M_2, \mathsf{F}(M_1))$ |
| 03 **return** $\mathsf{Enc}(pk, M_1 \| M_2)$ | 06 **return** $m$ |

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

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

**Lemma 3.6.** *Let* $\mathsf{PKE}$ *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 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* $\mathsf{PKE}$ *is* $\delta$*-average-case-correct then* $\mathsf{PKE}' := \mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$ *is* $\delta'$ *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.* With the expectation over choice of $\mathsf{F}$ and $(pk, sk) \leftarrow \mathsf{Gen}$, we have for the worst-case correctness of $\mathsf{PKE}'$:

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

[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 $R$.

where $\delta'(pk, sk) := \mathbb{E}\left[\max_{m \in \mathcal{M}'} \Pr\left[\mathsf{Dec}'(sk, \mathsf{Enc}'(pk, m)) \neq m\right]\right]$ is the expectation taken over choice of $\mathsf{F}$, for a fixed key pair $(pk, sk)$. For any fixed key pair and for any real $t$, we have that

$$\begin{aligned}
\delta'(pk, sk) &= \mathbb{E}\left[\max_{m \in \mathcal{M}'} \Pr\left[\mathsf{Dec}'(sk, \mathsf{Enc}'(pk, m)) \neq m\right]\right] \\
&\leq t \ + \Pr_{\mathsf{F}}\left[\max_{m \in \mathcal{M}'} \Pr\left[\mathsf{Dec}'(sk, \mathsf{Enc}'(pk, m)) \neq m\right] \geq t\right] \\
&= t \ + \Pr_{\mathsf{F}}\left[\max_{m \in \mathcal{M}'} \Pr_{M_1, r}[\mathsf{Dec}'(sk, \mathsf{Enc}(pk, M_1 || \mathsf{GOTP}(m, \mathsf{F}(M_1)); r) \neq m] \geq t\right] \ . \qquad (1)
\end{aligned}$$

For any fixed key pair and any real $c$, let $t(pk, sk) := \mu(pk, sk) + \|\psi\| \cdot \sqrt{(c + \ln|\mathcal{M}'|)/2}$, where $\mu(pk, sk) := \Pr_{M,r}[\mathsf{Dec}(sk, \mathsf{Enc}(pk, M; r)) \neq M]$ and $\psi = \psi_{\mathcal{M}_1}$. We can now use helper Lemma 3.7 below to argue that

$$\Pr_{\mathsf{F}}\left[\max_{m \in \mathcal{M}'} \Pr_{M_1, r}[\mathsf{Dec}'(sk, \mathsf{Enc}(pk, M_1 || \mathsf{GOTP}(m, \mathsf{F}(M_1)); r)) \neq m] > t(pk, sk)\right] \leq e^{-c} \ . \qquad (2)$$

To this end, we identify $r_1$ from Lemma 3.7 with $M_1$, $r_2$ with $r$, $g(m, r_1, r_2, u)$ with $(r_1 || \mathsf{GOTP}(m, u), r_2)$ and $B$ with $\{(M, r) \in | \ \mathsf{Dec}(sk, \mathsf{Enc}(pk, M; r) \neq m\}$. Plugging Eq. (2) into Eq. (1) and taking the expectation yields

$$\begin{aligned}
\delta' &\leq \mathbb{E}\left[\mu(pk, sk) + \|\psi_{\mathcal{M}_1}\| \cdot \sqrt{(c + \ln|\mathcal{M}'|)/2} + e^{-c}\right] \\
&= \delta + \|\psi_{\mathcal{M}_1}\| \cdot \sqrt{(c + \ln|\mathcal{M}'|)/2} + e^{-c} \ ,
\end{aligned}$$

and setting $c := -\ln\|\psi_{\mathcal{M}_1}\|$ gives the claim in the lemma. □

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

$$\forall m \in \mathcal{M}, \Pr_{r_1 \leftarrow \psi_1, r_2 \leftarrow \psi_2, u \leftarrow U}[g(m, r_1, r_2, u) \in B] \leq \mu, \qquad (3)$$

*where $\psi_1$ and $\psi_2$ are independent. Let $\mathsf{F}$ 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 $\mathsf{F}$, we have that $\forall m \in \mathcal{M}$,*

$$\Pr_{r_1 \leftarrow \psi_1, r_2 \leftarrow \psi_2}[g(m, r_1, r_2, \mathsf{F}(r_1)) \in B] \leq \mu + \|\psi_1\| \cdot \sqrt{(c + \ln|\mathcal{M}|)/2} \qquad (4)$$

*Proof.* The proof strategy will be to show that for any fixed $m \in \mathcal{M}$, the probability in (4) holds for all but a $e^{-c} \cdot |\mathcal{M}|^{-1}$-fraction of random functions $\mathsf{F}$. The claim of the statement then follows by the union bound.

Let us fix a specific $m \in \mathcal{M}$, and for each $r_1 \in \mathrm{supp}(\psi_1)$, define $p_{r_1} := \Pr_{r_2 \leftarrow \psi_2, u \leftarrow U}[g(m, r_1, r_2, u) \in B]$. By the assumption on $g$ in (3), we know that $\sum_{r_1} p_{r_1} \psi(r_1) \leq \mu$. For each $r_1$, define a random variable $X_{r_1}$ whose value is determined as follows: $\mathsf{F}$ chooses a random $u = \mathsf{F}(r_1)$ and $r_2 \leftarrow \psi_2$, and then checks whether $g(m, r_1, r_2, \mathsf{F}(r_1)) \in B$; if it is, then we set $X_{r_1} = 1$, and otherwise we set it to 0. Because $\mathsf{F}$ is a random function, the probability that $X_{r_1} = 1$ is exactly $p_{r_1}$.

The probability of (4) for our particular $m$ is now exactly the sum $\sum_{r_1} \psi_1(r_1) X_{r_1}$ and we will use the Hoeffding bound to show that this value is not much larger than $\mu$. Define the random variable $Y_{r_1} = \psi_1(r_1) X_{r_1}$. Notice that $Y_{r_1} \in [0, \psi_1(r_1)]$, and that $\mathbb{E}\left[\sum Y_{r_1}\right] = \mathbb{E}\left[\sum X_{r_1} \psi_1(r_1)\right] = \sum p_{r_1} \psi_1(r_1) = \mu$. By the Hoeffding bound, we have for all positive $t$,

$$\Pr\left[\sum Y_{r_1} > \mu + t\right] \leq \exp\left(\frac{-2t^2}{\sum \psi_1(r_1)^2}\right) = \exp\left(\frac{-2t^2}{\|\psi_1\|^2}\right). \tag{5}$$

Setting $t \geq \|\psi_1\| \cdot \sqrt{(c + \ln|\mathcal{M}|)/2}$, we obtain that for a *fixed* $m$, (4) holds for all but a $e^{-c} \cdot |\mathcal{M}|^{-1}$ fraction of random functions $\mathsf{F}$. Applying the union bound gives us the claim in the lemma. □

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

*Proof.* The proof is a simple observation, it follows by applying the law of total probability in order to fix $M_1$. For fixed $M_1$ one can then apply $\gamma$-spreadness for a fixed key pair $(pk, sk)$. By averaging over $(pk, sk)$ the claim then follows.

**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* $q_{\mathsf{F}}$ *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* $\mathbf{Time}(\mathcal{A}) + \mathcal{O}(q \cdot q_{\mathsf{F}})$ .

In the $q$-OW-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 $\mathsf{F}$ 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 $\mathsf{F}$ 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 $\mathsf{F}$ 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 $\mathsf{GOTP}$, $\mathcal{A}$'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 Appendix D.1.

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

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

*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 $\mathsf{ACWC}_0$, however, we can not simply apply the O2H lemma, as a reduction needs both a query to $\mathsf{F}$ 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 [DFMS19, DFM20], 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 Appendix D.2.

## 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 $\psi$ (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 $\psi$. 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 $\psi_k^d$ over $\mathbb{Z}^d$ to be the distribution

$$\sum_{i=1}^{k} \vec{a}_i - \sum_{i=1}^{k} \vec{b}_i, \quad \vec{a}_i, \vec{b}_i \leftarrow \{0,1\}^d. \tag{6}$$

The distribution $\bar{\psi}_k^d$ is the distribution over the set $\{-1,0,1\}^d$ defined as $\psi_k^d$ reduced modulo 3. We will mostly be working with $\bar{\psi}_k^d$ and $\psi_k^d$ 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

$\psi_2 =$

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

(7)

$$\bar{\psi}_2 = \begin{array}{|c||c|c|c|}\hline \text{Output} & \text{-1} & 0 & 1 \\ \hline \text{Prob.} & 5/16 & 6/16 & 5/16 \\ \hline \end{array} \tag{8}$$

We now state a lemma, which is used for the construction of NTRU-A in Figure 7 that shows that by creating the distribution $\psi_2$ 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 $\psi_2^d$ 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 [HPS98], 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 $(\mathbf{h}, \mathbf{hr} + \mathbf{e})$, 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 $(\mathbf{h}, \mathbf{hr} + \mathbf{e})$, 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 [SS11], 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$_\eta$ assumption).** *For a distribution $\eta$ over the ring $\mathcal{R}$ and an integer $p$ relatively-prime to $q$, the $\mathcal{R}$-NTRU$_\eta$ 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 $\eta$, 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** ($\mathcal{R}$-$\mathsf{LWE}_\eta$)**.** *Let $\eta$ be some distribution over $\mathcal{R}$. In the $\mathcal{R}$-$\mathsf{LWE}$ 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** $\mathcal{R}$-$\mathsf{LWE}_\eta$)**.** *Let $\eta$ be a distribution over $\mathcal{R}$. The decision $\mathcal{R}$-$\mathsf{LWE}$ 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 $\mathsf{LWE}$ definition of Regev [Reg09], the distribution $\eta$ 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 $\mathcal{R}$-$\mathsf{LWE}_\eta$. When implementing cryptographic primitives based on the hardness of $\mathcal{R}$-$\mathsf{LWE}_\eta$, it is more convenient to take $\eta$ 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 $\psi_k$ and $\bar{\psi}_k$ from (7) and (8).

