Round5: Compact and Fast Post-Quantum Public-Key Encryption

Sauvik Bhattacharya[1], Oscar Garcia-Morchon[1], Thijs Laarhoven[2], Ronald Rietman[1], Markku-Juhani O. Saarinen[3], Ludo Tolhuizen[1], and Zhenfei Zhang[4]

[1] Royal Philips N.V., Netherlands. Email: {first.lastname}@philips.com
[2] Eindhoven University of Technology, Netherlands. Email: mail@thijs.com
[3] PQShield Ltd., United Kingdom. Email: mjos@mjos.fi
[4] OnBoard Security, USA. Email: zzhang@onboardsecurity.com

Abstract. Standardization bodies such as NIST and ETSI are currently seeking quantum resistant alternatives to vulnerable RSA and elliptic curve-based public-key algorithms. In this context, we present Round5, a lattice-based cryptosystem providing a key encapsulation mechanism and a public-key encryption scheme. Round5 is based on the General Learning with Rounding problem, unifying non-ring and ring lattice rounding problems into one. Usage of rounding combined with a tight analysis leads to significantly reduced bandwidth and randomness requirements. Round5's reliance on prime-order cyclotomic rings offers a large design space allowing fine-grained parameter optimization. The use of sparse-ternary secret keys improves performance and significantly reduces decryption failure rates at minimal additional cost. The use of error-correcting codes further improves the latter. Round5 parameters have been carefully optimized for bandwidth, while the design facilitates efficient implementation. As a result, Round5 has leading performance characteristics among all NIST post-quantum candidates, and at the same time attains conservative security levels that fully fit NIST's security categories. Round5's schemes share common building blocks, simplifying (security and operational) analysis and code review. Finally, Round5 proposes various approaches of refreshing the system public parameter $\boldsymbol{A}$, which efficiently prevent precomputation and back-door attacks.

Keywords: Lattice cryptography · Post-quantum Cryptography · Learning With Rounding · Prime cyclotomic ring · Key encapsulation · CCA Security · CPA Security.

1 Introduction

Due to the inherent vulnerability of RSA and Elliptic Curve cryptography to attacks by quantum computers and the relatively long time period that public key encryption algorithms must guarantee the confidentiality of their secrets, a transition to quantum-secure alternatives has been initiated by the U.S. Government and the information security community. Standardization bodies such

as NIST [52] and ETSI [32,33] are currently in the process of evaluating and standardizing Post-Quantum Cryptography (PQC).

Lattice-based cryptography is a prominent branch of post-quantum cryptography that is based on well studied problems and often offers very good performance characteristics. Among others, there exist lattice-based proposals for key exchange [20,18,3], key encapsulation [19] [28], public key encryption [24,25] and digital signatures [31,30]. The main hard problem underlying the security of most lattice-based proposals is the Learning with Errors (LWE) problem defined on general *Euclidean* lattices.

The decision variant of the LWE problem refers to distinguishing uniform samples $(\boldsymbol{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ from samples of the form $(\boldsymbol{a}, \boldsymbol{a}^T\boldsymbol{s} + e)$ where $\boldsymbol{a}$ is uniform on $\mathbb{Z}_q^n$ (multiple m samples of which constitute the problem's public parameter $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$), the secret $\boldsymbol{s}$ is drawn uniformly from $\mathbb{Z}_q^n$, and e is drawn from a known error distribution on the integers modulo q. Matrix multiplication and vector addition are performed modulo q.

The ring variant of LWE (RLWE) introduces more structured *ideal* lattices [48] for better performance. *Module* lattices [46] allow for additional flexibility in the parameter choice and are structurally in between the former two.

In the Learning with Rounding (LWR) problem [10], the independent, randomly drawn error e from LWE is replaced by a deterministic error via rounding $\boldsymbol{as}$ to a smaller modulus p. An earlier version of this technique was used in "modulus switching" to limit the growth of noise in fully homomorphic encryption [21]. The decision variant of the LWR problem is to distinguish uniform samples $(\boldsymbol{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, from samples of the form $(\boldsymbol{a}, \left\lfloor \frac{p}{q}\boldsymbol{a}^T\boldsymbol{s} \right\rceil)$, where $\boldsymbol{a}$ is uniform on $\mathbb{Z}_q^n$, $\boldsymbol{s}$ is uniform on $\mathbb{Z}_q^n$ and $\lfloor \cdot \rceil$ denotes (coordinate-wise) rounding to the closest integer modulo p. LWR has practical advantages over LWE: it requires the generation of less randomness as there is no need to explicitly generate the components of the noise vector e. Furthermore, rounding results in smaller ciphertexts, as they have p-ary symbols instead of q-ary symbols.

1.1 Separate Solutions for LWE and RLWE

The (decision variants of the) ring learning with errors (RLWE) and ring learning with rounding (RLWR) are defined analogously to LWE and LWR, however with the parameter $\boldsymbol{a}$ and secret $\boldsymbol{s}$ replaced with elements from a chosen polynomial ring $\mathcal{R}$. Due to the structure of ideal lattices, the hardness assumptions for these problems are considered less conservative than for LWE and LWR. On the other hand, RLWE and RLWR are more efficient than their non-ring counterparts [50]. No scheme has however been fully defined with the flexibility of fitting diverse use-cases with diverse trust requirements, e.g., Ring-LWE against LWE assumptions. Furthermore, for some use-cases with critical performance requirements IND-CPA (indistinguishability under chosen plaintext attack [55]) security may be enough, in which cases designing for only slower IND-CCA (indistinguishability under chosen ciphertext attack [55]) security might be over-provisioning.

We will now give some examples of applications and their particular requirements. A high-performance IPSec [43] solution may require a ring-based scheme for shorter messages and lower latency; such a scheme also makes key refreshing easier, thus ensuring forward secrecy for which CPA-security may be sufficient. In contrast, securing email requires CCA-security since public keys are long-term; still, a well-performing solution is needed so that the overhead is low even for small emails. On the other hand, a governmental VPN [58] may want to trade some of the key exchange performance to the added security assurance offered by unstructured lattice. Similarly, long-term security of healthcare records requires a public key encryption solution that avoids additional security assumptions.

Looking into the state of the art, we find solutions that fit individual applications, but no solution that can be easily configured to fit *all* of them. For instance, Frodo [18] is based on the conservative LWE assumption, but is rather inefficient for performance-intensive scenarios, requiring bandwidth as high as 23 kilobytes. Kyber [19] is comparatively efficient but depends on one single underlying ring choice. Most schemes such as [18], [19], [28] or [65] are defined to provide either IND-CPA or IND-CCA security, but not both.

1.2 Inflexibility in Ring Selection

The choice of the underlying polynomial ring greatly affects the performance of schemes based on ideal lattices. A common choice [20,3] of the polynomial ring to instantiate an RLWE or RLWR problem is $\mathbb{Z}_q[x]/\Phi_{2n}(x)$ where n is a power of 2, so that $\Phi_{2n}(x)$ denotes the $2n$-th cyclotomic polynomial $x^n + 1$. However, requiring that n be a power of 2 restricts the choice of n. For example, $n = 512$ results in a lattice problem not hard enough to achieve a 128-bit security level; $n = 1024$ provides high security, but at the cost of bandwidth.

An optimal value is $n \approx 700$, resulting in a lattice dimension large enough, yet with moderate bandwidth requirements. This choice is reflected in proposals like Kyber [19], NTRUEncrypt [39], NTRU-KEM [42], SABER [28] and more. Kyber [19] and SABER [28] use modules of rank $k = 3$ over $\mathbb{Z}_q[x]/(x^{256} + 1)$, so that the underlying module lattice problem (conjecturally) relates to a lattice problem of dimension $n = 3 * 256 = 768$, allowing some additional flexibity via varying k. NTRUEncrypt uses the reduction polynomial $x^n - 1$, and its underlying problem remains hard for this ring. However, as suggested by [57, p. 6], the decisional RLWE problem over this ring is easy.

1.3 Parameter Selection: Prior Work

Applebaum et al. [5] showed that the LWE secrets $\boldsymbol{s}$ can be drawn according to the same distribution as the errors without impacting hardness. [22] further showed that LWE with binary errors is also provably hard. When used to construct actual schemes, such *small* secrets improve computational performance and operational correctness. This motivated [24,25] to propose schemes where the LWE secrets are sparse and ternary. NTRUPrime [14] also utilizes rounding and sparse-ternary secrets. The decryption/decapsulation failure probability can

be further reduced by using error-correcting codes. The analysis in [36] shows that the usage of error correction can result in significant increases in estimated bit-security and significantly reduced communication overhead.

A final aspect to consider refers to public parameters such as the matrix $\boldsymbol{A}$. Some schemes propose static parameters for improving performance [20]. Other proposals [18,3,19] rather argue that such parameters should be variable, e.g., in order to prevent pre-computation and backdoor attacks. The overhead for generating a new $\boldsymbol{A}$ can be high, particularly in the case of unstructured lattice-based schemes that must generate n^2 elements. We offer solutions to this problem in our work.

1.4 Our Contributions and Structure of This Paper

We present *Round5*, consisting of algorithms for an IND-CPA secure key encapsulation mechanism Round5.KEM and an IND-CCA secure public key encryption scheme Round5.PKE. Our main contributions are:

- **Unified Design.** Round5 instantiates *the LWR problem and the RLWR problem in a seamless manner*, through its reliance on the General Learning with Rounding (GLWR) problem (Section 2.3). The same algorithm(s) can instantiate LWR or RLWR depending on the input parameters, while also supporting both IND-CPA and IND-CCA security.

- **Prime Cyclotomic Ring.** Like [64], NTRU-KEM [42], and [65], Round5 uses as reduction polynomial the $(n+1)$-th cyclotomic polynomial $\Phi_{n+1}(x) = x^n + \cdots + x + 1$, for $n+1$ a prime. The choice of $n = 1$ leads to a non-ring configuration; taking $n > 1$ leads to a ring configuration. Compared with the power-of-2 cyclotomic polynomial $x^n + 1$, our ring choice offers a *larger design space*, allowing better parameter optimization. We further require that $\Phi_{n+1}(x)$ is irreducible modulo two, so that we hedge against possible vulnerabilities in power-of-2 cyclotomic rings [13,14]. In addition and importantly, as shown in [53], decisional RLWE over this ring remains hard for any modulus, including the power-of-2 moduli q, p as used in Round5.

- **Designed for Performance.** Round5 is designed to be *highly efficient*: its use of rounding firstly requires less randomness; further, combined with our tight analysis of the security of LWR for our chosen secret-key distribution, and highly optimized parameter selection, it results in some of the smallest key and ciphertext sizes in lattice-based cryptography [38]. As the moduli q and p are powers of two, modular operations can be implemented efficiently. Furthermore the use of ternary secrets and error correction codes leads to significant reduction in failure rate without compromising performance or security, improving Round5 parameters even further.

Round5 combines design features from the NIST PQC proposals "Round2" [6] (specifically, the unified design based on GLWR, the use of rounding, the secret-key distribution, the use of the prime cyclotomic polynomials, security and failure rate analysis) and "Hila5" [60] (specifically, the use of forward error correction).

The rest of the paper is organized as follows: In Section 2, we present preliminaries, notation, and the hard problem underlying the security of Round5. In Section 3, Round5.KEM and Round5.PKE and their internal building blocks are specified. Section 4 analyzes the correctness of Round5. In Section 5, the IND-CPA security of Round5.KEM and the IND-CCA Security of Round5.PKE are detailed. Section 6 presents concrete security analysis with respect to known attacks against Round5. Section 7 presents Round5 configuration parameters, performance and comparison with other schemes, followed by conclusions in Section 8.

2 Preliminaries

Let $\mathbb{Z}$ and $\mathbb{Z}_q$ denote respectively the ring of integers, and for an integer $q \geq 1$ the quotient ring $\mathbb{Z}/q\mathbb{Z}$. For a set A, we denote by $a \xleftarrow{\$} A$ that a is drawn uniformly from A. If χ is a probability distribution, then $a \leftarrow \chi$ means that a is drawn at random according to the probability distribution χ. Logarithms are in base 2, unless specified otherwise. All vectors are column vectors. Bold upper case letters are matrices. The transpose of a vector $\boldsymbol{v}$ or a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{v}^T$ or $\boldsymbol{A}^T$. For $x \in \mathbb{Q}$, we denote by $\lfloor x \rfloor$ and $\lfloor x \rceil$ rounding downwards to the next smaller integer and rounding to the closest integer (with rounding up in case of a tie) respectively. For a positive integer α and $x \in \mathbb{Q}$, we define $\{x\}_\alpha$ as the unique element x' in the interval $(-\alpha/2, \alpha/2]$ satisfying $x' \equiv x \pmod{\alpha}$. We define $\langle x \rangle_\alpha$ as the unique element x' in the interval $[0, \alpha)$ for which $x \equiv x' \pmod{\alpha}$.

2.1 Underlying Ring

Let $n + 1$ be prime. We denote by $\mathcal{R}_n$ the polynomial ring $\mathbb{Z}[x]/(\Phi_{n+1}(x))$, for the $(n+1)$-th cyclotomic polynomial $\Phi_{n+1}(x) = x^n + x^{n-1} + \cdots + x + 1$. When n equals 1, then $\mathcal{R}_n = \mathbb{Z}$. For each positive integer a, we write $\mathcal{R}_{n,a}$ for the set of polynomials of degree less than n with all coefficients in $\mathbb{Z}_a$. We call a polynomial in $\mathcal{R}_n$ *ternary* if all its coefficients are 0, 1 or -1. Throughout this document, regular font letters denote elements from $\mathcal{R}_n$, and bold lower case letters represent vectors with coefficients in $\mathcal{R}_n$.

2.2 Distributions

For each $v \in \mathcal{R}_n$, the Hamming weight of v is defined as its number of non-zero coefficients. The Hamming weight of a vector in $\mathcal{R}_n^k$ equals the sum of the Hamming weights of its components. We denote with $\mathcal{H}_{n,k}(h)$ the set of all vectors $\boldsymbol{v} \in \mathcal{R}_n^k$ of ternary polynomials of Hamming weight h, where $h \leq nk$. By considering the coefficients of a polynomial in $\mathcal{R}_n$ as a vector of length n, a polynomial in $\mathcal{H}_{n,k}(h)$ corresponds to a ternary vector of length nk with non-zeroes in h positions, so that $\mathcal{H}_{n,k}(h)$ has $\binom{nk}{h}2^h$ elements. When $k = 1$, we omit

it from the notation, and $\mathcal{H}_n(h)$ denotes the set of all ternary polynomials in $\mathcal{R}_n$ of Hamming weight h, corresponding to the set of all vectors $\boldsymbol{v} \in \{-1, 0, 1\}^n$ with Hamming weight h. Secret keys in Round5 consist of matrices that contain (column) vectors that are distributed according to some distribution χ_S defined over the set $\mathcal{H}_{n,d/n}(h)$.

2.3 Hardness Assumption: The General Learning with Rounding problem

The problem underlying the security of Round5 is the General Learning with Rounding (GLWR) Problem, formally defined as follows:

Definition 1 (General LWR (GLWR)). *Let d, n, p, q be positive integers such that $q \geq p \geq 2$, and $n \in \{1, d\}$. Let χ_S be a probability distribution on $\mathcal{R}_n^{d/n}$.*

The search version of the GLWR problem $\mathtt{sGLWR}_{d,n,m,q,p}(\chi_S)$ is as follows: Given m samples of the form $(\boldsymbol{a}_i, b_i = \left\langle \left\lfloor \frac{p}{q} \cdot \left\langle \boldsymbol{a}_i^T \boldsymbol{s} \right\rangle_q \right\rceil \right\rangle_p)$ with $\boldsymbol{a}_i \in \mathcal{R}_{n,q}^{d/n}$ for $1 \leq i \leq m$ and a fixed $\boldsymbol{s} \leftarrow \chi_S$, recover $\boldsymbol{s}$.