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

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

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 $\mathcal{L}^{\perp}_{\mathbf{0}}(\mathbf{h})$. If $\mathbf{h}$ is random, then a vector of $\ell_2$-norm less than $\Omega(\sqrt{qd})$ is very unlikely to exist in $\mathcal{L}^{\perp}_{\mathbf{c}}(\mathbf{h})$. On the other hand, if the coefficients of $\mathbf{f}, \mathbf{g}$ are noticeably less than $\sqrt{q}$, then $(\mathbf{f}, -\mathbf{g}) \in \mathcal{L}^{\perp}_{\mathbf{c}}(\mathbf{h})$, 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 $\mathbf{c} \neq 0$, $\mathcal{L}^{\perp}_{\mathbf{c}}(\mathbf{h})$ 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 $\mathcal{L}^{\perp}_{\mathbf{c}}(\mathbf{h})$ is significantly easier when $\mathbf{c} = 0$, as opposed to when $\mathbf{c}$ is random [ABD16, CJL16, KF17, DvW21]. 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 $\mathcal{R}$-$\mathsf{LWE}$ problem in which the adversary is not asked to recover the entire vector $\mathbf{e}$, but just $\mathbf{e} \bmod 2$.

**Definition 4.5** ($\mathcal{R}$-$\mathsf{LWE2}_\eta$)**.** *Let $\eta$ be a distribution over $\mathcal{R}$. In the $\mathcal{R}$-$\mathsf{LWE}$ 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 $\mathbf{e} \bmod 2$.*

While we do not have a formal reduction from $\mathcal{R}$-$\mathsf{LWE}$ 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 $\mathcal{R}$-$\mathsf{LWE}$ and $\mathcal{R}$-$\mathsf{LWE2}$.

Suppose that there is an algorithm that solves $\mathcal{R}$-LWE2$_\eta$ and we feed it an instance $(\mathbf{h}, \mathbf{hr} + \mathbf{e})$ of $\mathcal{R}$-LWE$_\eta$. If the $\mathcal{R}$-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 $\psi_2$, then each coefficient of $\mathbf{e}'$ has a probability 3/16 of being $\pm 1$ 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 $\mathcal{R}$-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 $\mathcal{R}$-LWE2$_\eta$. In the reduction, we simply add a noise with distribution $\eta$ 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 $\mathcal{R}$-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 [BLP$^+$13] considers this "First-is-Errorless" version of LWE and shows that it is essentially as hard as the usual version. Boudgoust et al. [BJRW21] 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 $\mathcal{R}$-LWE2$_\eta$ problem is the same as that of $\mathcal{R}$-LWE$_\eta$.

### 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 OW-CPA-secure encryption scheme, NTRU-A, whose hardness is based on the combination of the $\mathcal{R}$-NTRU$_\eta$ + $\mathcal{R}$-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 OW-CPA NTRU scheme, and yet we can only recover a part of it. This is not a OW-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. $\psi_2$) 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.

**OW-CPA Scheme.** The distribution of the coefficients of the secret polynomials used in key generation and encryption $\psi_2$ (see (7)) 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 $(b_1 -$

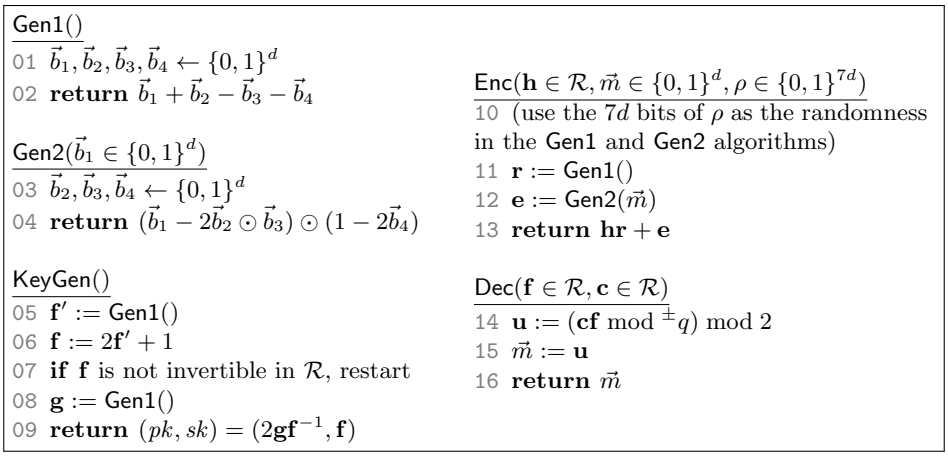


**Fig. 7.** OW-CPA Encryption Scheme NTRU-A based on the $\mathcal{R}$-NTRU$_{\psi_2}$ + $\mathcal{R}$-LWE2$_{\psi_2}$ problems. Only the procedures Gen1 and Gen2 are randomized. We include the coins $\rho$ as input for the Encryption algorithm (which will be passed to Gen1 and Gen2) because these are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

$2b_2b_3)(1-2b_4)$, 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 $\mathbf{h} = 2\mathbf{gf}^{-1}$ 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 $\psi_2^d$. To create such a distribution, we define $\mathbf{e} =$ Gen2($\vec{m}$). By Lemma 4.1, $\mathbf{e}$ is distributed according to $\psi_2^d$. The ciphertext is $\mathbf{c} = \mathbf{hr} + \mathbf{e}$.

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

$$(\mathbf{cf} \bmod^{\pm} q) \bmod 2 = 2\mathbf{gr} + \mathbf{ef} \bmod 2 = 2\mathbf{gr} + 2\mathbf{ef}' + \mathbf{e} \bmod 2 \tag{9}$$

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 [LS19, 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 (9) 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 $\mathbf{a} = \sum_{i=0}^{5} \mathbf{a}_i$ and $\mathbf{b} = \sum_{i=0}^{5} \mathbf{b}_i$ then their product $\mathbf{c} = \sum_{i=0}^{5} \mathbf{c}_i$ can be written as in (10).

$$\begin{bmatrix} \mathbf{a}_0 & -\mathbf{a}_5 & -\mathbf{a}_4 & -\mathbf{a}_3 & -\mathbf{a}_2 - \mathbf{a}_5 & -\mathbf{a}_1 - \mathbf{a}_4 \\ \mathbf{a}_1 & \mathbf{a}_0 & -\mathbf{a}_5 & -\mathbf{a}_4 & -\mathbf{a}_3 & -\mathbf{a}_2 - \mathbf{a}_5 \\ \mathbf{a}_2 & \mathbf{a}_1 & \mathbf{a}_0 & -\mathbf{a}_5 & -\mathbf{a}_4 & -\mathbf{a}_3 \\ \mathbf{a}_3 & \mathbf{a}_2 + \mathbf{a}_5 & \mathbf{a}_1 + \mathbf{a}_4 & \mathbf{a}_0 + \mathbf{a}_3 & \mathbf{a}_2 & \mathbf{a}_1 \\ \mathbf{a}_4 & \mathbf{a}_3 & \mathbf{a}_2 + \mathbf{a}_5 & \mathbf{a}_1 + \mathbf{a}_4 & \mathbf{a}_0 + \mathbf{a}_3 & \mathbf{a}_2 \\ \mathbf{a}_5 & \mathbf{a}_4 & \mathbf{a}_3 & \mathbf{a}_2 + \mathbf{a}_5 & \mathbf{a}_1 + \mathbf{a}_4 & \mathbf{a}_0 + \mathbf{a}_3 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{b}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \mathbf{c}_4 \\ \mathbf{c}_5 \end{bmatrix} \tag{10}$$

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

$$c = ba + b'(a + a'). \tag{11}$$

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 (11), we can think of it as the sum of three random variables of the form

$$c = ba + b'a'. \tag{12}$$

It should be clear that the distribution of (11) is wider than that of (12), 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 (11) and two from (12); while $\mathbf{c}_2$ is the sum of two from (11) and one from (12).

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 (11). The term $\mathbf{gr}$ in (9) has this exact distribution, where each coefficient of $\mathbf{g}, \mathbf{r}$ is distributed according to $\mathsf{Gen1}()$.

The term $\mathbf{f'e}$ 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 (13), we give the distribution of a particular coefficient of $\mathbf{e}_i$ conditioned on the message bit being either 0 or 1.

$\mathsf{Gen2}(0) =$

| Output | -2 | 0 | 2 |
|---|---|---|---|
| Probability | 0.125 | 0.75 | 0.125 |

$\mathsf{Gen2}(1) =$

| Output | -1 | 1 |
|---|---|---|
| Probability | 0.5 | 0.5 |

(13)

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 (13). As in [ADPS16, BDK+18, LS19], the error probability reported in Table 1 is computed via polynomial multiplications which represent convolutions of random variables.