The decision version of the GLWR problem $\mathtt{dGLWR}_{d,n,m,q,p}(\chi_S)$ is to distinguish between the uniform distribution for the samples $(\boldsymbol{a}_i, b_i)$ on $\mathcal{R}_{n,q}^{d/n} \times \mathcal{R}_{n,p}$ and m samples from the distribution $(\boldsymbol{a}_i, b_i = \left\langle \left\lfloor \frac{p}{q} \cdot \left\langle \boldsymbol{a}_i^T \boldsymbol{s} \right\rangle_q \right\rceil \right\rangle_p)$ with $\boldsymbol{a}_i \xleftarrow{\$} \mathcal{R}_{n,q}^{d/n}$ for $1 \leq i \leq m$ for some secret $\boldsymbol{s} \leftarrow \chi_S$ common to all i.

When the secret distribution χ_S is the uniform one over $\mathcal{R}_{n,q}^{d/n}$, it is omitted from notation. When the distribution χ_S is set to $\mathcal{U}\left(\mathcal{H}_{1,d}(h)\right)$, we denote the specialized version $\mathtt{GLWR}_{d,n,m,q,p}\left(\mathcal{U}\left(\mathcal{H}_{n,d/n}(h)\right)\right)$ of the (decision) GLWR problem, as $\mathtt{dGLWR}_{\mathrm{spt}}$ (spt denoting sparse-ternary secrets) for brevity. When $n = 1$, the above is equivalent to the LWR problem $\mathtt{LWR}_{\mathrm{spt}}$ with dimension d, large modulus q, rounding modulus p, and sparse-ternary secrets. The hardness of the LWR problem has been studied in [10,4,16,9] and established based on the hardness of the Learning with Errors (LWE) problem [56]. The most recent reductions are due to [9, Theorem 6.4] (that preserves the dimension n between the two problems) and [16, Theorem 3] (that preserves the number of samples m). We extend the above work by proving that there exists a *polynomial-time reduction* from the (decision) Learning with Errors (LWE) problem with secrets chosen uniformly from $\mathbb{Z}_q^d$ and errors chosen from a Gaussian distribution, to (decision) $\mathtt{LWR}_{\mathrm{spt}}$, for appropriate parameters. A full statement of the reduction and its proof can be found in Section 5.3.

When $n = d \geq 1$ is such that $n + 1$ is prime, and $\mathcal{R}_{n,q} = \mathbb{Z}_q[x]/\left(\Phi_{n+1}(x)\right)$, the $\mathtt{dGLWR}_{\mathrm{spt}}$ problem is equivalent to the Ring LWR problem $\mathtt{RLWR}_{\mathrm{spt}}$ defined on $\Phi_{d+1}(x)$, dimension d, large modulus q, rounding modulus p, and sparse-ternary secrets. We are only aware of a reduction from Decision-RLWE to Decision-RLWR due to [10, Theorem 3.2] which requires the underlying ring and secret to be the same for the two problems, that the RLWE noise is sampled from any (balanced) distribution in $\{-B, \ldots, B\}$, and q is super-polynomial in n, i.e.,

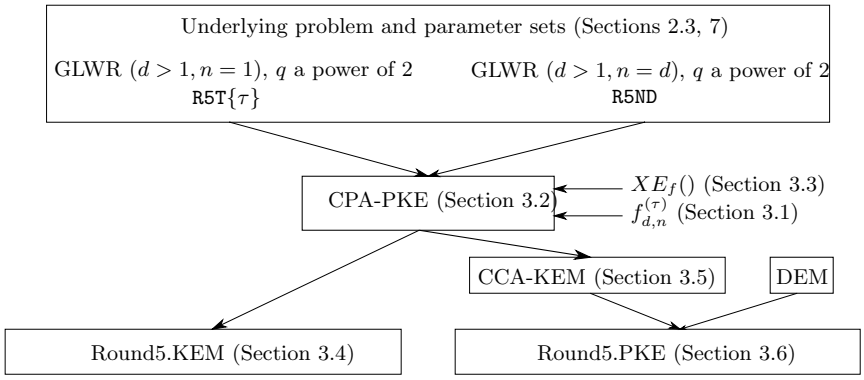

Fig. 1: Overview of Round5.

$q \geq pBn^{\omega(1)}$. The last condition may be too restrictive for practical schemes. Hence, although [10, Theorem 3.2] is relevant for the (asymptotic) security of our ring-based instantiations, it remains to be seen whether it can be improved and generalized to be directly applicable.

We define the GLWR oracle $O_{m,\chi_S,\boldsymbol{s}}$ for a secret distribution χ_S that returns m GLWR samples as follows:

$$O_{m,\chi_S,\boldsymbol{s}} : \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_{n,q}^{m \times d/n},\ \boldsymbol{s} \leftarrow \chi_S;\ \text{return}\ \left(\boldsymbol{A}, \left\langle \left\lfloor \frac{p}{q} \cdot \langle \boldsymbol{A}\boldsymbol{s} \rangle_q \right\rceil \right\rangle_p\right) \tag{1}$$

The $\mathtt{dGLWR}_{\mathrm{spt}}$ problem is to distinguish between the distributions $(\mathcal{U}(\mathcal{R}_{n,q}^{d/n}) \times \mathcal{U}(\mathcal{R}_{n,p}))^m$ and $O_{m,\chi_S,\boldsymbol{s}}$, with $\boldsymbol{s}$ common to all m samples and $\chi_S := \mathcal{U}(\mathcal{H}_{n,d/n}(h))$. For an adversary $\mathcal{A}$, we define

$$\begin{aligned} \mathrm{Adv}^{\mathtt{dGLWR}_{\mathrm{spt}}}_{d,n,m,q,p}(\mathcal{A}) = |\Pr\left[\mathcal{A}(\boldsymbol{A},\boldsymbol{b}) = 1 \mid (\boldsymbol{A},\boldsymbol{b}) \xleftarrow{\$} O_{m,\chi_S,\boldsymbol{s}}\right] - \\ \Pr\left[\mathcal{A}(\boldsymbol{A},\boldsymbol{b}) = 1 \mid \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_{n,q}^{m \times d/n}, \boldsymbol{b} \xleftarrow{\$} \mathcal{R}_{n,p}^{m}\right]| \end{aligned} \tag{2}$$

For an extended form of the decision-GLWR problem with the secret in form of a matrix consisting of $\overline{n}$ independent secret vectors, we define a similar oracle $O_{m,\chi_S,\overline{n},\boldsymbol{S}}$ as follows:

$$O_{m,\chi_S,\overline{n},\boldsymbol{S}} : \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_{n,q}^{m \times d/n},\ \boldsymbol{S} \leftarrow (\chi_S)^{\overline{n}};\ \text{return}\ \left(\boldsymbol{A}, \left\langle \left\lfloor \frac{p}{q} \cdot \langle \boldsymbol{A}\boldsymbol{S} \rangle_q \right\rceil \right\rangle_p\right) \tag{3}$$

The advantage of an adversary for this extended form of the decision-GLWR problem is defined in a similar manner as above.

3 Round5

Figure 1 provides an overview of Round5, and shows the different configurations of the schemes based on the underlying GLWR problem. Externally Round5 consists of:

– **Round5.KEM** (Section 3.4), an IND-CPA secure key encapsulation mechanism (KEM).
– **Round5.PKE** (Section 3.6), an IND-CCA secure public key encryption algorithm (PKE).

Section 3.1 describes options for generating Round5's public parameter $\boldsymbol{A}$. The public schemes are derived from internal building blocks; We first describe *CPA-PKE* (Section 3.2), and the error correction code *XEf* (Section 3.3), leading to Round5.KEM.

We then apply a KEM variant [40] of the Fujisaki-Okamoto transform to CPA-PKE, to obtain a key encapsulation mechanism *CCA-KEM* (Section 3.5), that is IND-CCA secure in the classical and quantum ROM model [12,17]. Round5.PKE is obtained by combining CCA-KEM with a secure one-time symmetric-key encryption scheme. Details of the IND-CPA and IND-CCA [11] security properties of CPA-PKE, CCA-KEM, and Round5.PKE are discussed in Section 5.

3.1 Internal building block: Definitions of $\boldsymbol{f}_{\boldsymbol{d,n}}^{(\boldsymbol{\tau})}(\boldsymbol{\sigma})$

Round5.KEM and Round5.PKE require the generation of the GLWR public parameter $\boldsymbol{A} \in \mathcal{R}_{n,q}^{d/n \times d/n}$. In order to make the choice for $\boldsymbol{A}$ explicit, a seed σ is used, as well as a description of how to construct $\boldsymbol{A}$ from $\sigma \in \{0,1\}^{\kappa}$. Round5 provides three functions for $\boldsymbol{f}_{d,n}^{(\tau)}(\sigma)$, with $\tau \in \{0,1,2\}$, for doing so. We note that for $\tau \in \{1,2\}$, the mapping $\boldsymbol{f}_{d,n}^{(\tau)}$ is only applied if $n = 1$, i.e., in the non-ring case. In the definitions given below, $s' = H(0x0000|s)$ indicates the derivation of seed s' from seed s by using padding $0x0000$ and applying a hash function H. With $PRNG(s)[k]$ we mean the k-th element of the deterministic pseudorandom number generator $PRNG$ applied to seed s.

0. $\boldsymbol{f}_{\boldsymbol{d,n}}^{(\boldsymbol{0})}$: A new $\boldsymbol{A} \in \mathcal{R}_{n,q}^{d/n \times d/n}$ is derived by a PRNG from a seed σ for each protocol instantiation. We define $\sigma_0 = H(0x0000|\sigma)$. If $n = 1$, then $\boldsymbol{A} \in \mathbb{Z}_q^{d \times d}$, and for $0 \leq i, j \leq d-1$, $a_{i,j} = PRNG(\sigma_0)[id + j]$. This is the same generation of the public parameter in [18]. If $n = d$, then $A = \sum_{k=0}^{d-1} a_k x^k \in \mathbb{Z}_q[x]$, and $a_k = PRNG(\sigma_0)[k]$ for $0 \leq k \leq d-1$, like in [3]. In both cases, $PRNG$ outputs symbols in $\{0, 1, \ldots, q-1\}$.
1. $\boldsymbol{f}_{\boldsymbol{d,1}}^{(\boldsymbol{1})}$: In this instantiation, which is only used for the case $n = 1$, a new $\boldsymbol{A}$ is derived using permutations on a long-term matrix $\boldsymbol{A}^{\text{master}} \in \mathbb{Z}_q^{d \times d}$. The permutation is computed by applying a PRNG to seed $\sigma_1 = H(0x0001|\sigma)$, as follows. For $0 \leq i \leq d-1$, we have $o_i = PRNG(\sigma_1)[i]$. For $0 \leq i, j \leq d-1$, $a_{i,j} = a^{\text{master}}_{i,(j+o_i)(mod \quad d)}$. In this case, $PRNG$ outputs symbols in $\{0, 1, \ldots, d-1\}$.
2. $\boldsymbol{f}_{\boldsymbol{d,1}}^{(\boldsymbol{2})}$ computes the elements in $\boldsymbol{A}$ by applying a permutation to a set of $L = q$ elements. The permutation is computed as follows:
$\sigma_1 = H(0x0001|\sigma)$, $o_i = PRNG(\sigma_1)[i]$ for $0 \leq i \leq d-1$. Here $PRNG$ outputs symbols from $\{0, 1, \ldots, L-1\}$.

The entries of the matrix $\boldsymbol{A}$ are obtained as
$a_{i,j} = PRNG_1(\sigma_0)[(j + o_i)(mod \quad L)]$, where $\sigma_0 = H(0x0000|\sigma)$. Here $PRNG_1$ outputs symbols from $\{0, 1, \ldots, q-1\}$.

The functions $\boldsymbol{f}_{d,n}^{(\tau)}$ stop both backdoor-like and precomputation attacks. Section 5.1 contains a discussion on the role of $\boldsymbol{f}_{d,n}^{(0)}$ and $\boldsymbol{f}_{d,n}^{(1)}$ in the provable security of Round5.

3.2 Internal building block: CPA-PKE

CPA-PKE consists of algorithms 1 (key-generation), 2 (encryption) and 3 (decryption), and various cryptosystem parameters, *viz* positive integers n, d, h, p, q, t, B, $\overline{n}$, $\overline{m}$, μ, y, and a security parameter κ. In the proposed configurations, $n \in \{1, d\}$, and q, p, t are powers of 2, such that $2^B|t|p|q$. It is required that $\mu \leq \overline{n} \cdot \overline{m} \cdot n$ and that $\mu B \geq \kappa$. The function $\text{Sample}_\mu : \boldsymbol{C} \in \mathcal{R}_{n,p}^{\overline{n} \times \overline{m}} \rightarrow \mathbb{Z}_p^\mu$ outputs the values of μ of the $\overline{n} \cdot \overline{m} \cdot n$ polynomial coefficients present in $\boldsymbol{C}$. For $n = d$, the parameters $\overline{n} = \overline{m} = 1$, then Sample_μ picks up the μ coefficients of highest order. If $n = 1$, Sample_μ picks up the last μ entries of the vector obtained by serializing the matrix row by row. CPA-PKE.Keygen generates a secret matrix $\boldsymbol{S}$ with ternary columns drawn independently according to a distribution χ_S with support on $\left(\mathcal{H}_{n,d/n}(h)\right)^{1 \times \overline{n}}$.

The integer y is the index for an error correction code $Y_y \subset \mathbb{Z}_{2^B}^\mu$. We have an encoding function $ECC\_Enc_y : \{0,1\}^\kappa \rightarrow Y_y$ and a decoding function $ECC\_Dec_y : \mathbb{Z}_{2^B}^\mu \rightarrow \{0,1\}^\kappa$ such that for each $m \in \{0,1\}^\kappa$:

$$ECC\_Dec_y(ECC\_Enc_y(m)) = m. \tag{4}$$

Algorithm CPA-PKE.Encrypt employs a deterministic function f_R for generating a secret matrix $\boldsymbol{R} \in \left(\mathcal{H}_{n,d/n}(h)\right)^{1 \times \overline{m}}$ from an input ρ. If ρ is uniformly distributed, each column of $f_R(\rho)$ is distributed according to χ_S. Defining ρ as an explicit input to CPA-PKE.Encrypt allows us to reuse this *same* algorithm as a building block for both IND-CPA and IND-CCA secure cryptographic constructions. Furthermore, CPA-PKE uses five rounding constants, $\boldsymbol{H}_1$ up to $\boldsymbol{h_5}$. These, combined with rounding downwards, implement all of the actual rounding operations in its algorithms. The matrix $\boldsymbol{H}_1 \in \mathcal{R}_{n,q}^{d/n \times \overline{n}}$ has all coefficients equal to $q/2p$. This constant leads to rounding to the closest integer, exactly as done in Round2 [6], as by definition for any $x \in \mathbb{Z}_q$, $\lfloor (p/q) \cdot \{x + (q/2p)\} \rfloor \equiv \lfloor (p/q) \cdot x \rceil$. The coefficients of $\boldsymbol{H}_2 \in \mathcal{R}_{n,q}^{d/n \times \overline{m}}$ and $\boldsymbol{h}_3 \in \mathbb{Z}_q^\mu$ all are equal to $q/2z$, for $z = \max(p, tq/p)$. This ensures that Round5's ciphertext $(\boldsymbol{U}, \boldsymbol{v})$ is provably pseudorandom under the GLWR assumption. Details are provided in the proof of IND-CPA security for CPA-PKE and Round5.KEM (Section 5). All coefficients of $\boldsymbol{H}_4 \in \mathcal{R}_{n,q}^{d/n \times \overline{m}}$ are equal to $\left(\frac{q}{2p} - \frac{q}{2z}\right)$. Finally, all coefficients of $\boldsymbol{h}_5 \in \mathbb{Z}_q^\mu$ are equal to $\left(\frac{q^2}{2pz} - \frac{q}{2t} - \frac{q}{2^{B+1}}\right)$.

Algorithm 1: CPA-PKE.Keygen()

Choose $\tau \in \{0, 1, 2\}$

$\sigma \xleftarrow{\$} \{0,1\}^{\kappa}$

$\boldsymbol{A} = \boldsymbol{f}_{d,n}^{(\tau)}(\sigma)$

$\boldsymbol{S} \leftarrow \chi_S^n$

$\boldsymbol{B} = \langle \left\lfloor \frac{p}{q} \cdot \langle \boldsymbol{AS} + \boldsymbol{H_1} \rangle_q \right\rfloor \rangle_p$

$pk = (\tau, \sigma, \boldsymbol{B})$

$sk = \boldsymbol{S}$

return (pk, sk)

Algorithm 2: CPA-PKE.Encrypt(pk, m, ρ)

$\boldsymbol{A} = \boldsymbol{f}_{d,n}^{(\tau)}(\sigma)$

$\boldsymbol{R} = f_R(\rho)$

$\boldsymbol{U} = \langle \left\lfloor \frac{p}{q} \cdot \langle \boldsymbol{A}^T \boldsymbol{R} + \boldsymbol{H_2} \rangle_q \right\rfloor \rangle_p$

$\boldsymbol{v} = \langle \left\lfloor \frac{t}{p} \cdot \left\langle \text{Sample}_{\mu}(\boldsymbol{B^T R}) + \boldsymbol{h}_3 \right\rangle_p \right\rfloor + \frac{t}{2^B} ECC\_Enc_y(m) \rangle_t$

$c = (\boldsymbol{U}, \boldsymbol{v})$

return c

Algorithm 3: CPA-PKE.Decrypt(sk, c)

$\boldsymbol{v}_q = \frac{q}{t} \boldsymbol{v}$

$z = \left\langle \left\lfloor \frac{2^B}{q} \left\langle \boldsymbol{v}_q - \boldsymbol{h}_5 - \text{Sample}_{\mu} \left(\left\langle \boldsymbol{S}^T \left(\frac{q}{p} \cdot \boldsymbol{U} + \boldsymbol{H_4} \right) \right\rangle_q \right) \right\rangle_q \right\rfloor \right\rangle_{2^B}$

$\hat{m} = ECC\_Dec_y(z)$

return $\hat{m}$

3.3 Error correction

Round5 has a trade-off between decryption error probability and security: the smaller $\frac{p}{q}$, the higher both the security and the failure probability. In [36], it is analyzed how error-correcting codes can be used the enhance the error resilience of protocols like NewHope, Frodo and Kyber, and it is shown that the usage of error correcting codes can significantly increase the estimated bit-security and decrease the communication overhead. Round5 uses an f-bit error correcting block code XEf to decrease the failure rate. The code is built using the same strategy as codes used by TRUNC8 [59] (2-bit correction) and Hila5 [60] (5-bit correction).

The XEf code is described by $2f$ "registers" R_i of size $|R_i| = l_i$. We view the κ-bits payload block m as a binary polynomial $m_{\kappa|-1}x^{|\kappa|-1} + \cdots + m_1 x + m_0$ of length κ. Registers are defined via cyclic reduction

$$R_i = m \bmod x^{l_i} - 1, \tag{5}$$

or equivalently by

$$r_{(i,j)} = \sum_{k \equiv j \bmod l_i} m_k \tag{6}$$

where $r_{(i,j)}$ is bit j of register R_i. A transmitted message consists of the payload m concatenated with register set r (a total of $\mu = \kappa + \sum l_i$ bits).

Upon receiving a message $(m' \mid r')$ one computes the register set r'' corresponding to m' and compares it to the received register set r' – that may also have errors. Errors are in coefficients m'_j where there is parity disagreements $r'_{(i,j \mod l_i)} \neq r''_{(i,j \mod l_i)}$ for multitude of registers R_i. We use a majority rule and flip bit m'_j if

$$\sum_{i=1}^{2f} \left(\left(r'_{(i,\langle j \rangle_{l_i})} - r''_{(i,\langle j \rangle_{l_i})} \right) \bmod 2 \right) \geq f + 1 \tag{7}$$

where the sum is taken as the number of disagreeing register parity bits at j.

It is easy to show that if all length pairs satisfy $\mathrm{lcm}(l_i, l_j) \geq \kappa$ when $i \neq j$, then this code always corrects at least f errors. Typically one chooses coprime lengths $l_1 < l_2 < \cdots < l_{2f}$ so that $l_1 l_2 \geq \kappa$.

The main advantage of XEf codes is that they avoid table look-ups and conditions altogether and are therefore resistant to timing attacks.

3.4 Round5.KEM

Round5.KEM, an IND-CPA-secure [11] key encapsulation method, builds on CPA-PKE (Section 3.2) and consists of algorithms 4, 5 and 6. It furthermore uses the hash function $H : \{0,1\}^* \to \{0,1\}^\kappa$, and a function $\texttt{bin}(\cdot)$ that expands an input into its full binary representation (concatenated in case of multiple inputs); the input and output types of `bin` shall be evident from the context.

Algorithm 4: Round5.KEM.Keygen()

1 $(pk, sk) = \text{CPA-PKE.Keygen}()$
2 **return** (pk, sk)

Algorithm 5: Round5.KEM.Encapsulate(pk)

1 $m \xleftarrow{\$} \{0,1\}^{\kappa}$
2 $\rho \xleftarrow{\$} \{0,1\}^{\kappa}$
3 $c = \text{CPA-PKE.Encrypt}(pk, m, \rho)$
4 $K = H(m, \texttt{bin}(c))$
5 **return** (c, K)

Algorithm 6: Round5.KEM.Decapsulate(sk, c)

1 $m = \text{CPA-PKE.Decrypt}(sk, c)$
2 $K = H(m, \texttt{bin}(c))$
3 **return** K

3.5 Internal building block: CCA-KEM

CCA-KEM, a building block for Round5.PKE, consists of the algorithms 7, 8, 9, and several system parameters and functions in addition to those from CPA-PKE and Round5.KEM. In addition to the hash function H from Round5.KEM, it uses another hash function $G : \{0,1\}^* \to \{0,1\}^{\kappa} \times \{0,1\}^{\kappa} \times \{0,1\}^{\kappa}$. CCA-KEM is actively secure as it is obtained by application of the Fujisaki-Okamoto transform [40] on CPA-PKE, similarly as in [19, Sec. 4]. On decapsulation failure, i.e. if the condition in line 4 of Algorithm 9, a pseudorandom key is returned, causing later protocol steps to fail implicitly. Explicit failure notification would complicate analysis, especially in the quantum random oracle (QROM) case.

3.6 Round5.PKE

The IND-CCA [55] public key encryption scheme Round5.PKE consists of algorithms 10, 11 and 12. Round5.PKE combines CCA-KEM with a data encapsulation mechanism (DEM), in the canonical way proposed by Cramer and Shoup [27]. CCA-KEM is used to encapsulate a key K that is then used by the DEM to encrypt an arbitrary-length plaintext, optionally adding integrity protection. In decryption, CCA-KEM is used to decapsulate K, which is then used by the DEM to decrypt and authenticate the plaintext.

4 Correctness of Round5

In this section, the decryption failure behavior of CPA-PKE is analyzed. In decryption, the vector $\boldsymbol{z} = \langle \lfloor \frac{2^B}{q} \boldsymbol{\zeta} \rfloor \rangle_{2^B}$ is computed, where

$$\boldsymbol{\zeta} = \left\langle \boldsymbol{v}_q - \boldsymbol{h}_5 - \text{Sample}_{\mu}\left(\left\langle \boldsymbol{S}^T \left(\frac{q}{p}\boldsymbol{U} + \boldsymbol{H}_4 \right) \right\rangle_q \right) \right\rangle_q .$$

Algorithm 7: CCA-KEM.Keygen()

$(pk, sk_{CPA-PKE})$ = CPA-PKE.Keygen()
$z \xleftarrow{\$} \{0,1\}^{\kappa}$
$sk = (sk_{CPA-PKE}, z, pk)$
return (pk, sk)

Algorithm 8: CCA-KEM.Encapsulate(pk)

$m \xleftarrow{\$} \{0,1\}^{\kappa}$
$(L, \rho, g) = G(m, \texttt{bin}(pk))$
$(\boldsymbol{U}, \boldsymbol{v})$ = CPA-PKE.Encrypt(pk, m, ρ)
$c = (\boldsymbol{U}, \boldsymbol{v}, g)$
$K = H(L, \texttt{bin}(\boldsymbol{U}, \boldsymbol{v}), g)$
return (c, K)

Algorithm 9: CCA-KEM.Decapsulate(sk, c)

m' = CPA-PKE.Decrypt$(sk_{CPA-PKE}, (\boldsymbol{U}, \boldsymbol{v}))$
$(L', \rho', g') = G(m', \texttt{bin}(pk))$
$(\boldsymbol{U'}, \boldsymbol{v'})$ = CPA-PKE.Encrypt(pk, m', ρ')
if $(\boldsymbol{U'}, \boldsymbol{v'}, g') = (\boldsymbol{U}, \boldsymbol{v}, g)$ **then**
    return $K = H(L', \texttt{bin}(\boldsymbol{U}, \boldsymbol{v}), g)$
else
    return $K = H(z, \texttt{bin}(\boldsymbol{U}, \boldsymbol{v}), g)$
end if

Algorithm 10: Round5.PKE.Keygen()

(pk, sk) = CCA-KEM.Keygen()
return (pk, sk)

Algorithm 11: Round5.PKE.Encrypt(pk, M)

$(c1, K)$ = CCA-KEM.Encapsulate(pk)
$(clen, c2)$ = DEM(K, M)
$c = (c1, clen, c2)$
return c

Algorithm 12: Round5.PKE.Decrypt(sk, c)

K = CCA-KEM.Decapsulate$(sk, c1)$
$(mlen, M) = \text{DEM}^{-1}(K, c2)$
return $(mlen, M)$

As a first step, we derive a sufficient condition so that $\boldsymbol{z}$ and $\boldsymbol{x} = ECC_y(m)$ agree in a given position, where $\boldsymbol{x}$ is considered as a vector of (κ/B) B-bits symbols.

We have that $\boldsymbol{v} \equiv \lfloor \frac{t}{p} \langle \text{Sample}_\mu(\boldsymbol{B}^T\boldsymbol{R} + \boldsymbol{h}_3) \rangle_p \rfloor + \frac{t}{2^B}\boldsymbol{x} = \frac{t}{p}\langle \text{Sample}_\mu(\boldsymbol{B}^T\boldsymbol{R} + \boldsymbol{h}_3)\rangle_p - \frac{t}{p}\boldsymbol{I}_v + \frac{t}{2^B}\boldsymbol{x} \pmod t$, where $\frac{t}{p}\boldsymbol{I}_v$ is the effect of rounding, with each component of $\boldsymbol{I}_v$ in $\mathbb{Z}_{p/t}$. Similarly, $\boldsymbol{B} = \langle (p/q)\,(\boldsymbol{AS} + \boldsymbol{H_1}) - (p/q)\boldsymbol{I}_B \rangle_p$, and $\boldsymbol{U} = \langle (p/q)(\boldsymbol{A}^T\boldsymbol{R} + \boldsymbol{H_2}) - (p/q)\boldsymbol{I}_U \rangle_p$, with all components of $\boldsymbol{I}_B$ and $\boldsymbol{I}_U$ in $\mathbb{Z}_{q/p}$. We thus have that
$\boldsymbol{\zeta} = \langle \frac{q}{2^B}\boldsymbol{x} + \frac{q}{p}\boldsymbol{h_3} - \boldsymbol{h_5} - \frac{q}{p}\boldsymbol{I}_v + \text{Sample}_\mu(\frac{q}{p}\langle \boldsymbol{B}^T\boldsymbol{R}\rangle_p - \langle \boldsymbol{S}^T(\frac{q}{p}\boldsymbol{U} + \boldsymbol{H}_4)\rangle_q)\rangle_q$.
As $\boldsymbol{z} = \lfloor \frac{2^B}{q}\boldsymbol{\zeta} \rfloor$, it holds that $x_i = z_i$ whenever

$$| \left[\boldsymbol{J}_v + \text{Sample}_\mu \left(\boldsymbol{J}_B^T R - S^T \boldsymbol{J}_U \right) \right]_i | < \frac{q}{2^{B+1}}, \tag{8}$$

where the subscript i means taking the i-th component, $\boldsymbol{J}_v = \frac{q}{p}\boldsymbol{h}_3 - \boldsymbol{h}_5 - \frac{q}{2^{B+1}}\boldsymbol{j} - \frac{q}{p}\boldsymbol{I}_v$, $\boldsymbol{J}_B = \boldsymbol{H}_1 - \boldsymbol{I}_B$ and $\boldsymbol{J}_U = \boldsymbol{H}_2 + \boldsymbol{H}_4 - \boldsymbol{I}_U$. The definitions of $\boldsymbol{h}_3$ and $\boldsymbol{h}_5$ imply that $\boldsymbol{J}_v = \frac{q}{p}(\frac{p}{2t} - \boldsymbol{I}_v)$. As each entry of $\boldsymbol{I}_v$ is in $\mathbb{Z}_{p/t}$, each component of $\boldsymbol{J}_v$ has absolute value at most $\frac{q}{p} \cdot \frac{p}{2t} = \frac{q}{2t}$. As a result, $x_i = z_i$ whenever

$$|[\text{Sample}_\mu(\boldsymbol{J}_B^T\boldsymbol{R} - \boldsymbol{S}^T\boldsymbol{J}_U)]_i| < \Delta := \frac{q}{2^{B+1}} - \frac{q}{2t}. \tag{9}$$

The definitions of $\boldsymbol{H}_1, \boldsymbol{H}_2$ and $\boldsymbol{H}_4$ imply that all entries of $\boldsymbol{J}_B$ and $\boldsymbol{J}_U$ are from the set $I := (-\frac{q}{2p}, \frac{q}{2p}]$. In our analysis, we assume that the entries of $\boldsymbol{J}_B$ and $\boldsymbol{J}_U$ are drawn independently and uniformly from I. Under this assumption, we analyse the probability that the condition in 9 is *not* satisfied.