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

**IND-CPA-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 NTRU-A. The concrete security bounds on the IND-CCA security of CCA-NTRU-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 IND-CCA secure NTRU-variants CCA-NTRU-A, CCA-NTRU-B, and CCA-NTRU-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 $\varepsilon$ 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 $\eta$ 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 compute $\mathbf{r} = (\mathbf{c} - \mathbf{e}) \cdot \mathbf{h}^{-1}$. Because we checked that $\mathbf{g}$ is invertible, it holds that $\mathbf{h} = 3\mathbf{gf}^{-1}$ also has an inverse.

$\underline{\mathsf{KeyGen}()}$
01 $\mathbf{f}', \mathbf{g} \leftarrow \eta$
02 $\mathbf{f} := 3\mathbf{f}' + 1$
03 **if** $\mathbf{f}$ or $\mathbf{g}$ is not invertible in $\mathcal{R}$, restart
04 **return** $(pk, sk) = (3\mathbf{gf}^{-1}, \mathbf{f})$

$\underline{\mathsf{Enc}(\mathbf{h} \in \mathcal{R}, \vec{m} \in \{-1, 0, 1\}^d)}$
05 $\mathbf{r} \leftarrow \eta$
06 **return** $\mathbf{c} := \mathbf{hr} + \vec{m}$

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

**Fig. 8.** Generic NTRU $\mathsf{GenNTRU}[\eta]$ relative to distribution $\psi$ 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 OW-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 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.

**NTRU-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 NTRU-B := ACWC[GenNTRU[$U_3^d$], GOTP, F], obtained by applying the ACWC transformation from Section 3.2 to 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 $\mathcal{M}_2 = \{-1,0,1\}^\lambda$.

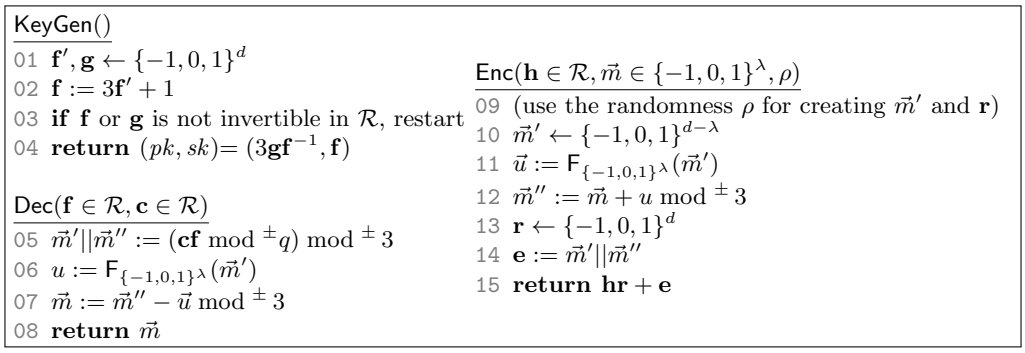


**Fig. 9.** Randomness-recoverable OW-CPA encryption scheme NTRU-B with worst-case correctness error based on the $\mathcal{R}$-NTRU$_{U_3^d}$ + $\mathcal{R}$-LWE$_{U_3^d}$ problems for $U_3^d$ being uniform over $\{-1,0,1\}^d$.

By Lemma 3.6, the average-case correctness error of GenNTRU[$U_3^d$] and the worst-case correctness error of NTRU-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 NTRU-B.

Finally, one can apply the Fujisaki-Okamoto transformation FO$^\perp$ from Fig. 4 to obtain the IND-CCA secure version CCA-NTRU-B := FO$^\perp$[NTRU-B, H] of NTRU-B. In the ROM, the concrete security bound on the IND-CCA security of CCA-NTRU-B from Table 4 can be derived by combining Lemma 2.2 with 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 := ACWC$_0$[GenNTRU[$\bar{\psi}_2^d$], F] with uniform message space $\mathcal{M}' = \{0,1\}^\lambda$, obtained by applying the ACWC$_0$ transformation with redundancy from Section 3.1 to GenNTRU[$\bar{\psi}_2^d$] is described in Fig. 10. By Lemma 3.1, the average-case correctness error of 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 FO$^\perp$ from Fig. 4 to obtain the IND-CCA secure version CCA-NTRU-C := FO$^\perp$[NTRU-C, 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.

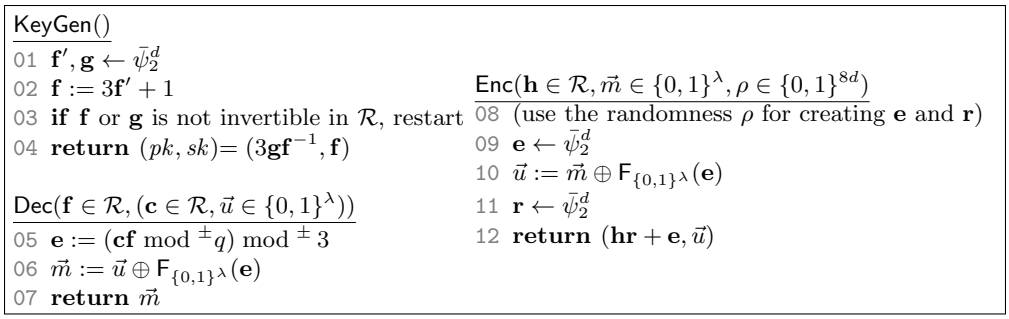


**Fig. 10.** NTRU-C: a randomness-recoverable OW-CPA encryption scheme with worst-case correctness error based on the $\mathcal{R}$-NTRU$_\eta$ + $\mathcal{R}$-LWE$_\eta$ problems for $\eta = \bar{\psi}_2^d$.

## Acknowledgements

Julien Duman was supported by the EU H2020 PROMETHEUS project 780701. Eike Kiltz was supported by the BMBF iBlockchain project, the EU H2020 PROMETHEUS project 780701, and by the DFG under Germany's Excellence Strategy - EXC 2092 CASA - 390781972. Vadim Lyubashevsky and Gregor Seiler are supported by the EU H2020 ERC Project 101002845 PLAZA.

## References

ABD16. Martin R. Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In *CRYPTO (1)*, volume 9814 of *Lecture Notes in Computer Science*, pages 153–178. Springer, 2016.

ADPS16. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In *USENIX Security Symposium*, pages 327–343. USENIX Association, 2016.

AHU19. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In *Advances in Cryptology – CRYPTO 2019*, volume 11693 of *Lecture Notes in Computer Science*, pages 269–295. Springer, 2019. `https://ia.cr/2018/904`.

APS15. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *J. Math. Cryptol.*, 9(3):169–203, 2015.

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *International conference on the theory and application of cryptology and information security*, pages 41–69. Springer, 2011.

BDK+18. Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In *EuroS&P*, pages 353–367. IEEE, 2018.

BJRW21. Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. On the hardness of module-lwe with binary secret. In *CT-RSA*, volume 12704 of *Lecture Notes in Computer Science*, pages 503–526. Springer, 2021.

BLP+13. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584, 2013.

CHK+21. Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, and Bo-Yin Yang. NTT multiplication for ntt-unfriendly rings new speed records for saber and NTRU on cortex-m4 and AVX2. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(2):159–188, 2021.

CJL16. Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. An algorithm for ntru problems and cryptanalysis of the ggh multilinear map without a low-level encoding of zero. *LMS Journal of Computation and Mathematics*, 19(A):255–266, 2016.

DFM20. Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round Fiat-Shamir and more. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology -*

*CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 602–631. Springer, 2020.

DFMS19. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019*, volume 11693 of *Lecture Notes in Computer Science*, pages 356–383. Springer, 2019.

DFMS21. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. Cryptology ePrint Archive, Report 2021/280, 2021. `https://ia.cr/2021/280`.

DHK+21. Julien Duman, Kathrin Hövelmanns, Eike Kiltz, Vadim Lyubashevsky, and Gregor Seiler. Faster lattice-based KEMs via a generic Fujisaki-Okamoto transform for multi-user security in the QROM. In *CCS*, 2021.

DKRV18. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure KEM. In *AFRICACRYPT*, pages 282–305, 2018.

Duc18. Léo Ducas. Shortest vector from lattice sieving: A few dimensions for free. In *EUROCRYPT (1)*, volume 10820 of *Lecture Notes in Computer Science*, pages 125–145. Springer, 2018.

DvW21. Léo Ducas and Wessel P. J. van Woerden. NTRU fatigue: How stretched is overstretched? *IACR Cryptol. ePrint Arch.*, page 999, 2021.

HHK17. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. In *TCC*, pages 341–371, 2017.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288, 1998.

HRSS17. Andreas Hülsing, Joost Rijneveld, John M. Schanck, and Peter Schwabe. High-speed key encapsulation from NTRU. In *CHES*, volume 10529 of *Lecture Notes in Computer Science*, pages 232–252. Springer, 2017.

KF17. Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched NTRU parameters. In *EUROCRYPT (1)*, volume 10210 of *Lecture Notes in Computer Science*, pages 3–26, 2017.

LM06. Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP (2)*, pages 144–155, 2006.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography*, 75(3):565–599, 2015.

LS19. Vadim Lyubashevsky and Gregor Seiler. NTTRU: truly fast NTRU using NTT. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):180–201, 2019.

Reg09. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

SS11. Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *EUROCRYPT*, pages 27–47, 2011.

SSTX09. Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT*, pages 617–635, 2009.