In the non-ring case, each entry of $\boldsymbol{J}_B^T\boldsymbol{R}$ and of $\boldsymbol{S}^T\boldsymbol{J}_U$ is the inner product of a row of $\boldsymbol{J}_B^T$ (resp. a column of $\boldsymbol{J}_U$) and a ternary vector with $h/2$ entries equal to one and $h/2$ entries equal to minus one. Hence each entry of $\boldsymbol{J}_B^T\boldsymbol{R} - \boldsymbol{S}^T\boldsymbol{J}_U$ is distributed as the sum of h uniform variables on I minus the sum of h uniform variables on I. The latter distribution can easily be computed explicitly. In the ring case, by straightforward calculation,

$$\langle s(x)e(x)\rangle_{\Phi_{n+1}(x)} = \sum_{k=0}^{n-1} d_k(s,e)x^k,$$

where for $0 \le k \le n-2$

$$d_k(s,e) = e_0 s_k + \sum_{j=1}^{k} e_j(s_{k-j} - s_{n-j}) - e_{k+1}s_{n-k-1} +$$

$$\sum_{j=k+2}^{n-1} e_j(s_{n+k+1-j} - s_{n-j}),$$

$$d_{n-2}(s,e) = e_0 s_{n-2} + \sum_{j=1}^{n-2} e_j(s_{n-2-j} - s_{n-j}) - e_{n-1}s_1, \text{ and}$$

$$d_{n-1}(s,e) = e_0 s_{n-1} + \sum_{j=1}^{n-1} e_j (s_{n-1-j} - s_{n-j}).$$

That is, we can write

$$d_k(s,e) = \sum_{j=0}^{n-1} w_{j,k}(s) e_j,$$

where each weighing term $w_{j,k}(s)$ is a single coefficients of s, or the difference of two coefficients of s. In case s is a ternary polynomial, $w_{j,k}(s) \in \{-2,-1,0,1,2\}$. For a ternary polynomial s, integers i,k with $-2 \le i \le 2$ and $0 \le k \le n-1$, we define

$$f_{i,k}(s) = |\{j \mid 0 \le j \le n-1, w_{j,k}(s) = i\}|.$$

With this notation and the above assumptions, the k-th component of the polynomial $j_B(x)r(x) - s(x)j_U(x)$ has the same distribution as

$$Y = \sum_{i=-2}^{2} i \sum_{j=1}^{w_i} X_{i,j}, \tag{10}$$

where each $X_{i,j}$ is a uniformly distributed variable on I, and $w_i = f_{i,k}(s) + f_{i,k}(-r)$. Assuming that the $X_{i,j}$'s in (10) are independent, the mean μ_Y and the variance σ_Y^2 of Y satisfy

$$\mu_Y = \mu \cdot \sum_{i=-2}^{2} w_i \text{ and } \sigma_Y^2 = \sigma^2 \sum_{i=-2}^{2} i^2 w_i,$$

$$\text{where } \mu = \frac{1}{2} \text{ and } \sigma^2 = \frac{1}{12}\left((\frac{q}{p})^2 - 1\right).$$

We approximate the tail distribution of Y by that of a Gaussian distribution with mean μ_Y and variance σ_Y^2, i.e., we approximate

$$\text{Prob}(|Y| \ge \Delta) \;\le\; \text{Prob}\left(|Y - \mu_Y| \ge \Delta - |\mu_Y|\right) \approx$$

$$\text{erfc}\left(\frac{\Delta - |\mu_Y|}{\sqrt{2}\sigma_Y}\right), \text{ where } \text{erfc}(x) := \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt.$$

The Gaussian approximation depends on w_{-2}, w_{-1}, w_1 and w_{-2}. We now given an upper bound on the approximation that depends on one single variable. As each of the h non-zero coefficients of s occurs in at most two weighing coefficients, $\sum_i |i| w_i \le 4h$, and so $|\mu_Y| = |\frac{1}{2}\sum_i i w_i| \le 2h$, and $\sigma_Y^2 = \sigma^2(w_1 + w_{-1} + 4w_2 + 4w_{-2}) \le \sigma^2(4h + 2(w_2 + 2w_{-2}))$, and so

$$\text{prob}(|Y| \ge \tfrac{q}{2^{B+1}} - \tfrac{q}{2t}) \approx u_y(w_2 + w_{-2}), \text{ where} \tag{11}$$

$$u_y(k) = \text{erfc}\left(\frac{\frac{q}{2^{B+1}} - \frac{q}{2t} - 2h}{(2\sigma^2(4h + 2k))^{1/2}}\right).$$

Finally, we approximate the per-symbol failure probability p_f as

$$p_f \approx \sum_k \text{prob}(w_2 + w_{-2} = k) \cdot u_y(k). \tag{12}$$

The secret polynomial s has n terms; $h/2$ of those have value 1, $h/2$ of them have value -1, and the remaining coefficients have value 0. The number of secret polynomials thus equals $\binom{n}{h/2}\binom{n-h/2}{h/2}$. The number of secret polynomials with a one position i and a minus one in position $j \neq i$ equals $\binom{n-2}{h/2-1}\binom{n-2-(h/2-1)}{h/2-1}$. The probability β that a weighing factor equals ± 2 thus equals $\frac{h}{n}\frac{h}{2(n-1)}$, twice the quotient of the two above products of binomial coefficients. We approximate the per-symbol failure probability by

$$p_f \approx \sum_k \binom{2n}{k} \beta^k (1-\beta)^{2n-k} u_y(k), \text{ where } \beta = \frac{h}{n} \cdot \frac{h}{2(n-1)}.$$

If $B = 1$, Round5 employs a code XEf that correct f bit errors. Assuming that bit failures occur independently and with probability p_f, the failure probability after decoding is at most

$$\sum_{j=f+1}^{\mu} \binom{\mu}{j} p_f^j (1-p_f)^{\mu-j}.$$

In case that no error correction is applied, by the union bound, the failure probability after decoding is at most $\mu \cdot p_f$.

5 Provable Security of Round5

In this section, we discuss proofs of security for both Round5 and its underlying hard problems.

We begin by giving an overview of the security reduction for Round5 when replacing the public parameter $\boldsymbol{A}$ sampled from a truly uniform distribution, with one expanded from a short random seed in a pseudorandom fashion in Section 5.1. Section 5.2 gives a proof of IND-CPA security for the Round5 core building block CPA-PKE, following which we sketch the proofs of security for Round5.KEM, CCA-KEM and Round5.PKE.

Finally, in Section 5.3, we give a proof of hardness of Round5's underlying problem – the decision GLWR problem with sparse-ternary secrets, assuming the hardness of decision Learning with Errors with uniform secrets and Gaussian errors [56].

5.1 Deterministic generation of $\boldsymbol{A}$

The General Learning with Rounding (GLWR) public parameter $\boldsymbol{A}$ in Round5 is generated using the function $f_{d,n}^{(\tau)}$ from a short random seed (see Section 3.1). The core component in $f_{d,n}^{(\tau)}$ responsible for deterministically expanding this short random seed into a longer random sequence is either AES256 [34] or SHAKE256 [35]. In order to relate Round5's security to the hardness of the GLWR problem, we reuse Naehrig *et al.*'s argument in [51] to argue that we

can replace a uniformly sampled matrix $\boldsymbol{A} \in \mathcal{R}_{n,q}^{d/n \times d/n}$ with matrices sampled according to Round5's key-generation algorithm, for both of the above two algorithms, while considering a realistic adversary with access to the seed. The proof for both the cases of AES256 and SHAKE256 proceeds by using the notion of indifferentiability [49,26, Def. 3], in exactly the same manner as in [51, Sec. 5.1.4].

In the case of SHAKE256, the proof of security applies directly to the instantiations $f_{d,n}^{(0)}$ and $f_{d,n}^{(1)}$. In case of AES256, it holds directly for the instantiation $f_{d,n}^{(0)}$, and also for $f_{d,n}^{(1)}$ when the function permutes complete AES blocks. We refer to [51, Sec. 5.1.4] for details.

5.2 Provable security of CPA-PKE, Round5.KEM, CCA-KEM and Round5.PKE

The following theorem proves the IND-CPA security of the Round5 building block CPA-PKE, under the decision-GLWR assumption with sparse-ternary secrets.

Theorem 1 *If $f_n : \{0,1\}^{\mu B} \to \mathcal{R}_{n,q}^{d/n \times d/n}$ is a secure mapping, f_R has output indistinguishable from $(\chi_S)^{\overline{m}}$, then CPA-PKE is IND-CPA secure under the hardness assumption of the decision-GLWR problem with sparse-ternary secrets, assuming $t|p|z|q$ for $z = \max(p, tq/p)$. More precisely, for every IND-CPA adversary $\mathcal{A}$, if $Adv_{CPA\text{-}PKE}^{IND\text{-}CPA}(\mathcal{A})$ is the advantage in winning the IND-CPA game, then there exist adversaries $\mathcal{D}$ and reduction algorithms $\mathcal{C}', \mathcal{E}'$ such that*

$$\begin{aligned} Adv_{PKE}^{IND\text{-}CPA}(\mathcal{A}) \leq \overline{n} \cdot Adv_{d,n,\frac{d}{n},q,p}^{\mathtt{dGLWR}_{spt}}(\mathcal{A} \circ \mathcal{C}') + Adv^{f_R}(\mathcal{D}) + \\ \overline{m} \cdot Adv_{d,n,\frac{d}{n}+\overline{n},q,z}^{\mathtt{dGLWR}_{spt}}(\mathcal{A} \circ \mathcal{E}') \end{aligned} \tag{13}$$

$Adv_{d,n,m,q_1,q_2}^{\mathtt{dGLWR}_{spt}}(\mathcal{Z})$ is the advantage of adversary $\mathcal{Z}$ in distinguishing m GLWR samples (with sparse-ternary secrets) from uniform, with the GLWR problem defined for the parameters d, n, q_1, q_2. Finally, the adversary $\mathcal{D}$ distinguishes between $U(\{f_R(\rho) \mid \rho \in \{0,1\}^{\mu B}\})$ and $(\chi_S)^{\overline{m}}$. The runtimes of $\mathcal{D}, \mathcal{A} \circ \mathcal{C}', \mathcal{A} \circ \mathcal{E}'$ are essentially the same as that of $\mathcal{A}$.

Proof. The proof of Theorem 1 proceeds via a sequence of seven games:

Game 0 This is the real IND-CPA game for CPA-PKE: $\mathrm{Adv}_{\text{CPA-PKE}}^{\text{IND-CPA}}(\mathcal{A}) = |\Pr(S_0) - 1/2|$.

Game 1 $(\boldsymbol{A}, \boldsymbol{B})$ is sampled from the uniform distribution on $\mathcal{R}_{n,q}^{d/n \times d/n} \times \mathcal{R}_{n,p}^{d/n \times \overline{n}}$ instead of from $O_{d/n,\chi_S,\overline{n},\boldsymbol{s}}$. Distinguishing between this and Game 1 leads, by a standard hybrid argument, to a distinguisher $\mathcal{C}'$ between $O_{d/n,\chi_S,\boldsymbol{s}}$ and the uniform distribution: $|\Pr S_0 - \Pr(S_1)| \leq \overline{n} \cdot \mathrm{Adv}_{d,n,\frac{d}{n},q,p}^{\mathtt{dGLWR}_{\mathrm{spt}}}(\mathcal{A} \circ \mathcal{C}')$.

Game 2 $\boldsymbol{R}$ is sampled uniformly from $\chi_S^{\overline{m}} = (\mathcal{U}(\mathcal{H}_{n,d/n}(h))^{1 \times \overline{m}}$ instead of via f_R. Distinguishing this game from game 2 leads to a distinguisher $\mathcal{D}$ for the above distributions: $\mathrm{Adv}^{f_R}(\mathcal{D}) \geq |\Pr(S_1) - \Pr(S_2)|$.

Game 3 $\boldsymbol{B}$ is replaced by $\boldsymbol{B}_q$ that is sampled uniformly from $\mathcal{R}_{n,q}^{d/n \times \overline{n}}$, and the ciphertext component $\boldsymbol{v}$ is replaced by

$$\boldsymbol{v}' = \left\langle \text{Sample}_\mu\left(\left\lfloor \frac{tq}{p} \cdot \frac{1}{q} \cdot \left\langle \boldsymbol{B}_q^T \boldsymbol{R} + \boldsymbol{H}_3 \right\rangle_q \right\rfloor \right) + \frac{t}{2^B} \cdot m_b \right\rangle_{\frac{tq}{p}}$$

Note that $\langle \boldsymbol{v}' \rangle_t = \boldsymbol{v}$. As $p|q$, the pairs $(\langle \boldsymbol{B}_q \rangle_p, \langle v' \rangle_t)$ and $(\boldsymbol{B}, \boldsymbol{v})$ in games 4 and 3 respectively, are equally distributed. So by providing $\mathcal{A}$ with input $(\langle \boldsymbol{B}_q \rangle_p, \langle v' \rangle_t)$, we obtain that: $\Pr(S_2) = \Pr(S_3)$. This technique is originally due to the authors of [29].

Game 4 For $z = max(p, tq/p)$, we define

$$\begin{aligned} \boldsymbol{U}' &= \left\langle \left\lfloor \frac{z}{q} \cdot \left\langle \boldsymbol{A}^T \boldsymbol{R} + \boldsymbol{H}_2 \right\rangle_q \right\rfloor \right\rangle_z, \\ \boldsymbol{v}'' &= \left\langle \text{Sample}_\mu(\left\lfloor \frac{z}{q} \cdot \left\langle \boldsymbol{B}_q^T \boldsymbol{R} + \boldsymbol{H}_3 \right\rangle_q \right\rfloor) + \frac{pz}{q2^B} \cdot m_b \right\rangle_z . \end{aligned}$$

We consider the following lemma:

Lemma 1 *For $a, b, c, Y \in \mathbb{Z}$, such that $a|b|c$,*

$$\left\lfloor \frac{a}{c} \cdot Y \right\rfloor = \left\lfloor \frac{a}{b} \cdot \left\langle \left\lfloor \frac{b}{c} \cdot Y \right\rfloor \right\rangle_b \right\rfloor \ (mod\ a).$$

Using the above lemma, we infer that $\boldsymbol{U} = \left\langle \frac{p}{z} \cdot \boldsymbol{U}' \right\rangle_p$, and $\boldsymbol{v}' = \left\langle \left\lfloor \frac{tq}{pz} \cdot \boldsymbol{v}'' \right\rfloor \right\rangle_{tq/p}$. We now introduce the matrix $\boldsymbol{V''} = \left\lfloor \frac{z}{q} \langle \boldsymbol{B}_q^T \boldsymbol{R} + \boldsymbol{H}_3 \rangle_q \right\rfloor + \frac{pz}{2^b} \boldsymbol{M}_b$, with all components of $\boldsymbol{M}_b$ in $\mathbb{Z}_{2^B}$ such that $v'' = \text{Sample}_\mu(\boldsymbol{V''})$. In Game 5, the cipher text $(\boldsymbol{U}, \boldsymbol{v'})$ is replaced by $(\boldsymbol{U'}, \boldsymbol{V''})$. As shown above, $(\boldsymbol{U'}, \boldsymbol{V''})$ can be transformed into $(\boldsymbol{U}, \boldsymbol{v'})$. Hence, if $\mathcal{A}$ is provided with these transformed inputs, then $\Pr(S_3) = \Pr(S_4)$. As all polynomial coefficients in $\boldsymbol{H}_2$ and $\boldsymbol{H}_3$ are equal to $\frac{q}{2z}$, we have that $\left[\begin{smallmatrix} \boldsymbol{U}' \\ \boldsymbol{V''} \end{smallmatrix} \right] = \left\lfloor \frac{z}{q} \cdot \left\langle \left[\begin{smallmatrix} \boldsymbol{A}^T \\ \boldsymbol{B_q}^T \end{smallmatrix} \right] \boldsymbol{R} \right\rangle_q \right\rfloor + \left[\begin{smallmatrix} 0 \\ \frac{pz}{q2^B} \boldsymbol{M}_b \end{smallmatrix} \right]$. As $\boldsymbol{A}, \boldsymbol{B}_q$ and $\boldsymbol{R}$ are uniformly distributed, the above implies that $\left[\begin{smallmatrix} \boldsymbol{U}' \\ \boldsymbol{V''} \end{smallmatrix} \right] - \left[\begin{smallmatrix} 0 \\ \frac{pz}{q2^B} \boldsymbol{M}_b \end{smallmatrix} \right]$ form $d/n + \overline{n}$ LWR samples.

Game 5 The components $\boldsymbol{U}'$ and $\boldsymbol{V}''$ are replaced by uniformly distributed matrices. Equivalently, $\boldsymbol{U}'$ and $\boldsymbol{V''} - \frac{pz}{q2^B} \boldsymbol{M}_b$ are replaced by uniformly distributed matrices. As this equivalence holds for any $\boldsymbol{M}_b$, it is irrelevant that $\boldsymbol{M}_b$ is chosen from an error-correcting code. Distinguishing between this and game 5 leads to a distinguisher $\mathcal{E}'$ between the uniform and GLWR distribution (with parameters as follows): $|\Pr(S_5) - \Pr(S_6)| \leq \overline{m} \cdot \mathrm{A}^{\mathtt{dGLWR}_{\mathrm{spt}}}_{d,n,\frac{d}{n}+\overline{n},q,z}(\mathcal{A} \circ \mathcal{E}')$. Furthermore, for each independently chosen message m_b, the distribution of the inputs to $\mathcal{A}$ is indistinguishable from uniform. Therefore $\Pr(S_6) = 1/2$.

Combining the equations above completes the proof of IND-CPA security for CPA-PKE.

We now prove the IND-CPA security claim of Round5.KEM.

Theorem 2 *If f_R has output indistinguishable from $(\chi_S)^{\overline{m}}$, and H is a secure pseudorandom function, then Round5.KEM is IND-CPA secure under the hardness assumption of the decision-GLWR problem with sparse-ternary secrets, assuming $t|p|z|q$ for $z = \max(p, tq/p)$. More precisely, for every IND-CPA adversary $\mathcal{A}$, if $Adv^{\text{IND-CPA}}_{\text{Round5.KEM}}(\mathcal{A})$ is the advantage in winning the IND-CPA game, then there exist adversaries $\mathcal{D}$, $\mathcal{G}$ and reduction algorithms $\mathcal{C}', \mathcal{E}'$ such that*

$$\begin{aligned} Adv^{\text{IND-CPA}}_{\text{Round5.KEM}}(\mathcal{A}) \leq \overline{n} \cdot Adv^{\texttt{dGLWR}_{spt}}_{d,n,\frac{d}{n},q,p}\left(\mathcal{A} \circ \mathcal{C}'\right) + Adv^{f_R}(\mathcal{D}) \\ + \overline{m} \cdot Adv^{\texttt{dGLWR}_{spt}}_{d,n,\frac{d}{n}+\overline{n},q,z}\left(\mathcal{A} \circ \mathcal{E}'\right) + Adv^{H}(\mathcal{G}) \end{aligned} \tag{14}$$

$Adv^{\texttt{dGLWR}_{spt}}_{d,n,m,q_1,q_2}(\mathcal{Z})$ is the advantage of adversary $\mathcal{Z}$ in distinguishing m GLWR samples (with sparse-ternary secrets) from uniform, with the GLWR problem defined for the parameters d, n, q_1, q_2. The adversary $\mathcal{D}$ distinguishes between $U(\{f_R(\rho) \mid \rho \in \{0,1\}^{\mu B}\})$ and $(\chi_S)^{\overline{m}}$. Finally, the adversary $\mathcal{G}$ distinguishes the output of the pseudorandom function H (given uniform input) from random. The runtimes of $\mathcal{D}, \mathcal{G}, \mathcal{A} \circ \mathcal{C}', \mathcal{A} \circ \mathcal{E}'$ are essentially the same as that of $\mathcal{A}$.

Proof. The IND-CPA security of Round5.KEM can be proved through a sequence of 8 games. The first 7 of them are similar as for CPA-PKE. In the final game, the shared key K is generated uniformly. An adversary that can distinguish between this game and the previous one leads to a distinguisher $\mathcal{G}$ between the output of the pseudorandom function H and the uniform distribution.

Next, as Round5.PKE is constructed from the key encapsulation mechanism CCA-KEM and a secure data-encapsulation mechanism in the canonical way as proposed by Cramer and Shoup [27], it is sufficient to show the IND-CCA security of CCA-KEM. When the hash functions G and H in Algorithms 8 and 9 are modeled as random oracles, CCA-KEM is IND-CCA secure, assuming the hardness of the decision GLWR problem with sparse-ternary secrets.

Theorem 3 *For any adversary $\mathcal{A}$ that makes at most q_H queries to the random oracle H, at most q_G queries to the random oracle G, and at most q_D queries to the decryption oracle, there exists an adversary $\mathcal{B}$ such that*

$$Adv^{\text{IND-CCA}}_{\text{CCA-KEM}}(\mathcal{A}) \leq 3 \cdot Adv^{\text{IND-CPA}}_{\text{CPA-PKE}}(\mathcal{B}) + q_G \cdot \delta + \frac{2q_G + q_H + 1}{2^{\mu B}} \tag{15}$$

when CPA-PKE and CCA-KEM both have a probability of decryption/decapsulation failure that is at most δ.

Proof. The proof of Theorem 3 proceeds via two transformation reductions due to [40]. First, Lemma 2 below establishes that the OW-PCA[5] security of the deterministic public key encryption scheme PKE_1 obtained from the public key encryption scheme PKE via transformation T [40], tightly reduces to IND-CPA

[5] The security notion of One-Way against Plaintext Checking Attacks.

security of PKE_1. This lemma is a special case of [40, Theorem 3.2] with $q_v = 0$, since by definition OW-PCA security is OW-PCVA[6] security where the attacker is not allowed to query the ciphertext validity checking oracle.

Lemma 2 (Adapted from [40, Theorem 3.2]) *Assume PKE to be δ correct. Then, for any OW-PCA adversary $\mathcal{B}$ that issues at most q_G queries to the random oracle G, q_P queries to a plaintext checking oracle P_{CO}, there exists an IND-CPA adversary $\mathcal{C}$ such that*

$$Adv^{\text{OW-PCA}}_{PKE_1}(\mathcal{B}) \leq q_G \cdot \delta + \frac{2q_G + 1}{|\mathcal{M}|} + 3 \cdot Adv^{\text{IND-CPA}}_{PKE}(\mathcal{C}) \tag{16}$$

where $\mathcal{M}$ is the message/plaintext space of the public key encryption schemes PKE and PKE_1.

Next, combination of Lemma 2 and the reduction in [40, Theorem 3.4] shows that the IND-CCA security of a KEM with implicit rejection that is constructed using a non-deterministic PKE (like CCA-KEM), tightly reduces to the IND-CPA security of said PKE.

Direct application of [40, Theorem 4.6], similarly as in [19, Theorem 4.2], shows that CCA-KEM is IND-CCA secure in the quantum random oracle model. The resulting security bound however is not tight.

Theorem 4 *For any quantum adversary $\mathcal{A}$ that makes at most q_H queries to the quantum random oracle H, at most q_G queries to the quantum random oracle G, and at most q_D (classical) queries to the decapsulation oracle, there exists a quantum adversary $\mathcal{B}$ such that*

$$Adv^{\text{IND-CCA}}_{\text{CCA-KEM}}(\mathcal{A}) \leq 4q_H \sqrt{q_D \cdot q_H \cdot \delta + q_G \cdot \sqrt{Adv^{\text{IND-CPA}}_{\text{CPA-PKE}}(\mathcal{B})}} \tag{17}$$

5.3 Hardness of Sparse-Ternary LWR

In this section, we prove that the Decision-LWR problem with sparse-ternary secrets is hard assuming that the small modulus p divides the large modulus q, and that decision-LWE with Gaussian noise and secrets chosen uniformly from $\mathbb{Z}_q^d$ is hard.

Theorem 5 *Let $k, p, q \geq 1$ and $m \geq n \geq h \geq 1$ be integers such that p divides q, and $k \geq m' = \frac{q}{p} \cdot m$. Let $\epsilon \in (0, \frac{1}{2})$, and $\alpha, \delta > 0$ such that*

$$\alpha \geq q^{-1} \sqrt{\left(\frac{2}{\pi}\right) \ln(2n(1+\epsilon^{-1}))}, \ \binom{n}{h} 2^h \geq q^{k+1} \cdot \delta^{-2}, \ m = O\left(\frac{\log n}{\alpha \sqrt{10h}}\right). \tag{18}$$

There exist three (transformation) reductions from $\mathtt{dLWE}_{k,m',q,D_\alpha}$ to $\mathtt{dLWE}_{n,m',q,D_{\alpha\sqrt{10h}}}(\mathcal{U}(\mathcal{H}_n(h)))$ such that for any algorithm for the latter problem

[6] The security notion of OW-PCA, with access to a ciphertext Validity checking oracle.

with advantage ζ, at least one of the reductions produces an algorithm for the former with advantage at least $(\zeta-\delta)/(3m')-41\epsilon/2-\sum_{s|q,s\ prime} s^{-k-1}$. Moreover, there is a reduction from $\mathtt{dLWE}_{n,m',q,D_{\alpha\sqrt{10h}}}(\mathcal{U}(\mathcal{H}_n(h)))$ to $\mathtt{dLWR}_{n,m,q,p}(\mathcal{U}(\mathcal{H}_n(h)))$.

Proof. Combination of Lemma 3 and Lemma 6 with $\alpha' = \alpha\sqrt{10h}$.

Step 1: Reduction from LWE with secrets in $\mathbb{Z}_q$ and Gaussian errors to Sparse-ternary LWE. In [25, Theorem 1], specializing [24, Theorem 4], it is shown that if $\binom{n}{h}2^h > q^{k+1}$ and $\omega > \alpha\sqrt{10h}$, then the $\mathtt{dLWE}_{n,m,q,D_\omega}(\mathcal{U}(\mathcal{H}_n(h)))$ problem is at least as hard as the $\mathtt{dLWE}_{k,m,q,D_\alpha}$ problem. More formally, generalizing [22, Theorem 4.1], the following holds.

Lemma 3 *Let $k, q \geq 1$ and $m \geq n \geq h \geq 1$ be integers, and let $\epsilon \in (0, \frac{1}{2})$, and $\alpha, \delta > 0$ such that $\alpha \geq q^{-1}\sqrt{(2/\pi)\ln(2n(1+\epsilon^{-1}))}$, and $\binom{n}{h}2^h \geq q^{k+1}\cdot\delta^{-2}$. There exist three (transformation) reductions from $\mathtt{dLWE}_{k,m,q,D_\alpha}$ to $\mathtt{dLWE}_{n,m,q,D_{\alpha\sqrt{10h}}}(\mathcal{U}(\mathcal{H}_n(h)))$ such that for any algorithm for the latter problem with advantage ζ, at least one of the reductions produces an algorithm for the former with advantage at least $(\zeta-\delta)/(3m) - 41\epsilon/2 - \sum_{s|q,s\ prime} s^{-k-1}$.*

Step 2: Reduction from Sparse-ternary LWE to Sparse-ternary LWR. Bai et al. provide in [9, Theorem 6.4] a reduction from LWE with Gaussian noise to LWR, that is based on two independent reductions. One of these reductions [9, Theorem 6.3] holds for any secret distribution with support on $\mathbb{Z}_q^{n*} = \{(x_1, \ldots, x_n) \in \mathbb{Z}_q^n \mid \gcd(x_1, x_2, \ldots, x_n, q) = 1\}$, and therefore can be applied when the secret is chosen from $\{-1, 0, 1\}^n$. The other reduction [9, Theorem 5.1] however, implicitly assumes the secret to be chosen uniformly at random from $\mathbb{Z}_q^n$. Below, we describe an extension of [9, Theorem 5.1] that describes a reduction from LWE with Gaussian noise and sparse ternary secrets to LWR with sparse-ternary secrets. U_B denotes the continuous uniform distribution in $[-B, \ldots, B]$.

Lemma 4 (Adapted from [9, Theorem 5.1]) *Let n, m, q be positive integers. Let $\alpha, B > 0$ be real numbers with $B = \Omega(m\alpha/\log n)$ and $Bq \in \mathbb{Z}$. Let $m > \log\left(\binom{n}{h}2^h\right)/\log(\alpha+B)^{-1} \geq 1$. Then there is a polynomial time reduction from $\mathtt{LWE}_{n,m,q,D_\alpha}(\mathcal{U}(\mathcal{H}_n(h)))$ to $\mathtt{LWE}_{n,m,q,\phi}(\mathcal{U}(\mathcal{H}_n(h)))$ with $\phi = \frac{1}{q}\lfloor qU_B \rceil$.*

Proof. The reduction proceeds similar to that of [9, Theorem 5.1], relying on five steps.

1. A reduction from $\mathtt{dLWE}_{n,m,q,D_\alpha}$ to $\mathtt{dLWE}_{n,m,q,\psi}$, with $\psi = D_\alpha + U_B$.
2. A reduction from $\mathtt{dLWE}_{n,m,q,\psi}$ to $\mathtt{sLWE}_{n,m,q,\psi}$. We adapt the corresponding step in [9, Theorem 5.1] to work for the uniform distribution on $\mathcal{H}_n(h)$ instead of that on $\mathbb{Z}_q^n$, resulting in the bound on m as in our lemma.
3. A reduction from $\mathtt{sLWE}_{n,m,q,\psi}$ to $\mathtt{sLWE}_{n,m,q,U_B}$.
4. A reduction from $\mathtt{sLWE}_{n,m,q,U_B}$ to $\mathtt{sLWE}_{n,m,q,\phi}$, with $\phi = \frac{1}{q}\lfloor qU_B \rceil$.

5. A reduction from $\mathtt{sLWE}_{n,m,q,\phi}$ to $\mathtt{dLWE}_{n,m,q,\phi}$. Since the modulus q is not a prime, the argument from [9, Theorem 5.1] cannot be applied. Instead, we extend an argument due to Regev (see, e.g, [56]) to prove the search-to-decision reduction, which requires that Bq is an integer. We first state an easy lemma.

   Lemma 5 *Let $a > 1$, and let ϕ be the discrete probability distribution obtained by rounding the continuous uniform probability on $[-a, a]$ to the closest integer. If a is an integer, then $\sum_{k\,even}\phi(k) = \sum_{k\,odd}\phi(k) = \frac{1}{2}$.*

   Proof. For $|k| \leq \lfloor a \rfloor - 1$, the interval $[k - \frac{1}{2}, k + \frac{1}{2}]$ is a subset of $-[a, a]$, so that $\sum_{k\equiv 1-\lfloor a\rfloor \pmod 2}\phi(k) = \sum_{j=0}^{\lfloor a\rfloor -1}\phi(2j - \lfloor a\rfloor + 1) = \frac{\lfloor a\rfloor}{2a}$.