Unr15. Dominique Unruh. Revocable quantum timed-release encryption. *J ACM*, 62(6):49:1–49:76, 2015.

## A Quantum Preliminaries

We recall some quantum computation preliminaries.

QUBIT. A qubit $|x\rangle = \alpha|0\rangle + \beta|1\rangle$ is a 2-dimensional unit vector with coefficients in $\mathbb{C}$, i.e. $x = (\alpha, \beta) \in \mathbb{C}^2$ fulfilling the normalization constraint $|\alpha|^2 + |\beta|^2 = 1$. When neither $\alpha = 1$ nor $\beta = 1$, we say that $|x\rangle$ is in *superposition*.

$n$-QUBIT STATE. An n-bit quantum register $|x\rangle = \sum_{i=1}^{2^n-1} \alpha_i|i\rangle$ is a unit vector of $\mathbb{C}^{2^n} = (\mathbb{C}^2)^{\otimes n}$, that is $\alpha_i \in \mathbb{C}$ and $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$. We call the set $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$ the *computational basis*. When $|x\rangle$ can not be written as the tensor product of single qubits, we say that $|x\rangle$ is *entangled*.

Measurement. Unless otherwise stated, measurements are done in the computational basis. After measuring a quantum register $|x\rangle = \sum_{i=0}^{2^n-1} a_i|i\rangle$ in the computational basis, the state *collapses* and $|x\rangle = \pm|i\rangle$ with probability $|\alpha_i|^2$.

Quantum Algorithms. A quantum algorithm $\mathcal{A}$ is a series of unitary operations $U_i$, where unitary operations are defined as to map unit vectors to unit vectors, preserving the normalization constraint of quantum registers. A quantum oracle algorithm $\mathcal{A}^{\mathsf{O}}$ is defined similarly, except it can query the oracle $\mathsf{O}$ after (or before) executing a unitary $U_i$. Since quantum computation needs to be reversible, we model an oracle $\mathsf{O} : X \to Y$ by a unitary $U_{\mathsf{O}}$ that maps $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus \mathsf{O}(x)\rangle$.

Quantum Random Oracle Model. Following [BDF+11], we model quantum adversaries to have quantum access to random oracles since quantum adversaries can evaluate hash functions in superposition.

Below, we recall the oneway-to-hiding lemma, which is used to reprogram random oracle values. Informally, Theorem A.1 states that if a random oracle is reprogrammed on a set $S \subset R$ of inputs, the probability of an adversary $\mathcal{A}$ behaving differently can be related to the success probability of an extractor algorithm $\mathcal{B}$ which extracts at least on element of $S$ by measuring the query register of one of $\mathcal{A}$'s randomly chosen oracle queries.

**Theorem A.1 (Classical O2H, Theorem 3 from the eprint version of [AHU19]).** *Let $S \subset \mathcal{R}$ be random. Let $\mathsf{F}, \mathsf{G}$ be random functions satisfying $\forall r \notin S : \mathsf{F}(r) = \mathsf{G}(r)$. Let $z$ be a random classical value. ($S, G, H, z$ may have arbitrary joint distribution.) Let $\mathcal{C}$ be a quantum oracle algorithm with query depth $q_{\mathsf{F}}$, expecting input $z$. Let $\mathcal{D}$ be the algorithm which on input $z$ samples a uniform $i$ from $\{1, \ldots, q_{\mathsf{F}}\}$, runs $\mathcal{C}$ right before its $i$th query to $\mathsf{G}$, measures all query input registers and outputs the set $\mathcal{T}$ of measurement outcomes. Then*

$$\left|\Pr[\mathcal{C}^{\mathsf{F}}(z) \Rightarrow 1] - \Pr[\mathcal{C}^{\mathsf{G}}(z) \Rightarrow 1]\right| \leq 2d_{\mathsf{F}}\sqrt{\Pr\left[S \cap T \neq \emptyset : T \leftarrow \mathcal{D}^{\mathsf{G}}(z)\right]}.$$

We slightly reformulated the theorem: We changed variable names to better fit the present context. We do not require $z$ to be a bitstring but allow it to be an arbitrary classical value (in this proof: a tuple). And in the original, the argument to $\mathcal{D}^{\mathsf{G}}$ would be $\mathsf{F}$, but by symmetry it also holds with $\mathsf{G}$.

## B Proof of Theorem 2.3 (Tight IND-CCA security of $\mathsf{FO}^{\perp}$ from $q_{\mathsf{H}}$-OW-CPA security of PKE )

*Proof.* Consider the games given in Fig. 11. We proceed by analyzing the changes introduced by the different games.

Game $G_0$. This is the original IND-CCA game, by definition we have

$$\left|\Pr\left[G_0^{\mathcal{A}} \Rightarrow 1\right] - \frac{1}{2}\right| = \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A})\,.$$

Game $G_1$. In this game we change the decapsulation oracle to return $\perp$, when it is queried on a ciphertext with no preimage from the list $\mathcal{L}_{\mathsf{H}}$ in line 13. By the weak $\gamma$-spreadness, the probability that the re-encryption check passes for a message which does not use randomness $\rho$ from $\mathsf{H}(m)$ is upper bounded by $2^{-\gamma}$. By a union bound over $q_{\mathsf{D}}$ decapsulation queries we have

$$\left|\Pr\left[G_0^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[G_1^{\mathcal{A}} \Rightarrow 1\right]\right| = q_{\mathsf{D}}2^{-\gamma}\,.$$

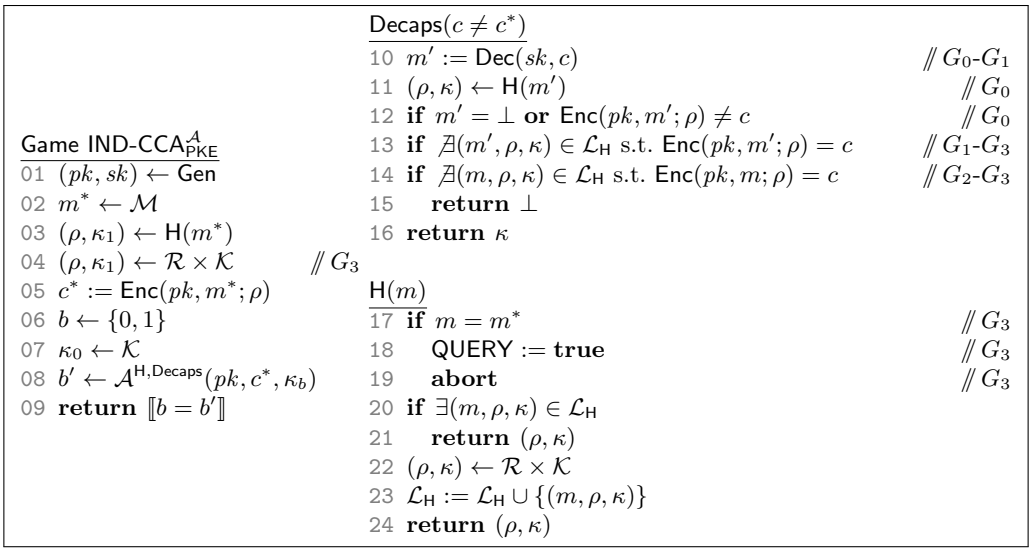


**Fig. 11.** Game $G_0$-$G_3$ for the proof of Theorem 2.3

GAME $G_2$. In this game, we change the decapsulation oracle to not need the secret-key anymore. Concretely, $G_2$ introduces line 14, which is the same as line 13, except that any $m$ which passes the re-encryption check is checked, instead of $m = m' := \mathsf{Dec}(sk, c)$. Clearly, whenever Decaps returns $\perp$ in $G_2$, it returns $\perp$ in $G_1$, since if no $m$ exists, this means there can also not exist $m = m'$. The other direction holds by $\delta$-correctness, since when no $m'$ exists, the boolean expression in $G_2$ (line 14) will only evaluate to false, if there exists another $m \neq m'$ with $\mathsf{Enc}(pk, m; \rho) = c$, but this implies a correctness error, since $m' := \mathsf{Dec}(sk, c) \neq m$. If a message $m \in \mathcal{L}_{\mathsf{H}}$ exists, it will be unique, since collisions imply correctness errors, by a similar argument. By the $\delta$-correctness with a union bound over the $q_{\mathsf{H}}$ random oracle queries, we have

$$\left|\Pr\left[G_1^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[G_2^{\mathcal{A}} \Rightarrow 1\right]\right| = q_{\mathsf{H}}\delta\,.$$

GAME $G_3$. In this game we introduce the lines 17-19 to the random oracle H. The event QUERY is set to true, if H is queried on $m^*$ in which case the game is aborted. Additionally, in line 04 we sample $\kappa_0 \leftarrow \mathcal{K}$ and the randomness $\rho$ for the challenge ciphertext from $\mathcal{K}$, instead from H, which will not be noticed by $\mathcal{A}$ as long as QUERY does not happen. We have by the difference lemma

$$\left|\Pr\left[G_2^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[G_3^{\mathcal{A}} \Rightarrow 1\right]\right| = \Pr\left[\mathsf{QUERY}\right]\,.$$

By a reduction to the $q_{\mathsf{H}}$-OW-CPA security of PKE we can show that there exists an adversary $\mathcal{B}$ with

$$\Pr\left[\mathsf{QUERY}\right] \leq \mathsf{Adv}^{q_{\mathsf{H}}\text{-OW-CPA}}_{\mathsf{PKE}}(\mathcal{B})\,. \tag{14}$$

The reduction $\mathcal{B}$ works as follows: it obtains by the $q_{\mathsf{H}}$-OW-CPA a challenge public-key and challenge ciphertext, which are forwarded to the CCA adversary together with $\kappa \leftarrow \mathcal{K}$. The ciphertext and $\kappa$ have the right distribution due to line 04. It then simulates the decapsulation and random oracle as in $G_2$ and $G_3$ and saves all random oracle queries in $\mathcal{L}_{\mathsf{H}}$. It outputs to the $q_{\mathsf{H}}$-OW-CPA challenger

the random oracle quries as the solution set $\mathcal{Q}$. When $\mathsf{QUERY}$ happens, the reduction $\mathcal{B}$ wins. We have thus shown equation (14). [12]

GAME $G_3$ In game $G_3$ the bit $b$ is independent of the view of the adversary, since $\kappa_0, \kappa_1 \leftarrow \mathcal{K}$ due to line 04, thus we have

$$\Pr\left[G_3^{\mathcal{A}} \Rightarrow 1\right] = \frac{1}{2}\,.$$

Summing up the inequalities yields the claimed bound, which concludes our proof.

# C Proofs of Theorem 3.3 and Theorem 3.4 (IND-CPA security of $\mathsf{ACWC_0}$ in the (Q)ROM)

## C.1 Proof of Theorem 3.3 (IND-CPA security of $\mathsf{ACWC_0}$ in the ROM)

GAME $G_1$. In game $G_1$ (see Fig. 12), we have the original $\mathsf{IND\text{-}CPA}$ game, with $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$ plugged in in line 06. By definition, we have that

$$\left|\Pr\left[G_1^{\mathcal{A}} \Rightarrow 1\right] - \frac{1}{2}\right| = \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ACWC}_0[\mathsf{PKE},\mathsf{F}]}(\mathcal{A}).$$

GAME $G_2$. In game $G_2$, we raise flag $\mathsf{QUERY}$ and abort on the event that $\mathcal{A}$ queries $\mathsf{F}$ on $r^*$. We have that

$$\left|\Pr\left[G_2^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[G_1^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \Pr\left[\mathsf{QUERY}\right].$$

Unless $\mathsf{QUERY}$ occurs, $\mathsf{F}(r)$ is a uniform value independent of $\mathcal{A}$'s view, meaning that $u^* = \mathsf{F}(r) \oplus m_b$ is uniformly random, and the input $b'$ of $\mathcal{A}^{\mathsf{F}}$ is independent of $b$. Thus, $\Pr\left[G_2^{\mathcal{A}} \Rightarrow 1\right] = 1/2$ and

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ACWC}_0[\mathsf{PKE},\mathsf{F}]}(\mathcal{A}) \leq \Pr\left[\mathsf{QUERY}\right].$$

We construct adversary $\mathcal{B}$ against the underlying scheme in Fig. 12. $\mathcal{B}$ perfectly simulates the input to $\mathcal{A}$ until $\mathsf{QUERY}$ occurs, simulates the random oracle via lazy sampling (see lines 17, 19 and 20), and keeps track of $\mathcal{A}$'s oracle queries by adding them to its internal list $\mathcal{L}_r$ (see line 21), which $\mathcal{B}$ forwards as the solution set to its $q_{\mathsf{F}}$-$\mathsf{OW\text{-}CPA}$ challenger . Since $\mathcal{B}$ wins if $\mathsf{QUERY}$ occurs and $\mathcal{B}$'s output list has at most $q_{\mathsf{F}}$ many elements,

$$\Pr\left[\mathsf{QUERY}\right] \leq \mathsf{Adv}^{q_{\mathsf{F}}\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}).$$

## C.2 Proof of Theorem 3.4 (IND-CPA security of $\mathsf{ACWC_0}$ in the QROM)

*Proof.* The idea of the proof is to reprogramm $\mathsf{F}$ on $r$ using the O2H lemma. In order to apply the lemma, we need to introduce conceptual game changes in $G_2$ and $G_3$. After applying the O2H lemma, the adversary wins in $G_4$ only with probability $1/2$, since $u \oplus m_b$ information-theoretically hides $b$, for uniform $u$. We proceed by analyzing the games given in Figs. 13 to 15.

[12] When $\mathsf{QUERY}$ happens the adversary $\mathcal{B}$ won't be able to simulate properly anymore, but this does not matter since it will win the $q_{\mathsf{H}}$-$\mathsf{OW\text{-}CPA}$ game anyway, since $m^*$ will be in $\mathcal{Q}$. The reduction needs an upper bound on the running time of $\mathcal{A}$ though for this argument to hold, since otherwise $\mathcal{A}$ could change its running time (e.g. run in an infinite loop) when it notices that it runs in the simulation. Alternatively, if no upper bound is known to $\mathcal{B}$, it can run $\mathcal{A}$ several times, each time doubling its running time.

| Game $G_1$ | $\mathcal{B}(pk, c_1^*)$ | $\mathsf{F}(r)$ // As simulated by $\mathcal{B}$ |
|---|---|---|
| 01 $\mathsf{F} \leftarrow (\mathcal{R} \to \{0,1\}^\lambda)$ | 09 $\mathcal{L}_\mathsf{F}, \mathcal{L}_r := \emptyset$ | 16 **if** $\exists (r, m) \in \mathcal{L}_\mathsf{F}$ |
| 02 $(pk, sk) \leftarrow \mathsf{Gen}$ | 10 $b \leftarrow \{0,1\}$ | 17 **return** $m$ |
| 03 $b \leftarrow \{0,1\}$ | 11 $u^* \leftarrow \{0,1\}^\lambda$ | 18 **else** |
| 04 $(m_0, m_1) \leftarrow \mathcal{A}_1^\mathsf{F}(pk)$ | 12 $(m_0, m_1) \leftarrow \mathcal{A}_1^\mathsf{F}(pk)$ | 19 $m \leftarrow \{0,1\}^\lambda$ |
| 05 $r^* \leftarrow \psi_\mathcal{R}$ | 13 $c^* := (c_1^*, u^* \oplus m_b)$ | 20 $\mathcal{L}_\mathsf{F} := \mathcal{L}_\mathsf{F} \cup \{(r, m)\}$ |
| 06 $(c_1^*, u^*) \leftarrow (\mathsf{Enc}(pk, r^*), \mathsf{F}(r^*) \oplus m_b)$ | 14 $b' \leftarrow \mathcal{A}_2^\mathsf{F}(pk, c^*)$ | 21 $\mathcal{L}_r := \mathcal{L}_r \cup \{r\}$ |
| 07 $b' \leftarrow \mathcal{A}_2^\mathsf{F}(pk, (c_1^*, u^*))$ | 15 **return** $\mathcal{L}_r$ | 22 **return** $m$ |
| 08 **return** $[\![b = b']\!]$ | | |

**Fig. 12.** Games $G_1$ and adversary $\mathcal{B}$ for the proof of Theorem 3.4

GAME $G_1$. In game $G_1$ we have the original IND-CPA game, with $\mathsf{ACWC}_0[\mathsf{PKE}, \mathsf{F}]$ plugged in, in line 06. By definition, we have

$$\left| \Pr\left[G_1^\mathcal{A} \Rightarrow 1\right] - \frac{1}{2} \right| = \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ACWC}_0[\mathsf{PKE},\mathsf{F}]}(\mathcal{A}).$$

| Game $G_1$ |
|---|
| 01 $\mathsf{F} \leftarrow (\mathcal{R} \to \{0,1\}^\lambda)$ |
| 02 $(pk, sk) \leftarrow \mathsf{Gen}$ |
| 03 $b \leftarrow \{0,1\}$ |
| 04 $(m_0, m_1) \leftarrow \mathcal{A}_1^\mathsf{F}(pk)$ |
| 05 $r \leftarrow \psi_\mathcal{R}$ |
| 06 $c^* \leftarrow (\mathsf{Enc}(pk, r), \mathsf{F}(r) \oplus m_b)$ |
| 07 $b' \leftarrow \mathcal{A}_2^\mathsf{F}(pk, c^*)$ |
| 08 **return** $[\![b = b']\!]$ |

**Fig. 13.** Game $G_1$ for the proof of Theorem 3.4

GAME $G_2$. In $G_2$ we restructure game $G_1$ by moving part of the game into an adversary $\mathcal{C}^\mathsf{F}$, defined in Fig. 14. We also introduce an auxiliary variable for $u := \mathsf{F}(r)$, i.e., we query $\mathsf{F}(r)$ earlier. The resulting game is $G_2$. By unfolding the definitions of $\mathcal{C}^\mathsf{F}$ and $u$, we get $G_1$, therefore the change is only conceptual, hence

$$\Pr\left[G_1^\mathcal{A} \Rightarrow 1\right] = \Pr\left[G_2^\mathcal{A} \Rightarrow 1\right].$$

GAME $G_3$. In game $G_3$ we change how $\mathsf{F}$ and $u$ are chosen. Instead of choosing $\mathsf{F}$ uniformly, and then computing $u$, we choose $\mathsf{G}$ and $u$ uniformly, and let $\mathsf{F} := \mathsf{G}$, except that $\mathsf{F}(r) = u$. This leads to the same distribution of $\mathsf{F}$ and $u$, hence

$$\Pr\left[G_2^\mathcal{A} \Rightarrow 1\right] = \Pr\left[G_3^\mathcal{A} \Rightarrow 1\right].$$