   We are now in a position to extend Regev's reduction. Let ϕ be a probability distribution on $\mathbb{Z}_q$ such that $\sum_k \phi(2k) = \sum_k \phi(2k+1) = \frac{1}{2}$. For each $\mathbf{s} \in \mathbb{Z}_q^n$, the probability distribution $A_{\mathbf{s},\phi}$ on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is obtained by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly, e according to ϕ, and outputting $(\mathbf{a}, (\mathbf{a}, \mathbf{s}) + e)$ (additions modulo q). If qB is integer, then a distinguisher for $\mathtt{dLWE}_{n,m,q,\phi}(D_s)$ will lead to a solver for $\mathtt{sLWE}_{n,m,q,\phi}(D_s)$ for any secret distribution D_s supported on $\{-1, 0, 1\}^n$, where ϕ is the discrete noise $\frac{1}{q}\lfloor qU_B \rfloor$. If Bq is integer, ϕ is distributed as $\phi(k) = \frac{1}{2B}$ for $|k| \leq B - 1$, and $\phi(B) = \phi(-B) = \frac{1}{4B}$.
   If Bq is integer, then a distinguisher for deciding between uniform samples $(\mathbf{a}, u) \in U(\mathbb{Z}_q^n) \times U(Z_q)$ and samples $(\mathbf{a}, b)$ from $A_{\mathbf{s},\phi}$ for some unknown $s \in \mathcal{S} \subset \{-1, 0, 1\}^n$ can be used for solving: first, we show how to find s_1, the secret's first coordinate. For each $k \in \mathbb{Z}_q$, consider the transformation: for each pair $(\mathbf{a}, b)$, we choose a random $r \in \mathbb{Z}_q$ and output $(\mathbf{a}', b') = (\mathbf{a} + (r, 0, \ldots, 0), b + rk)$. This transformation takes the uniform distribution to itself. Now assume that $b = (\mathbf{a}, \mathbf{s}) + e$ for some $s \in \mathcal{S}$ and some error e. Then $b' = (\mathbf{a}', \mathbf{s}) + r(k - s_1) + e$. If $k = s_1$, then $(\mathbf{a}', b')$ is from $A_{\mathbf{s},\phi}$. If $|k - s_1| = 1$, then $r(k - s_1)$ is uniform over $\mathbb{Z}_q$, and so $(\mathbf{a}', \mathbf{b})$ follows the uniform distribution. Finally, it can be that $|k - s_1| = 2$. We consider $k - s_1 = 2$, the other case being similar. Then, $b' = (\mathbf{a}, \mathbf{s}) + 2r + e \pmod q$. If q is odd, $2r$ is uniformly distributed on $\mathbb{Z}_q$, so $(\mathbf{a}', \mathbf{b})$ is uniformly distributed. If q is even, $2r$ is distributed uniformly on the even elements of $\mathbb{Z}_q$. With our specific error distribution, e is even with probability $\frac{1}{2}$, so that $2r + e$ is distributed uniformly on $\mathbb{Z}_q$. So in this case too, $(\mathbf{a}', b)$ is distributed uniformly.

Finally, we state the reduction from $\mathtt{dLWE}_{n,m,q,D_\alpha}$ to $\mathtt{dLWR}_{n,m,q,p}$, for the sparse-ternary secret distribution.

Lemma 6 *Let p, q be positive integers such that p divides q. Let $\alpha' > 0$. Let $m' = m \cdot (q/p)$ with $m = O(\log n/\alpha')$ for $m' \geq m \geq n \geq 1$. There is a polynomial time reduction from $\mathtt{dLWE}_{n,m',q,D_{\alpha'}}$ to $\mathtt{dLWR}_{n,m,q,p}$, both defined for the sparse-ternary secret distribution.*

Proof. Let $B = q/2p$. The reduction has two steps: *first*, a reduction from $\mathtt{dLWE}_{n,m',q,D_{\alpha'}}$ to $\mathtt{dLWE}_{n,m',q,\phi}$, where $B = \Omega(m'\alpha'/\log n)$, due to Lemma 4.

Second, a reduction from $\texttt{dLWE}_{n,m',q,\phi}$ to $\texttt{dLWR}_{n,m,q,p}$, due to [9, Theorem 6.3]. As $m' = m \cdot (q/p) = (q/p)O(\frac{\log n}{\alpha'})$, it follows that $B = q/2p = \Omega(m'\alpha'/\log n)$, so that Lemma 4 indeed is applicable.

Note that the conditions imposed by Lemma 4 imply that $1/\alpha$ must at least grow linearly in n. This is a common bottleneck in known LWE to LWR reductions [9,16,10], and is an open problem. As such, it stands as an obstacle in using the above reduction in selecting concrete parameters for our scheme. The reduction does still strongly demonstrate the asymptotic underlying security of our scheme. We note finally, that no lattice-based cryptosystem to the best of our knowledge that demonstrates practical performance, actually selects concrete parameters from reductions to underlying worst-case problems.

6 Concrete Security of Round5

In this section we analyze the security of Round5 against known attacks. In our analysis, we adopt the conservative approach introduced in [3, Sec. 6.1] of considering the *core-SVP* hardness of (Ring) Learning with Rounding, i.e., we assume that the number of calls by the lattice reduction algorithm to the SVP oracle is *one*. This is a conservative lower bound on the attack cost, as the number of calls in practice is more than one (increasing the cost) but difficult to accurately estimate.

Furthermore, we consider sieving algorithms instead of enumeration as this SVP oracle since they lead to stronger attacks for lattice dimensions in the range we consider [3], further enhanced with Grover's quantum search algorithm [37] to fit a post-quantum scenario. This leads to an attack cost estimate of $2^{0.265b+o(b)}$ for block-size b in BKZ lattice reduction [23,62], for example. Ignoring Grover speedup leads to a *classical* atack cost estimate of $2^{0.292b+o(b)}$. Both the above heuristic costs stem from the work in [44,45], and are assumed to be the best known running time of a *sieve* algorithm, *quantum* and *classical* respectively. To remain consistently conservative in our analysis, we ignore the sub-exponential factor $o(b)$ in the attack cost, which is known to be greater than 1 in practice [3].

We optimize Round5 parameters such that the best known attacks result in at least a minimum targeted cost – both for post-quantum and classical attack scenarios, following which we choose parameters that result in minimum bandwidth requirements.

6.1 Lattice Reduction-based attacks

As $\boldsymbol{B} = \left\langle \left\lfloor \frac{p}{q} \cdot \langle \boldsymbol{AS} \rangle_q \right\rceil \right\rangle_p$, the definition of the rounding function $\lfloor \cdot \rceil$, implies that $\boldsymbol{B} \equiv \frac{p}{q}\langle \boldsymbol{AS} \rangle_q + \boldsymbol{E} \pmod p$ with $\boldsymbol{E} \in (-1/2, 1/2]$. By multiplying with $\frac{q}{p}$, we infer that

$$\frac{q}{p}\boldsymbol{B} \equiv \boldsymbol{AS} + \boldsymbol{E'} \pmod q \text{ with } \boldsymbol{E'} \in (-\frac{q}{2p}, \frac{q}{2p}] \cap \mathbb{Z} \tag{19}$$

6.2 Weighted Primal Attack

We consider Eq. (19) for one column and the m top rows of $\frac{q}{p}\boldsymbol{B}$. We denote the corresponding column by $\boldsymbol{b}$, write $\boldsymbol{s}$ for the corresponding column of $\boldsymbol{S}$, and $\boldsymbol{A}_m$ for the m top rows of $\boldsymbol{A}$. We then have, for some $\boldsymbol{e} \in (-\frac{q}{2p}, \frac{q}{2p}]^m$, $\boldsymbol{b} \equiv \boldsymbol{A}_m\boldsymbol{s} + \boldsymbol{e} \pmod q$. In order to benefit from the fact that $\|\boldsymbol{s}\| \ll \|\boldsymbol{e}\|$, the attacker considers the scaled lattice [8,1,25]: $\Lambda_\omega = \{(\omega\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \in (\omega\mathbb{Z})^d \times \mathbb{Z}^m \times \mathbb{Z} : \left(\boldsymbol{A}_m|\boldsymbol{I}_m|-\boldsymbol{b}\right)\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{z}\end{pmatrix} = \boldsymbol{0} \pmod q\}$, which contains the vector $\boldsymbol{v}_\omega = (\omega\boldsymbol{s}^T, \boldsymbol{e}^T, 1)^T$. The attacker then searches for the shortest vector in Λ_ω, that he hopes to be equal to $\boldsymbol{v}_\omega$. A lattice reduction algorithm can be used to obtain a reduced basis of Λ_ω, the first vector of which will be the shortest due to a common heuristic. Assuming that BKZ [23,62] with block-size b is used as the lattice reduction algorithm, $\boldsymbol{v}_\omega$ will be detected if its projection $\widetilde{\boldsymbol{v}}_b$ onto the vector space of the last b Gram-Schmidt vectors of Λ_ω is shorter than the expected norm of the $(d'-b)^{th}$ Gram-Schmidt vector $\widetilde{\boldsymbol{b}}_{d'-b}$, where d' is the dimension of Λ_ω [3, Sec. 6.3],[18]; in other words, if (using the Geometric Series Assumption), $\|\widetilde{\boldsymbol{v}}_b\| < \|\widetilde{\boldsymbol{b}}_{d'-b}\| = \delta^{2b-d'-1} \cdot (\mathrm{Vol}(\Lambda_\omega))^{\frac{1}{d'}}$, where $\delta = ((\pi b)^{\frac{1}{b}} \cdot \frac{b}{2\pi\mathrm{e}})^{\frac{1}{2(b-1)}}$ and $\mathrm{Vol}(\Lambda_\omega) = \omega^d q^m$. Consequently, Round5 is secure against the primal attack in Λ_ω with BKZ with block size b if:

$$\sqrt{(\omega^2 \cdot h + {\sigma'}^2 m) \cdot b/(d+m)} \geq \delta^{2b-d'-1} \cdot (q^m\omega^d)^{\frac{1}{d'}}, \text{ where} \tag{20}$$

$$\delta = ((\pi b)^{\frac{1}{b}} \cdot \frac{b}{2\pi\mathrm{e}})^{\frac{1}{2(b-1)}}, \sigma' = (q/2\sqrt{3}p), \text{ and } d' = d+m+1.$$

It can be shown from Eq. (20) that the optimal choice of ω for the attacker satisfies $\omega^2 = \frac{dm\sigma'^2}{h(d'-d)} = \frac{dm\sigma'^2}{h(m+1)} \approx \frac{d}{h}\sigma'^2$.

6.3 Weighted Dual Attack

The dual attack against LWE/LWR [3],[1] employs a short vector $(\boldsymbol{v}, \boldsymbol{w}) \in \mathbb{Z}^m \times \mathbb{Z}^d$ in the dual lattice $\Lambda^* = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{Z}^m \times \mathbb{Z}^d : \boldsymbol{A}_m^T\boldsymbol{x} = \boldsymbol{y} \pmod q\}$. It constructs a distinguisher for LWR using $z = \{\boldsymbol{v}^T\boldsymbol{b}\}_q = \{\boldsymbol{v}^T(\boldsymbol{A}_m\boldsymbol{s} + \boldsymbol{e})\}_q = \{\boldsymbol{w}^T\boldsymbol{s} + \boldsymbol{v}^T\boldsymbol{e}\}_q$. Since $\|\boldsymbol{s}\| \ll \|\boldsymbol{e}\|$ in our case, the attacker can enforce that $\|\boldsymbol{w}\| \gg \|\boldsymbol{v}\|$ to ensure that $\|\boldsymbol{w}^T\boldsymbol{s}\| \approx \|\boldsymbol{v}^T\boldsymbol{e}\|$ similar to [1]. He does so by choosing $\omega = \sigma'\sqrt{m/h}$ (for the LWR rounding error with variance $\sigma'^2 = q^2/12p^2$), and considering the lattice: $\Lambda_\omega^* = \{(\boldsymbol{x}, \boldsymbol{y}/\omega) \in \mathbb{Z}^m \times \left(\frac{1}{\omega} \cdot \mathbb{Z}^d\right) : \boldsymbol{A}_m^T\boldsymbol{x} = \boldsymbol{y} \pmod q\}$.

A short vector $(\boldsymbol{v}, \boldsymbol{w}) \in \Lambda_\omega^*$ gives a short vector $(\boldsymbol{v}, \omega\boldsymbol{w}) \in \Lambda^*$ that is used to construct the distinguisher z. If $\boldsymbol{b}$ is uniform modulo q, so is z. If $\boldsymbol{b}$ is an LWR sample, then $z = \{(\omega\boldsymbol{w})^T\boldsymbol{s} + \boldsymbol{v}^T\boldsymbol{e}\}_q = \{\boldsymbol{w}^T(\omega\boldsymbol{s}) + \boldsymbol{v}^T\boldsymbol{e}\}$ has a distribution approaching a Gaussian of zero mean and variance $\|(\boldsymbol{v}, \boldsymbol{w})\|^2 \cdot {\sigma'}^2$ as the lengths of the vectors increase, due to the Central limit theorem. Note that ω has been chosen such that $\|\omega\boldsymbol{s}\| \approx \|\boldsymbol{e}\|$. The maximal statistical distance between this and the uniform distribution modulo q is bounded by $\epsilon \approx 2^{-1/2}\exp(-2\pi^2(\|(\boldsymbol{v}, \boldsymbol{w})\|^2 \cdot \sigma'/q)^2)$ [15, Appendix B]. Lattice reduction with root-Hermite factor δ yields a short(est)

vector of length $\delta^{d'-1} \cdot \mathrm{Vol}(\Lambda_\omega^*)^{1/d'}$, where $d' = m + d$ and $\mathrm{Vol}(\Lambda_\omega^*) = (q/\omega)^d$ are Λ_ω^*'s dimension and volume, respectively. However, finding only one such short vector is not enough, as the resulting ϵ is too small to distinguish a final key which is hashed. The attack must therefore be repeated at least $\max(1, 1/2^{0.2075b} \cdot \epsilon^2)$ times [3], when considering BKZ with block size b. The cost of the weighted dual attack on LWR (with dimension d, large modulus q, rounding modulus p) using m samples thus is

$$\begin{gathered}(b \cdot 2^{cb}) \cdot \max(1, 1/(\epsilon^2 \cdot 2^{0.2075 \cdot b})), \text{where} \\ \epsilon = 2^{-1/2} \cdot \mathrm{e}^{-2\pi^2\left(\left(\|(\boldsymbol{v},\boldsymbol{w})\|^2 \cdot \sigma'\right)/q\right)^2},\ \|(\boldsymbol{v},\boldsymbol{w})\|^2 = \delta^{m+d-1} \cdot (q/\omega)^{d/(m+d)}, \\ \delta = ((\pi b)^{\frac{1}{b}} \cdot \frac{b}{2\pi\mathrm{e}})^{\frac{1}{2(b-1)}},\ \omega = \sigma' \cdot \sqrt{m/h},\ \text{and } \sigma' = (q/2\sqrt{3}p).\end{gathered} \tag{21}$$

The first term $(b \cdot 2^{cb})$ in the overall attack cost is that of running BKZ with block-size b. To obtain conservative security estimates, we choose the BKZ sieving exponent $c = 0.265$, which is the best known complexity estimate of lattice sieving algorithms when enhanced with Grover's quantum search algorithm [44,45].

6.4 Hybrid Attack

In this section, we consider a hybrid lattice reduction and meet-in-the-middle attack originally due to [41] and analyzed further in [66], that benefits from the fact that secret-keys in Round5 are sparse and ternary. This attack considers the lattice $\Lambda' = \{\boldsymbol{x} \in \mathbb{Z}^{m+d+1} \mid (\boldsymbol{I}_m | \boldsymbol{A}_m | -\boldsymbol{b})\boldsymbol{x} \equiv 0 \bmod q\}$ for some $m \in [1, d]$, with basis $\boldsymbol{B}' = \left[\begin{smallmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{I}_r \end{smallmatrix}\right]$, where $0 < r < d$ is the meet-in-the-middle dimension chosen by the attacker. Λ' contains a short vector $\boldsymbol{v} = (\boldsymbol{e}^T, \boldsymbol{s}^T, \boldsymbol{1})^T$. Rewriting $\boldsymbol{v} = (\boldsymbol{v}_l^T \boldsymbol{v}_g^T)^T$, the attacker first tries to recover $\boldsymbol{v}_g$ of length $r < d$ by guessing, which if successful, allows him to recover $\boldsymbol{v}_l$ given a sufficiently reduced $\widetilde{\boldsymbol{B}}$ by considering $\boldsymbol{C}\boldsymbol{v}_g$ as the target of Babai's Nearest Planes algorithm [7], for instance. This is due to the existence of a $\boldsymbol{x} \in \mathbb{Z}^{d+m+1-r}$ such that the lattice vector $-\boldsymbol{B}\boldsymbol{x} \in \Lambda(\boldsymbol{B})$ is *close* to the one $\boldsymbol{C}\boldsymbol{v}_g$ (due to $\boldsymbol{v}_l$ being short), as $\left(\begin{smallmatrix} \boldsymbol{v}_l \\ \boldsymbol{v}_g \end{smallmatrix}\right) = \boldsymbol{B}'\left(\begin{smallmatrix} \boldsymbol{x} \\ \boldsymbol{v}_g \end{smallmatrix}\right) = \left(\begin{smallmatrix} \boldsymbol{B}\boldsymbol{x}+\boldsymbol{C}\boldsymbol{v}_g \\ \boldsymbol{v}_g \end{smallmatrix}\right)$. The cost of recovering $\boldsymbol{v}_l$ (in terms of calls to the Nearest Plane algorithm) from guessed $\boldsymbol{v}_g$ candidates may be reduced to its square root using a Meet-in-the-Middle (MITM) approach [41]. Considering the sparse-ternary restriction on $\boldsymbol{v}_g$ allows further optimizations. We estimate this cost as $2^{\frac{1}{2}r \cdot H}$ [39], H being the per-coordinate entropy of the the guessed $\boldsymbol{v}_g$ (which depends on the secret-key distribution), and the square root resulting from either the use of MITM or Grover's search [37]. This leaves the cost of finding a sufficiently reduced basis $\widetilde{\boldsymbol{B}}$ of $\Lambda(\boldsymbol{B})$. Here, the attacker can again use a similar rescaling trick with $\omega^2 = \frac{d}{h}\sigma'^2$ as in Section 6.2 to exploit Round5's sparse-ternary secrets. We estimate this lattice reduction cost in terms of (the block-size of) BKZ similar to the previous sections, and we minimize the total cost of the Hybrid attack over all possible (r, m) pairs.