For the next proof step, we will use Theorem A.1. Applying this theorem with $\mathcal{C}$ being the adversary described above, $S := \{r\}$, and $z := (r, u)$, we immediately get:

$$\left|\Pr\left[G_3^\mathcal{A} \Rightarrow 1\right] - \Pr\left[G_4^\mathcal{A} \Rightarrow 1\right]\right| \leq 2 d_\mathsf{F} \sqrt{\Pr\left[G_5 \Rightarrow 1\right]}.$$

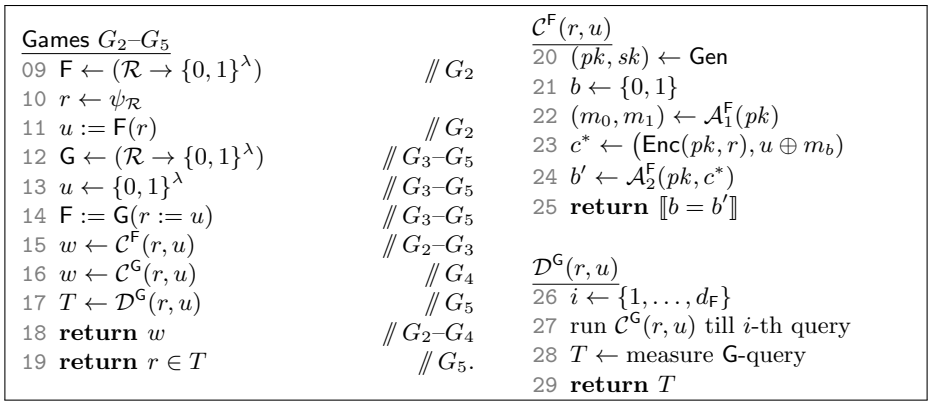


**Fig. 14.** Games $G_2$-$G_5$ for the proof of Theorem 3.4. In this definition, "run ... till the $i$-th query" means till the $i$-th block of up to $p_{\mathsf{F}}$ parallel queries. And $T \leftarrow$ measure G-query means that we measure the query input register of $\mathsf{G}$ in all of the parallel queries in that query block and let $T$ be the set of the measurement outcomes.

(The initial lines of those games, where $\mathsf{G}, r, u, \mathsf{F}$ are chosen, define a joint distribution as in the theorem.)

Game $G_4$. In game $G_4$, the uniformly random value $u$ (line 13) is only used in the expression $u \oplus m_b$ (line 23). Thus $u \oplus m_b$ is uniformly random and hence the input $b'$ of $\mathcal{A}^{\mathsf{F}}$ is independent of $b$. Thus $b = b'$ with probability $1/2$. Therefore,

$$\Pr\left[G_4^{\mathcal{A}} \Rightarrow 1\right] = \frac{1}{2}\,.$$

Game $G_5$ and $G_6$. We wrap most of $G_5$ into the adversary $\mathcal{B}$ defined in Fig. 15, except for the choice of $pk, sk, r$ and the first half of the ciphertext $c^*$. The result is game $G_6$ in Fig. 15 and we have $\Pr[G_6 \Rightarrow 1] = \Pr[G_5 \Rightarrow 1]$. Note that

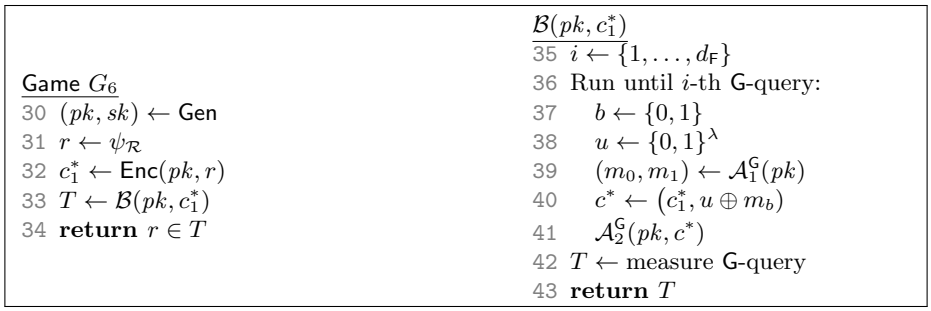


**Fig. 15.** Game $G_6$ for the proof of Theorem 3.4. The adversary $\mathcal{B}$ is the extractor algorithm from the O2H lemma.

$$\mathsf{Adv}^{p_{\mathsf{F}}\text{-}\mathsf{OW}\text{-}\mathsf{CPA}}_{\mathsf{PKE}}(\mathcal{B}) = \Pr[G_6 \Rightarrow 1]$$

because $T$ will always contain at most $p_{\mathsf{F}}$ elements (the query parallelism). Putting all equalities and bounds in this proof together, we get

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ACWC}_0[\mathsf{PKE},\mathsf{F}]}(\mathcal{A}) \leq 2d_{\mathsf{F}} \sqrt{\mathsf{Adv}^{p_{\mathsf{F}}\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B})}\,,$$

which concludes our proof.

# D Proofs of Theorem 3.9 and Theorem 3.10 ((*q*-)OW-CPA security of ACWC in the (Q)ROM)

## D.1 Proof of Theorem 3.9 (*q*-OW-CPA security of ACWC in the ROM)

*Proof.* Consider the games given in Fig. 16.

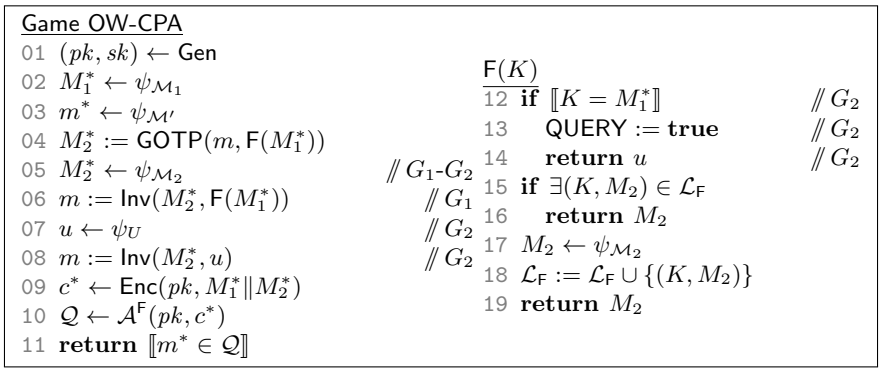


**Fig. 16.** $q$-OW-CPA game against $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$.

Game $G_0$. This is the original $q$-OW-CPA game, we have

$$\Pr\left[G_0^{\mathcal{A}} \Rightarrow 1\right] = \mathsf{Adv}^{q\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{ACWC}[\mathsf{PKE},\mathsf{GOTP},\mathsf{F}]}(\mathcal{A})\,.$$

Game $G_1$. In game $G_1$ we introduce the following conceptual change: instead of sampling $m^* \leftarrow \mathcal{M}'$, $M_1^* \leftarrow \psi_{\mathcal{M}_1}$ and computing $M_2^* := \mathsf{GOTP}(m^*, \mathsf{F}(M_1))$, we instead sample $M_2^* \leftarrow \psi_{\mathcal{M}_2}$ (using the randomness-hiding property) and $M_1^* \leftarrow \psi_{\mathcal{M}_1}$ and then define $m^* := \mathsf{Inv}(M_2^*, \mathsf{F}(M_1^*))$ (follows implictly by the decoding property). Due to the randomness-hiding and decoding property of the generalized one-time pad this change is only conceptual, we have

$$\Pr\left[G_0^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[G_1^{\mathcal{A}} \Rightarrow 1\right]\,.$$

Game $G_2$. In game $G_2$ we set the flag QUERY, when $\mathcal{A}$ queries $M_1$ which was used to create the challenge ciphertext. Since this is only conceptual, we have

$$\Pr\left[G_1^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[G_2^{\mathcal{A}} \Rightarrow 1\right]\,.$$

We proceed by bounding the probability of the adversary $\mathcal{A}$ winning in game $G_2$. We have

$$\Pr\left[G_2^{\mathcal{A}} \Rightarrow 1\right] \leq \Pr\left[G_2^{\mathcal{A}} \Rightarrow 1 \wedge \mathsf{QUERY}\right] + \Pr\left[G_2^{\mathcal{A}} \Rightarrow 1 \wedge \overline{\mathsf{QUERY}}\right] .$$

We claim that there exists an adversary $\mathcal{B}$ with

$$\Pr\left[G_2^{\mathcal{A}} \Rightarrow 1 \wedge \mathsf{QUERY}\right] \leq \mathsf{Adv}_{\mathsf{PKE}}^{q \cdot q_{\mathsf{F}}\text{-}\mathsf{OW\text{-}CPA}}(\mathcal{B}) \tag{15}$$

and

$$\Pr\left[G_2^{\mathcal{A}} \Rightarrow 1 \wedge \overline{\mathsf{QUERY}}\right] \leq q \cdot 2^{-H_\infty(\psi_{\mathcal{M}'})} . \tag{16}$$

To see why (16) holds, observe that $m^*$ is defined as $\mathsf{Inv}(M_2^*, \mathsf{F}(M_1^*))$. Since $\mathcal{A}$ did not query $M_1^*$, the value $\mathsf{F}(M_1^*)$ looks uniformly random to $\mathcal{A}$, and therefore $\mathsf{Inv}(M_2^*, \mathsf{F}(M_1^*))$ is distributed as $\psi_{\mathcal{M}'}$. Thus, the probability of $m^* \in \mathcal{Q}$ is upper bounded by $q \cdot 2^{-H_\infty(\psi_{\mathcal{M}'})}$. We proceed by showing (15). Consider the reduction given in Fig. 17. The reduction $\mathcal{B}$ runs $\mathcal{A}$ and records its random oracle queries and obtains a candidate set $\mathcal{Q}_{\mathcal{A}}$. It then constructs the set $\mathcal{Q}$ which contains 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 $m' \in \mathcal{Q}_{\mathcal{A}}$, which is a set of size at most $q \cdot q_{\mathsf{F}}$. Since we conditioned on $\mathsf{QUERY}$, we know that $\mathcal{L}_{\mathsf{F}}$ will contain the right $M_1^*$ used to create the challenge ciphertext. Therefore, $\mathcal{B}$ wins if $\mathcal{A}$ wins and the claim follows, which concludes the proof.

$\underline{\mathcal{B}(pk, c^*)}$
20 $\mathcal{Q} := \emptyset$
21 $\mathcal{Q}_{\mathcal{A}} \leftarrow \mathcal{A}^{\mathsf{F}}(pk, c^*)$
22 **for** $(M_1', m') \in \mathcal{L}_{\mathsf{F}} \times \mathcal{Q}_{\mathcal{A}}$
23 $\quad M_2' := \mathsf{GOTP}(m', \mathsf{F}(M_1'))$
24 $\quad M' := M_1' || M_2'$
25 $\quad \mathcal{Q} := \mathcal{Q} \cup \{M'\}$
26 **return** $\mathcal{Q}$

**Fig. 17.** Reduction $\mathcal{B}$ against the $(q \cdot q_{\mathsf{F}})$-OW-CPA-security of PKE. The set $\mathcal{L}_{\mathsf{F}}$ is the set of recorded random oracle queries.

### D.2 Proof of Theorem 3.10 (OW-CPA security of ACWC in the QROM)

*Proof.* The main idea of the proof is to run the adversary $\mathcal{A}$ to obtain $m$ and measure and reprogram one of the random oracle queries to obtain $M_1$. This can be done using Theorem 2 from [DFM20], after carefully introducing several conceptual game changes ($G_1$-$G_4$). After some additional conceptual game changes, we upper bound the probability of $G_8$ by the OW-CPA advantage. We proceed by analyzing the games given in Figs. 18 to 21.

Game $G_1$. In game $G_1$ we have the original OW-CPA game, where we already unfolded the definition of $\mathsf{ACWC}[\mathsf{PKE}, \mathsf{GOTP}, \mathsf{F}]$. (In slight abuse of notation, we write $\mathsf{F} \leftarrow (\mathcal{M}_1 \to \mathcal{U})$ for sampling $\mathsf{F}$ such that every output is independently $\psi_{\mathcal{U}}$-distributed.) By definition, we have

$$\Pr\left[G_1 \Rightarrow 1\right] = \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{ACWC}[\mathsf{PKE},\mathsf{GOTP},\mathsf{F}]}(\mathcal{A}) .$$

| Games $G_1$–$G_3$ | |
|---|---|
| 01 $\mathsf{F} \leftarrow (\mathcal{M}_1 \to \mathcal{U})$ | |
| 02 $(pk, sk) \leftarrow \mathsf{Gen}$ | |
| 03 $m^* \leftarrow \psi_{\mathcal{M}'}$ | // $G_1$–$G_2$ |
| 04 $M_1^* \leftarrow \psi_{\mathcal{M}_1}$ | |
| 05 $M_2^* := \mathsf{GOTP}(m^*, F(M_1^*))$ | // $G_1$–$G_2$ |
| 06 $M_2^* \leftarrow \psi_{\mathcal{M}_2}$ | // $G_3$ |
| 07 $c^* \leftarrow \mathsf{Enc}(pk, (M_1^*, M_2^*))$ | |
| 08 $m \leftarrow \mathcal{A}^F(pk, c^*)$ | |
| 09 **return** $m = m^*$ | // $G_1$ |
| 10 **return** $V(M_1^*, \mathsf{F}(M_1^*), (M_1^*, M_2^*, m))$ | // $G_2$–$G_3$ |

**Fig. 18.** Games $G_1$-$G_3$ for the proof of Theorem 3.10. The function $V$ is defined in (17).

GAME $G_2$. In game $G_2$, we change the winning condition (return value). Instead of checking $m = m^*$ ine line 09, we check $V(M_1, \mathsf{F}(M_1), (M_1^*, M_2^*, m))$ in line 10, which is defined as

$$V(M_1, \Theta, (M_1^*, M_2^*, m)) := \Big(m = \mathsf{Inv}(M_2^*, \Theta) \wedge M_1 = M_1^* \wedge M_2^* \in \operatorname{im} \mathsf{GOTP}(-, \Theta)\Big). \tag{17}$$

Since $\mathsf{Inv}(M_2^*, \mathsf{F}(M_1^*)) = \mathsf{Inv}(\mathsf{GOTP}(m^*, \mathsf{F}(M_1^*)), \mathsf{F}(M_1^*)) = m^*$ (the last equality by the properties of the generalize one-time pad $f$), and since $M_1^* = M_1^*$ and $M_2^* \in \operatorname{im} \mathsf{GOTP}(-, F(M_1^*))$ hold trivially by choice of $M_2^*$, the return values in both games are equal. Thus,

$$\Pr[G_1 \Rightarrow 1] = \Pr[G_2 \Rightarrow 1].$$

GAME $G_3$. In game $G_3$, we replace the definition $M_2^* := \mathsf{GOTP}(m^*, \mathsf{F}(M_1^*))$ in line 05 (for $m^* \leftarrow \psi_{\mathcal{M}'}$) by the random sampling $M_2^* \leftarrow \psi_{\mathcal{M}_2}$ in line 06. Note that $m^*$ is never used anywhere else. By the randomness-hiding property of the generalized one-time pad $\mathsf{GOTP}$, the value $M_2^*$ has the same distribution when sampled in this way. Thus,

$$\Pr[G_2 \Rightarrow 1] = \Pr[G_3 \Rightarrow 1].$$

GAME $G_4$. Define the adversary $\tilde{\mathcal{A}}$ as shown in Fig. 19. Then, game $G_3$ results from the game $G_4$ as shown in Fig. 19 by unfolding $\tilde{\mathcal{A}}$. Thus,

$$\Pr[G_3 \Rightarrow 1] = \Pr[G_4 \Rightarrow 1] .$$

[DFM20] shows the following theorem that we will use for the next proof step:

**Theorem D.1 (Measure-and-reprogram).** *Let $\mathcal{M}_1$,$\mathcal{U}$ be finite non-empty sets. Fix an oracle quantum algorithm $\tilde{\mathcal{A}}$ making $q_\mathsf{F}$ queries to a uniformly random $\mathsf{F} : \mathcal{M}_1 \to \mathcal{U}$ that outputs $(M_1, z)$ with $M_1 \in \mathcal{M}_1$ and $z$ classical or quantum (here, we only use classical $z$). Let $V(M_1, \Theta, z)$ be a predicate (where $\Theta \in \mathcal{U}$). Let $I := (\{0, \ldots, q_\mathsf{F} - 1\} \times \{0, 1\}) \cup \{(q_\mathsf{F}, 0)\}$.*

*Let $\mathcal{S}_1^{\tilde{\mathcal{A}}}, \mathcal{S}_2^{\tilde{\mathcal{A}}}$ be the following quantum algorithms:*

| $\underline{\mathcal{S}_1^{\tilde{\mathcal{A}}}}$ | $\underline{\mathcal{S}_2^{\tilde{\mathcal{A}}}(\Theta)}$ |
|---|---|
| 20 $(i, b) \leftarrow I$ | 25 $\tilde{\mathcal{A}}^\mathsf{F}$ *till* $(i+b)$*-th query* |
| 21 $\mathsf{F} \leftarrow (\mathcal{M}_1 \to \mathcal{U})$ | 26 $\mathsf{F} \leftarrow (\mathcal{M}_1 \to \mathcal{U})$ |
| 22 $\tilde{\mathcal{A}}^\mathsf{F}$ *till* $i$*-th query* | 27 $\tilde{\mathcal{A}}^\mathsf{F}$ *till* $i$*-th query* |
| 23 $M_1 \leftarrow$ *measure* $\mathsf{F}$*-query* | 28 $(M_1', (M_1^*, M_2^*, m)) \leftarrow \tilde{\mathcal{A}}^{\mathsf{F}(M_1 := \Theta)}()$ *till the end* |
| 24 ***return*** $M_1$ | 29 ***return*** $(M_1^*, M_2^*, m)$. |

| Game $G_4$ | $\tilde{\mathcal{A}}^{\mathsf{F}}()$ |
|---|---|
| 11 $\mathsf{F} \leftarrow (\mathcal{M}_1 \to \mathcal{U})$ | 14 $(pk, sk) \leftarrow \mathsf{Gen}$ |
| 12 $(M_1, (M_1^*, M_2^*, m)) \leftarrow \tilde{\mathcal{A}}^{\mathsf{F}}()$ | 15 $M_1^* \leftarrow \psi_{\mathcal{M}_1}$ |
| 13 **return** $V(M_1, \mathsf{F}(M_1), (M_1^*, M_2^*, m))$ | 16 $M_2^* \leftarrow \psi_{\mathcal{M}_2}$ |
| | 17 $c^* \leftarrow \mathsf{Enc}(pk, (M_1^*, M_2^*))$ |
| | 18 $m \leftarrow \mathcal{A}^{\mathsf{F}}(pk, c^*)$ |
| | 19 **return** $(M_1^*, (M_1^*, M_2^*, m))$. |