Our above analysis of the hybrid attack could be further extended following the work of Wunderer [66], which concerns a more accurate analysis of the

attack: he considers that the attacker may choose a faster lattice reduction (a trade off against the consequently increased cost of Babai's algorithm due to the lower reduction quality). He also considers that likelier candidates for $\boldsymbol{v}_g$ may be generated at an earlier phase, reducing the number of calls to Babai's algorithm. An improved analysis along these lines would enable more accurate estimation of the attack cost, leading to better Round5 parameters. We leave this as future work.

6.5 Attacks against Sparse Secrets

We consider an attack against *sparse* secrets due to Albrecht *et al.* [1]. In the primal attack, the attacker tries to find $\boldsymbol{s}$ from the equation $\boldsymbol{b} \equiv \boldsymbol{A}_m\boldsymbol{s} + \boldsymbol{e}$. By setting k random components of $\boldsymbol{s}$ to zero, the attacker effectively removes the k corresponding columns of $\boldsymbol{A}_m$, and solves an LWE/LWR problem in a lattice of dimension $d - k$, hence at a lower cost. As $\boldsymbol{s}$ has $d - h$ zeroes, the guess is correct with probability $\binom{d-h}{k}/\binom{d}{k}$, and therefore on average should be repeated $\binom{d}{k}/\binom{d-h}{k}$ times. The overall cost thus equals this number of repetitions times the cost for lattice reduction in such a lattice of dimension $d-k$. A similar analysis can be made for the dual attack. We optimize over the Hamming weight to choose the smallest value such that Albrecht *et al.*'s attack results in at least a minimum targeted cost (both for the standard attack embodiment mentioned above as well as an adaptive embodiment described in [1]). We finally note that for all our chosen parameters, the Hybrid attack (Section 6.4) outperforms this one.

7 Parameter Selection and Performance

The security of Round5 depends, among other parameters, on the dimension d and the moduli q and p. Round5 can instantiate different underlying problems depending on n: $n = 1$ for LWR and $n = d$ for RLWR. The moduli are chosen to be powers of 2, ensuring that operations remain efficient in both the LWR and RLWR instantiations. A restriction that we enforce in our parameter choices is that $\Phi_{n+1}(x)$ must be irreducible modulo two to avoid any possible vulnerabilities as in some cases of power-of-2 cyclotomic rings [13,14].

In this paper, parameter sets are designated as follows: For ring variants ($n = d$) we have the format `R5ND_`$\{l\}$`KEM` and `R5ND_`$\{l\}$`PKE`, where $l \in \{1, 3, 5\}$ denotes NIST security level, and ending `KEM` indicates IND-CPA secure KEM parameter set while `PKE` indicates IND-CCA secure public key encryption scheme. Function $\boldsymbol{f}^{(0)}_{d,n}$ is always used to generate the public value $\boldsymbol{A}$ in ring setting. In the non-ring setting $n = 1$ however, we have three options, $\boldsymbol{f}^{(0)}_{d,n}, \boldsymbol{f}^{(1)}_{d,n}$ and $\boldsymbol{f}^{(2)}_{d,n}$, so the designator takes the form `R5T`$\{\tau\}$`_`$\{l\}$`KEM` and `R5T`$\{\tau\}$`_`$\{l\}$`PKE`, where $\tau \in \{0, 1, 2\}$ is the index for the function $f^{(\tau)}_{d,n}$, $l \in \{1, 3, 5\}$ is the security level, and KEM/PKE has the same meaning as before.

Table 1: Round5 parameter sets, with performance estimates, post-quantum and classical security levels, and failure rate. Round5.PKE ciphertext sizes do not include the overhead required for DEM (typically 16 bytes for an authentication tag).

		Round5.KEM			Round5.PKE		
	Parameters	**CPA NIST1**	**CPA NIST3**	**CPA NIST5**	**CCA NIST1**	**CCA NIST3**	**CCA NIST5**
$n = d$, Ring variants.	d, n, h	522, 522, 208	756, 756, 242	1018, 1018, 254	546, 546, 158	786, 786, 204	1108, 1108, 198
	q, p, t	$2^{14}, 2^8, 2^4$	$2^{15}, 2^8, 2^4$	$2^{15}, 2^8, 2^4$	$2^{16}, 2^8, 2^4$	$2^{16}, 2^8, 2^6$	$2^{16}, 2^8, 2^5$
	$B, \bar{n}, \bar{m}, f$	1, 1, 1, 3	1, 1, 1, 3	1, 1, 1, 3	1, 1, 1, 3	1, 1, 1, 3	1, 1, 1, 3
	μ	$128 + 91$	$192 + 103$	$256 + 121$	$128 + 91$	$192 + 103$	$256 + 121$
	Bandwidth	1170 B	1684 B	2257 B	1234 B	1842 B	2516 B
	Public key	538 B	780 B	1050 B	562 B	810 B	1140 B
	Ciphertext	632 B	904 B	1207 B	672 B	1032 B	1376 B
	PQ Security	2^{117}	2^{176}	2^{242}	2^{120}	2^{181}	2^{246}
	Classical	2^{128}	2^{193}	2^{257}	2^{128}	2^{193}	2^{256}
	Failure rate	2^{-76}	2^{-75}	2^{-64}	2^{-129}	2^{-128}	2^{-129}
	Version ($f^{(0)}_{d,d}$)	`R5ND_1KEM`	`R5ND_3KEM`	`R5ND_5KEM`	`R5ND_1PKE`	`R5ND_3PKE`	`R5ND_5PKE`
$n = 1$, Non-ring variants.	d, n, h	635, 1, 266	929, 1, 268	1186, 1, 712	694, 1, 152	932, 1, 540	1198, 1, 574
	q, p, t	2^{15}, 2^{11}, 2^{10}	2^{14}, 2^{11}, 2^{10}	2^{14}, 2^{12}, 2^7	2^{13}, 2^{11}, 2^{10}	2^{14}, 2^{12}, 2^9	2^{14}, 2^{12}, 2^{10}
	$B, \bar{n}, \bar{m}, f$	4, 6, 6, 0	4, 6, 8, 0	4, 8, 8, 0	4, 5, 7, 0	4, 6, 8, 0	4, 8, 8, 0
	μ	32	48	64	32	48	64
	Bandwidth	10535 B	17969 B	28553 B	11553 B	19703 B	28925 B
	Public key	5256 B	7690 B	14265 B	4789 B	8413 B	14409 B
	Ciphertext	5279 B	10279 B	14288 B	6764 B	11290 B	14516 B
	PQ Security	2^{119}	2^{182}	2^{233}	2^{122}	2^{176}	2^{233}
	Classical	2^{128}	2^{192}	2^{256}	2^{128}	2^{192}	2^{256}
	Failure rate	2^{-65}	2^{-65}	2^{-84}	2^{-128}	2^{-135}	2^{-129}
	Version ($f^{(0)}_{d,d}$)	`R5T0_1KEM`	`R5T0_3KEM`	`R5T0_5KEM`	`R5T0_1PKE`	`R5T0_3PKE`	`R5T0_5PKE`
	Version ($f^{(1)}_{d,d}$)	`R5T1_1KEM`	`R5T1_3KEM`	`R5T1_5KEM`	`R5T1_1PKE`	`R5T1_3PKE`	`R5T1_5PKE`
	Version ($f^{(2)}_{d,d}$)	`R5T2_1KEM`	`R5T2_3KEM`	`R5T2_5KEM`	`R5T2_1PKE`	`R5T2_3PKE`	`R5T2_5PKE`

Table 1 summarizes the parameters for Round5.KEM and Round5.PKE targeting NIST security categories I, III, and V, along with (bandwidth) requirement and security levels considering the best known (classical and quantum) attacks against Round5. The parameter f in this table refers to the parameter of XEf (Section 3.3), that is the instantiation of the (generic) error-correction mechanism ECC_Enc_y used in the core Round5 building block CPA-PKE (see Section 3.2, Algorithm 2).

The security estimates of both the Round2 [6] and Hila5 [60] NIST PQC proposals are conservative and were independently verified by an independent analysis [2] of various lattice-based proposals to the NIST standardization process. Round5, which combines features from the two above cryptosystems, further improves on their security analyses, and achieves even better (post-quantum and classical) bit-security to performance ratios, while remaining highly conservative and also fully compliant with NIST PQC security categories.

7.1 Comparison and Discussion

Table 2 leads to two key observations about Round5:

- ***Round5 is very fast***: Especially the ring variants offer superior speed performance when compared to other lattice-based candidates. Findings in [61] also leading performance characteristics at least on (Cortex M). The non-ring variants have not received the same level of optimization; we expect that those can be made to run an order of magnitude faster.
- ***Round5 is very compact***: The ring variants achieve a huge bandwidth reduction compared with other ring designs such as NewHope [54]. Module lattice designs come closer, but still Round5's flexibility in choice of the parameter n together with rounding and the capability to use error correction code allows for messages that are between 10 % and 20 % smaller. Non-ring configurations of Round5 such as `R5T0_3PKE` provide around 70 bit more security compared with Frodo for the same bandwidth requirements.

The two above features make Round5 a leading candiate. This great performance in CPU and bandwidth is due to the following specific design choices:

- ***LWE vs LWR***: As expected, LWR leads to lower bandwidth requirements, as observed, e.g., when comparing `R5T0_`$\{l\}$`PKE` with Frodo [51], or Saber [28] with Kyber [63].

- ***Prime cyclotomic ring with q power of two***: This allows fine-tuning of parameters in Round5. For instance, NewHope [54] only offers two configurations for fixed n and q as required for the NTT optimized implementation. This forces NewHope to use the same parameters for its CPA and CCA configurations while Round5 can be configured with tailored parameter sets so that its CPA version provides better performance.

Table 2: Performance of Round5 C implementation on Intel Xeon Platinum 8168 CPU. For each scheme, columns from left to right represent respectively, the underlying hardness assumption, claimed quantum security level, failure probability (FP) during decryption, sizes of public key (PK) and ciphertext without DEM (CT) in bytes and finally CPU requirements for key generation (KG), encryption (Enc), and decryption (Dec) in 1000s of cycles. We are including some NIST PQC candidates for reference – security estimates and failure probability are according to the submissions and performance measurement was under identical conditions.

		PQ		Bandwidth		Kilo CPU Cycles		
Scheme	**Prob.**	**Sec.**	**FP**	**PK**	**CT**	**KG**	**Enc**	**Dec**
IND-CPA Security								
`R5ND_1KEM`	RLWR	2^{117}	2^{-76}	538	632	45.0	64.6	27.3
`R5ND_3KEM`	RLWR	2^{176}	2^{-75}	780	904	53.2	81.1	43.0
`R5ND_5KEM`	RLWR	2^{242}	2^{-64}	1050	1207	71.4	116.6	61.2
`R5T0_3KEM`	LWR	2^{182}	2^{-65}	7690	10279	23766	28461	378
`R5T1_3KEM`	LWR	2^{182}	2^{-65}	7690	10279	8811	17009	372
`R5T2_3KEM`	LWR	2^{182}	2^{-65}	7690	10279	6076	8913	369
NewHope512 [54]	RLWE	2^{101}	2^{-213}	928	1088	105.4	158.4	37.0
NewHope1024 [54]	RLWE	2^{233}	2^{-216}	1824	2176	201.5	313.6	73.1
IND-CCA Security								
`R5ND_1PKE`	RLWR	2^{120}	2^{-129}	562	672	34.9	69.2	65.3
`R5ND_3PKE`	RLWR	2^{181}	2^{-128}	810	1032	50.9	82.7	111.3
`R5ND_5PKE`	RLWR	2^{246}	2^{-129}	1140	1376	72.7	117.9	153.7
`R5T0_3PKE`	LWR	2^{176}	2^{-135}	8413	11290	27888	36408	36429
`R5T1_3PKE`	LWR	2^{176}	2^{-135}	8413	11290	12617	25264	25380
`R5T2_3PKE`	LWR	2^{176}	2^{-135}	8413	11290	18691	27617	28087
LAC128 [47]	RLWE	2^{133}	2^{-240}	544	1024	87.1	161.0	251.2
Saber [28]	MLWR	2^{180}	2^{-136}	992	1088	170.4	298.4	309.4
Kyber768 [63]	MLWE	2^{161}	2^{-142}	1088	1152	216.7	302.9	341.8
NewHope1024 [54]	RLWE	2^{233}	2^{-216}	1824	2208	239.3	371.2	425.6
Frodo640 [51]	LWE	2^{103}	2^{-149}	9616	9736	38560	38521	38692
Frodo976 [51]	LWE	2^{150}	2^{-200}	15632	15768	87722	89750	89539

– ***RLWR vs MLWR***: Saber [28] offers three configurations corresponding to ranks $\{2, 3, 4\}$ in a module lattice. In contrast, the scalability of Round5 allows finding fine-tuned parameters to fit *any* security target.

– ***Secret-key distribution***: Round5 and Frodo have similarities such as the usage of a modulus q that is a power of 2. When we compare the parameter set Frodo640, using AES-NI instructions, and `R5T0_3PKE` we observe that Round5 offers approximately $\times 3$ faster performance even if the generation of A is done in a similar way as in Frodo. The main reason for this faster performance is the choice of sparse-ternary secrets.

– ***Generation of A***: We observe that $\boldsymbol{f}_{d,n}^{(1)}$ allows for a 1.5x computational speed-up compared with $\boldsymbol{f}_{d,n}^{(0)}$ Round5 variants when SHAKE is used as the pseudo-random number generator. In this way, our `R5T1_3PKE` generic implementation achieves approximately $\times 1.5$ faster performance than Frodo976 when using AES-NI instructions.

– ***Unified design***: Round5 offers both IND-CPA and IND-CCA security notions relying on the same building blocks. Similarly, it is configurable to rely on a ring or non-ring structure. Thus, Round5 can fit multiple applications' needs. For instance, some applications require the efficiency of a ring-based IND-CPA secure construction, e.g., a fast VPN connection, while some users might dislike any approach based on structured lattices and need to ensure security against active attackers even if it comes at the price of a higher overhead.