**Fig. 19.** Game $G_4$ and adversary $\tilde{\mathcal{A}}$ for the proof of Theorem 3.10.

*This adversary runs $\tilde{\mathcal{A}}$ internally, but not in one go. "Run $\tilde{\mathcal{A}}^{\mathsf{F}}$ till the $i$-th query" means that the execution (with oracle $\mathsf{F}$) is stopped just after executing the $i$-th query. (Where numbering of queries starts with the 0-th query. And running till the $q$-th query just means running till the end.) "$M_1 \leftarrow$ measure $\mathsf{F}$-query" means that the content of the query input register used when querying $\mathsf{F}$ is measured and the outcome of that measurement assigned to $M_1$. (If we run till the $q$-th query, i.e., till the end, then this measures the final output $M_1$ of the adversary.) Note that $\mathcal{S}_1^{\tilde{\mathcal{A}}}, \mathcal{S}_2^{\tilde{\mathcal{A}}}$ share state between each other.*

*Then for all $x_0 \in \mathcal{M}_1$,*

$$\Pr\left[M_1 = x_0 \wedge V(M_1, \Theta, z) : \Theta \leftarrow (U, M_1) \leftarrow \mathcal{S}_1^{\tilde{\mathcal{A}}}, z \leftarrow \mathcal{S}_2^{\tilde{\mathcal{A}}}(\Theta)\right] \geq \frac{1}{(2q_{\mathsf{F}}+1)^2} \Pr\left[M_1 = x_0 \wedge V(M_1, \mathsf{F}(M_1), z) : (M_1, z) \leftarrow \tilde{\mathcal{A}}^{\mathsf{F}}\right].$$

(We have slightly reformulated Theorem 2 from [DFM20] here: We changed variable names to match the ones used here. We wrote the simulator $\mathcal{S}^{\tilde{\mathcal{A}}}$ as two algorithms instead of a "two-stage algorithm". And we made the definition of $\mathcal{S}^{\tilde{\mathcal{A}}}$ explicit; in [DFM20] it can be found in the proof of the theorem.)

Game $G_5$. If we sum the inequality in Theorem D.1 over all $x_0 \in \mathcal{M}_1$, the term $M_1 = x_0$ vanishes on both sides. The probability on the right hand side is then $\Pr[G_4 \Rightarrow 1]$. And the probability on the left hand side is $\Pr[G_5 \Rightarrow 1]$ for the game $G_5$ below that we obtain by unfolding the definition of $\mathcal{S}_1^{\tilde{\mathcal{A}}}, \mathcal{S}_2^{\tilde{\mathcal{A}}}$. Thus, we get

$$\Pr[G_4 \Rightarrow 1] \leq (2q_{\mathsf{F}}+1)^2 \Pr[G_5 \Rightarrow 1].$$

Game $G_6$. By unfolding the definition of $\tilde{\mathcal{A}}$ in $G_5$, we get game $G_6$. The choices of $pk, sk, M_1^*, M_2^*, c^*$ are performed by $\tilde{\mathcal{A}}$ before any queries are performed, so they are always performed at the beginning of "run $\tilde{\mathcal{A}}^{\mathsf{F}}$ till $i$-th query". Thus,

$$\Pr[G_5 \Rightarrow 1] = \Pr[G_6 \Rightarrow 1].$$

Game $G_7$. In game $G_7$, we replace the return value $V(M_1, \Theta, (M_1^*, M_2^*, m))$ in line 44 by $(M_1^*, M_2^*) = (M_1, \mathsf{GOTP}(m, \Theta))$ in line 45. (Note that it is possible that $m \notin \mathcal{M}'$ since it is adversarially generated. In that case, we make no assumptions what $\mathsf{GOTP}(m, \Theta)$ has to return. It could be $\perp$ or some arbitrary value. This freedom makes it easier to find an efficient implementation of the adversary $\mathcal{B}$ below in case that membership in $\mathcal{M}'$ is hard to decide.) We need to show

$$V(M_1, \Theta, (M_1^*, M_2^*, m)) \implies (M_1^*, M_2^*) = (M_1, \mathsf{GOTP}(m, \Theta)). \tag{18}$$

Games $G_5$–$G_7$

$\Theta \leftarrow \mathcal{U}$
$(i, b) \leftarrow I$
$\mathsf{F} \leftarrow (\mathcal{M}_1 \to \mathcal{U})$
$\tilde{\mathcal{A}}^{\mathsf{F}}$ till $i$-th query // $G_5$
$(pk, sk) \leftarrow \mathsf{Gen}$ // $G_6$–$G_7$
$M_1^* \leftarrow \psi_{\mathcal{M}_1}$ // $G_6$–$G_7$
$M_2^* \leftarrow \psi_{\mathcal{M}_2}$ // $G_6$–$G_7$
$c^* \leftarrow \mathsf{Enc}(pk, (M_1^*, M_2^*))$ // $G_6$–$G_7$
$\mathcal{A}^{\mathsf{F}}(pk, c^*)$ till $i$-th query // $G_6$–$G_7$
$M_1 \leftarrow$ measure $\mathsf{F}$-query
$\tilde{\mathcal{A}}^{\mathsf{F}}$ till $(i+b)$-th query // $G_5$
$\mathcal{A}^{\mathsf{F}}$ till $(i+b)$-th query // $G_6$–$G_7$
$(M_1', (M_1^*, M_2^*, m)) \leftarrow \tilde{\mathcal{A}}^{\mathsf{F}(M_1 := \Theta)}()$ till the end // $G_5$
$m \leftarrow \mathcal{A}^{\mathsf{F}(M_1 := \Theta)}()$ till the end // $G_6$–$G_7$
**return** $V(M_1, \Theta, (M_1^*, M_2^*, m))$ // $G_5$–$G_6$
**return** $(M_1^*, M_2^*) = (M_1, \mathsf{GOTP}(m, \Theta))$ // $G_7$

**Fig. 20.** Games $G_5$-$G_7$ for the proof of Theorem 3.10.

That the lhs implies $M_1^* = M_1$ is trivial. To see that the lhs implies $M_2^* = \mathsf{GOTP}(m, \Theta)$, note the following: $\mathsf{Inv}(-, \Theta)$ is a left inverse of $\mathsf{GOTP}(-, \Theta)$ by the decoding property of the generalized one-time pad $\mathsf{GOTP}$. Any function composed with its left inverse is the identity on its image. $M_2^*$ is in the image of $\mathsf{GOTP}(-, \Theta)$ and $m = \mathsf{Inv}(M_2^*, \Theta)$ by definition of $V$. Thus $\mathsf{GOTP}(m, \Theta) = \mathsf{GOTP}(\mathsf{Inv}(M_2^*, \Theta), \Theta) = M_2^*$. This shows that the rhs is implied. From (18) we get

$$\Pr[G_6 \Rightarrow 1] \leq \Pr[G_7 \Rightarrow 1] .$$

Game $G_8$. Finally, we define the adversary $\mathcal{B}$ in Fig. 21, and then game $G_8$ in Fig. 21 is the same as $G_7$ up to unfolding of the definition of $\mathcal{B}$, reordering of independent statements, and using the fact that $M^* \leftarrow \psi_{\mathcal{M}_1 \times \mathcal{M}_2}$ gives the same distribution as $M_1^* \leftarrow \psi_{\mathcal{M}_1}, M_2^* \leftarrow \psi_{\mathcal{M}_2}$ if $M^*$ is the pair $(M_1^*, M_2^*)$. Thus,

$$\Pr[G_7 \Rightarrow 1] = \Pr[G_8 \Rightarrow 1] .$$

By definition of $\mathsf{OW\text{-}CPA}$, we have

$$\Pr[G_8 \Rightarrow 1] = \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}) .$$

Combining all inequalities, we get

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

which concludes our proof.

$\underline{\mathsf{Game}\ G_8}$
46 $(pk, sk) \leftarrow \mathsf{Gen}$
47 $M^* \leftarrow \psi_{\mathcal{M}_1 \times \mathcal{M}_2}$
48 $c^* \leftarrow \mathsf{Enc}(pk, M^*)$
49 $M \leftarrow \mathcal{B}(pk, c^*)$
50 **return** $M = M^*$

$\underline{\mathcal{B}(pk, c^*)}$
51 $\Theta \leftarrow \mathcal{U}$
52 $(i, b) \leftarrow I$
53 $\mathsf{F} \leftarrow (\mathcal{M}_1 \to \mathcal{U})$
54 $\mathcal{A}^{\mathsf{F}}(pk, c^*)$ till $i$-th query
55 $M_1 \leftarrow$ measure $\mathsf{F}$-query
56 $\mathcal{A}^{\mathsf{F}}$ till $(i + b)$-th query
57 $m \leftarrow \mathcal{A}^{\mathsf{F}(M_1 := \Theta)}$ till the end
58 **return** $(M_1, \mathsf{GOTP}(m, \Theta))$.

**Fig. 21.** Game $G_8$ and $\mathcal{B}$ for the proof of Theorem 3.10.