8 Conclusions

In this paper, we presented the Round5 lattice-based cryptosystem. Round5 offers flexibility in the choice of the underlying problem (LWR or RLWR), security definition (IND-CPA or IND-CCA) and parameters, so that a wide variety of performance and security requirements can be met. On one hand, this allows Round5 to fit the needs of diverse applications. On the other hand, the unified design and implementation of Round5 allows for easy post-deployment adaptation of configuration parameters if future advances in cryptanalysis would require us to do so.

The use of (Ring)-LWR instead of LWE contributes to reduction of bandwidth requirements. In the ring case, the cyclotomic polynomial $\Phi_{n+1}(x)$ with $n+1$ prime is used as a reduction polynomial. This results in a large set of potential choices for n, satisfying various performance and security requirements. A further reduction of the bandwidth requirements for the ring case was obtained by the use of the XEf error-correcting codes, which by design are resistant to timing attacks.

We have shown that the ring variants of Round5 are faster than most comparable RLWE schemes. In the general lattice case the functions $\boldsymbol{f}^{(1)}_{d,n}$ and $\boldsymbol{f}^{(2)}_{d,n}$ allow for very fast generation of the public parameter $\boldsymbol{A}$, while both stopping precomputation and backdoor-like attacks.

Trust on Round5 comes from the fact that it relies on well-studied variants of the Learning with Rounding problem. We strengthen this aspect by providing proofs of both the security of Round5's schemes and of the hardness of the underlying problem.

References

1. Albrecht, M.R.: On dual lattice attacks against small-secret lwe and parameter choices in HElib and SEAL. Cryptology ePrint Archive, Report 2017/047 (2017), https://eprint.iacr.org/2017/047
2. Albrecht, M.R., Curtis, B.R., Deo, A., Davidson, A., Player, R., Postlethwaite, E.W., Virdia, F., Wünderer, T.: Estimate all the LWE, NTRU schemes! Cryptology ePrint Archive, Report 2018/331 (2018), https://eprint.iacr.org/2018/331
3. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - a new hope. Cryptology ePrint Archive, Report 2015/1092 (2015), https://eprint.iacr.org/2015/1092
4. Alwen, J., Krenn, S., Pietrzak, K., Wichs, D.: Learning with rounding, revisited: New reduction, properties and applications. Cryptology ePrint Archive, Report 2013/098 (2013), https://eprint.iacr.org/2013/098
5. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems, pp. 595–618. Springer (2009), https://doi.org/10.1007/978-3-642-03356-8_35
6. Baan, H., Bhattacharya, S., Garcia-Morchon, O., Rietman, R., Tolhuizen, L., Torre-Arce, J.L., Zhang, Z.: Round2: Kem and pke based on glwr. Cryptology ePrint Archive, Report 2017/1183 (2017), https://eprint.iacr.org/2017/1183
7. Babai, L.: On Lovász lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986), https://doi.org/10.1007/BF02579403
8. Bai, S., Galbraith, S.D.: Lattice decoding attacks on binary LWE. Cryptology ePrint Archive, Report 2013/839 (2013), https://eprint.iacr.org/2013/839
9. Bai, S., Langlois, A., Lepoint, T., Sakzad, A., Stehlé, D., Steinfeld, R.: Improved security proofs in lattice-based cryptography: using the rényi divergence rather than the statistical distance. Cryptology ePrint Archive, Report 2015/483 (2015), https://eprint.iacr.org/2015/483
10. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. Cryptology ePrint Archive, Report 2011/401 (2011), https://eprint.iacr.org/2011/401
11. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes, pp. 26–45. Springer (1998), https://doi.org/10.1007/BFb0055718
12. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. pp. 62–73. CCS '93, ACM (1993), https://doi.org/10.1145/168588.168596
13. Bernstein, D.J.: A subfield-logarithm attack against ideal lattices (February 2014), available from https://blog.cr.yp.to/20140213-ideal.html

14. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: NTRU prime: reducing attack surface at low cost. Cryptology ePrint Archive, Report 2016/461 (2016), https://eprint.iacr.org/2016/461
15. Bhattacharya, S., Garcia-Morchon, O., Rietman, R., Tolhuizen, L.: spKEX: An optimized lattice-based key exchange. Cryptology ePrint Archive, Report 2017/709 (2017), https://eprint.iacr.org/2017/709
16. Bogdanov, A., Guo, S., Masny, D., Richelson, S., Rosen, A.: On the hardness of learning with rounding over small modulus. Cryptology ePrint Archive, Report 2015/769 (2015), https://eprint.iacr.org/2015/769
17. Boneh, D., Dagdelen, O., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. Cryptology ePrint Archive, Report 2010/428 (2010), https://eprint.iacr.org/2010/428
18. Bos, J., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. Cryptology ePrint Archive, Report 2016/659 (2016), https://eprint.iacr.org/2016/659
19. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Stehlé, D.: CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM. Cryptology ePrint Archive, Report 2017/634 (2017), https://eprint.iacr.org/2017/634
20. Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In: 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015. pp. 553–570 (2015), https://doi.org/10.1109/SP.2015.40
21. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive, Report 2011/277 (2011), https://eprint.iacr.org/2011/277
22. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing. pp. 575–584. STOC '13, ACM (2013), https://doi.org/10.1145/2488608.2488680
23. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. LNCS, vol. 7073, pp. 1–20. Springe (2011), https://doi.org/10.1007/978-3-642-25385-0_1
24. Cheon, J.H., Han, K.H., Kim, J., Lee, C., Son, Y.: A practical post-quantum public-key cryptosystem based on spLWE. Cryptology ePrint Archive, Report 2016/1055 (2016), https://eprint.iacr.org/2016/1055
25. Cheon, J.H., Kim, D., Lee, J., Song, Y.: Lizard: Cut off the tail! practical post-quantum public-key encryption from LWE and LWR. Cryptology ePrint Archive, Report 2016/1126 (2016), https://eprint.iacr.org/2016/1126
26. Coron, J.S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård Revisited: How to Construct a Hash Function. In: Shoup, V. (ed.) Advances in Cryptology – CRYPTO 2005. pp. 430–448. Springer (2005)
27. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. Cryptology ePrint Archive, Report 2001/108 (2001), https://eprint.iacr.org/2001/108
28. D'Anvers, J.P., Karmakar, A., Roy, S.S., Vercauteren, F.: SABER. Tech. rep., National Institute of Standards and Technology (2017), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions
29. D'Anvers, J.P., Karmakar, A., Roy, S.S., Vercauteren, F.: Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. Cryptology ePrint Archive, Report 2018/230 (2018), https://eprint.iacr.org/2018/230

30. Ducas, L., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS – Dilithium: Digital signatures from module lattices. Cryptology ePrint Archive, Report 2017/633 (2017), https://eprint.iacr.org/2017/633
31. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. Cryptology ePrint Archive, Report 2013/383 (2013), https://eprint.iacr.org/2013/383
32. ETSI: "ETSI launches Quantum Safe Cryptography specification group" (March 2015), http://www.etsi.org/news-events/news/947-2015-03-news-etsi-launches-quantum-safe-cryptography-specification-group
33. ETSI: Terms of reference for ETSI TC cyber working group for quantum-safe cryptography (ETSI TC cyber WG-QSC) (2017), https://portal.etsi.org/TBSiteMap/CYBER/CYBERQSCToR.aspx, accessed: 15-02-2017
34. FIPS: Specification for the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197 (November 2001), https://doi.org/10.6028/NIST.FIPS.197
35. FIPS: SHA-3 standard: Permutation-based hash and extendable-output functions. Federal Information Processing Standards Publication 202 (August 2015), https://doi.org/10.6028/NIST.FIPS.202
36. Fritzmann, T., Pöppelmann, T., Sepulveda, J.: Analysis of error-correcting codes for lattice-based key exchange. Cryptology ePrint Archive, Report 2018/150 (2018), https://eprint.iacr.org/2018/150
37. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. pp. 212–219. STOC '96, ACM (1996), https://doi.org/10.1145/237814.237866
38. Hamburg, M.: Graphs of "estimate all the LWE, NTRU schemes!" indexed to the "pqc lounge" data. (2017), https://bitwiseshiftleft.github.io/estimate-all-the-lwe-ntru-schemes.github.io/graphs
39. Hoffstein, J., Pipher, J., Schanck, J.M., Silverman, J.H., Whyte, W., Zhang, Z.: Choosing parameters for NTRUEncrypt (2017), https://doi.org/10.1007/978-3-319-52153-4_1
40. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. Cryptology ePrint Archive, Report 2017/604 (2017), https://eprint.iacr.org/2017/604
41. Howgrave-Graham, N.: A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU. In: Menezes, A. (ed.) Advances in Cryptology - CRYPTO 2007: 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007. Proceedings. pp. 150–169. Springer (2007), https://doi.org/10.1007/978-3-540-74143-5_9
42. Hülsing, A., Rijneveld, J., Schanck, J.M., Schwabe, P.: High-speed key encapsulation from NTRU. In: Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. pp. 232–252 (2017), https://doi.org/10.1007/978-3-319-66787-4_12
43. Kent, S., Seo, K.: Security Architecture for the Internet Protocol. RFC 4301, RFC Editor (December 2005), http://www.rfc-editor.org/rfc/rfc4301.txt
44. Laarhoven, T.: Search problems in cryptography. Ph.D. thesis, Eindhoven University of Technology (2015)
45. Laarhoven, T., Mosca, M., van de Pol, J.: Finding shortest lattice vectors faster using quantum search. Cryptology ePrint Archive, Report 2014/907 (2014), https://eprint.iacr.org/2014/907

46. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Cryptology ePrint Archive, Report 2012/090 (2012), https://eprint.iacr.org/2012/090
47. Lu, X., Liu, Y., Jia, D., Xue, H., He, J., Zhang, Z.: LAC. Tech. rep., National Institute of Standards and Technology (2017), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions
48. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. Cryptology ePrint Archive, Report 2012/230 (2012), https://eprint.iacr.org/2012/230
49. Maurer, U., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. Cryptology ePrint Archive, Report 2003/161 (2003), https://eprint.iacr.org/2003/161
50. Micciancio, D., Regev, O.: Lattice-based cryptography, pp. 147–191. Springer (2009)
51. Naehrig, M., Alkim, E., Bos, J., Ducas, L., Easterbrook, K., LaMacchia, B., Longa, P., Mironov, I., Nikolaenko, V., Peikert, C., Raghunathan, A., Stebila, D.: FrodoKEM. Tech. rep., National Institute of Standards and Technology (2017), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions
52. NIST: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. POST-QUANTUM CRYPTO STANDARDIZATION. Call For Proposals Announcement (2016), http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-final-dec-2016.pdf
53. Peikert, C., Regev, O., Stephens-Davidowitz, N.: Pseudorandomness of Ring-LWE for any ring and modulus. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017. pp. 461–473 (2017), https://doi.org/10.1145/3055399.3055489
54. Pöppelmann, T., Alkim, E., Avanzi, R., Bos, J., Ducas, L., de la Piedra, A., Schwabe, P., Stebila, D.: NewHope. Tech. rep., National Institute of Standards and Technology (2017), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions
55. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) Advances in Cryptology — CRYPTO '91: Proceedings. pp. 433–444. Springer (1992), https://doi.org/10.1007/3-540-46766-1_35
56. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing. pp. 84–93. STOC '05, ACM (2005), https://doi.org/10.1145/1060590.1060603
57. Regev, O.: The learning with errors problem (invited survey). In: Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, June 9-12, 2010. pp. 191–204 (2010), https://doi.org/10.1109/CCC.2010.26
58. Rosen, E., Rekhter, Y.: BGP/MPLS ip virtual private networks (vpns). RFC 4364, RFC Editor (February 2006)
59. Saarinen, M.J.O.: Ring-LWE ciphertext compression and error correction: Tools for lightweight post-quantum cryptography. In: Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security. pp. 15–22. IoTPTS '17, ACM (April 2017), https://eprint.iacr.org/2016/1058

60. Saarinen, M.J.O.: HILA5: On reliability, reconciliation, and error correction for Ring-LWE encryption. In: Adams, C., Camenisch, J. (eds.) SAC 2017. Lecture Notes in Computer Science, vol. 10719, pp. 192–212. Springer (2018)
61. Saarinen, M.J.O., Bhattacharya, S., Garcia-Morchon, O., Rietman, R., Tolhuizen, L., Zhang, Z.: Shorter messages and faster post-quantum encryption with Round5 on Cortex M. Submitted for Publication (2018), https://eprint.iacr.org/2018/723
62. Schnorr, C.P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Math. Program. **66**(2), 181–199 (Sep 1994), https://doi.org/10.1007/BF01581144
63. Schwabe, P., Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Seiler, G., Stehlé, D.: CRYSTALS-KYBER. Tech. rep., National Institute of Standards and Technology (2017), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions
64. Singh, V.: A practical key exchange for the internet using lattice cryptography. Cryptology ePrint Archive, Report 2015/138 (2015), https://eprint.iacr.org/2015/138
65. Smart, N.P., Albrecht, M.R., Lindell, Y., Orsini, E., Osheter, V., Paterson, K., Peer, G.: LIMA. Tech. rep., National Institute of Standards and Technology (2017), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions
66. Wunderer, T.: Revisiting the hybrid attack: Improved analysis and refined security estimates. Cryptology ePrint Archive, Report 2016/733 (2016), https://eprint.iacr.org/2016/733

A Features of Round2, Hila5 and Round2

Table 3 summarizes design features of Round2 [6] and Hila5[60], and their application in Round5.

Prime cyclotomic polynomials were selected because they allow for a fine-grained optimization over the degree. By stipulating that $\Phi_{n+1}(x)$ is irreducible modulo two, we hedge against possible vulnerabilities in power-of-2 cyclotomic rings [13,14]. As shown in [53], decisional RLWE over this ring remains hard for any modulus, including the power-of-2 moduli q, p as used in Round5. As explained in [6, Sec. 2.9], multiplication in $\mathbb{Z}_q[x]/\Phi_{n+1}(x)$ can be implemented by lifting polynomials to the ring $\mathbb{Z}_q[x]/(x^{n+1}-1)$ and operating in this ring.

Hila5 operates over the ring $Z_q[x]/(x^{1024}+1)$, where $q = 12289 = 3 * 2^{12} + 1$, which allows to use a number-theoretic transform (NTT) for efficient implementation. Round2 supports NTT-friendly parameters as well. With the choice for prime cyclotomic polynomials and moduli that are a power of two, implementations are already so efficient that support of NTT was not considered necessary.

The sparse ternary secret-key distribution was selected to simplify failure analysis, especially in the ring-case. Round5 uses balanced sparse ternary secrets, that is, the number of plus ones and minus ones in any column of the secret matrices are equal.

The Round5 parameter choices result in security levels that comply with the NIST levels. Round5 uses rounding constants to implement the GLWR problem and to prove the IND-CPA security of the CPA-PKE building block.

Table 3: Design features of Round2, Hila5, and Round5.

Feature	**Round2**	**Hila5**	**Round5**
Functionality	Key encapsulation and PKE	Key encapsulation	Key encapsulation and PKE
Underlying problem	GLWR	R-LWE	GLWR
Unified design	Yes	Only ring	Yes
Secret distribution	Balanced sparse ternary	Centered binomial	Balanced sparse ternary
Supported NIST levels	All	5	1,3,5
Error Correction	No	XE code	XE code
Ring choice	Prime cyclotomic	$x^{1024}+1$ "negacyclic"	Prime cyclotomic
Support NTT Arithmetic	Yes	Yes	No
Underlying method	Noisy El-Gamal	Noisy Diffie-Hellman	Noisy El-Gamal
"Safe bits"	No	Yes	No