Round5: Optimizing Bandwidth and Performance in Rounding-based Public Key Encryption

Sauvik Bhattacharya, Oscar Garcia-Morchon, Ronald Rietman, Markku-Juhani O. Saarinen, Ludo Tolhuizen, Zhenfei Zhang, Hayo Baan, and Jose-Luis Torre-Arce

***Abstract*—International standardization bodies such as NIST, ETSI, and IETF are currently actively seeking quantum resistant alternatives to RSA and Elliptic Curve based public key cryptography. In this context, we present the lattice-based cryptosystem Round5 consisting of a key-encapsulation mechanism and a public-key encryption scheme. Round5 is based on the General Learning with Rounding (GLWR) problem, unifying older special case instance problems into one. Usage of rounding leads to significantly reduced bandwidth and randomness requirements. The parameters of the schemes have been optimized for bandwidth, while the design facilitates very efficient implementation. Round5's schemes share common building blocks, simplifying (security and operational) analysis and code review. Round5's reliance on prime cyclotomic rings offers a large design space allowing fine-grained parameter optimization. The use of sparse-trinary secret keys improves performance and decryption success rates at minimal additional cost. The use of error-correcting codes increases the decryption success rate, allowing further optimization of other parameters. Finally, Round5 proposes various approaches of refreshing the system public parameter $\boldsymbol{A}$, which efficiently prevent precomputation and back-door attacks.**

***Index Terms*—Lattice cryptography, Post-quantum Cryptography, Learning With Rounding, Prime cyclotomic ring, Key encapsulation, CCA Security, CPA Security**

I. 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 confidentialty 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 [1] and ETSI [2], [3] 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. There exist lattice-based proposals for key exchange [4]–[6], key encapsulation [7] [8], public-key encryption [9], [10] and digital signatures [11], [12]. The main hard problem underlying the security of most lattice-based proposals is the Learning with Errors (LWE) problem defined on general *Euclidean* lattices.

Sauvik Bhattacharya, Ronald Rietman, Ludo Tolhuizen are with Philips Research, Eindhoven, The Netherlands. Oscar Garcia-Morchon is with Philips IP&S, Eindhoven, The Netherlands. Email: firstname.lastname@philips.com. Markku-Juhani O. Saarinen is with PQShield Ltd., Oxford, UK. Email: mjos@mjos.fi. Zhenfei Zhang is with Onboard Security Inc., Wilmington, USA. Email: zzhang@onboardsecurity.com.

For a public parameter $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$, the LWE problem refers to distinguishing uniform samples $(\boldsymbol{A}, \boldsymbol{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ from samples of the form $(\boldsymbol{A}, \boldsymbol{As} + \boldsymbol{e})$ where $\boldsymbol{A}$ is uniform on $\mathbb{Z}_q^{m \times n}$, the secret $\boldsymbol{s}$ is drawn uniformly from $\mathbb{Z}_q^n$, and $\boldsymbol{e}$ is drawn from a known error distribution on the integers. Matrix multiplication and vector addition are performed modulo q.

The ring variant of LWE (RLWE) introduces more structured *Ideal* lattices [13] for better performance. *Module* lattices [14] allow for additional flexibility in the parameter choice and are structurally in between the former two.

In the Learning with Rounding (LWR) problem [15], the independent, randomly drawn error $\boldsymbol{e}$ from LWE is replaced by a deterministic error via rounding $\boldsymbol{As}$ to a smaller modulus p. The LWR problem is to distinguish uniform samples $(\boldsymbol{A}, \boldsymbol{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, from samples of the form $(\boldsymbol{A}, \left\lfloor \frac{p}{q} \boldsymbol{As} \right\rceil)$, where $\boldsymbol{A}$ uniform on $\mathbb{Z}_q^{m \times n}$, $\mathbf{s}$ 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: ciphertexts are smaller, as they have p-ary symbols instead of q-ary symbols, and there is no need to explicitly generate the components of the noise vector $\boldsymbol{e}$.

A. Separate Solutions for LWE and RLWE

Due to the structure of ideal lattices, the hardness assumptions for ring learning with errors (RLWE) and ring learning with rounding (RLWR) are considered less conservative than for LWE and LWR. On the other hand, RLWE and RLWR are more efficient than their non-ring counterparts [16]. However, no scheme has been fully defined with the flexibility of fitting diverse use-cases with diverse trust requirements, e.g., Ring-LWE against LWE assumptions. For some use-cases with critical performance requirements IND-CPA (indistinguishability under chosen plaintext attack [17]) security may be enough, in which cases designing for only slower IND-CCA (indistinguishability under chosen ciphertext attack [17]) security might be over-provisioning.

We will now give some examples of applications and their particular requirements. A high-performance IPSec [18] 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. However a governmental VPN [19] 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 [5] is based on the conservative LWE assumption, but is rather inefficient for performance-intensive scenarios, requiring bandwidth as high as 23 kilobytes. Kyber [7] is comparatively efficient but depends on one single underlying ring choice. Most schemes such as [5], [7], [8] or [20] are defined to provide either IND-CPA or IND-CCA security, but not both.

B. Inflexibility in Ring Selection

Choice of ring greatly affects the performance of schemes based on ideal lattices. A common choice [4], [6] 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 the $2n$-th cyclotomic polynomial $\Phi_{2n}(x) = x^n + 1$. 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 lower bandwidth requirements. This choice is reflected in proposals like Kyber [7], NTRUEncrypt [21], NTRU-KEM [22], SABER [8] and more. Kyber [7] and SABER [8] use modules of rank $k = 3$ over $\mathbb{Z}_q[x]/(x^{256}+1)$, so that the underlying module lattice problem relates to the hardness 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 [23, p. 6], the decisional RLWE problem over this ring is easy.

C. Parameter Selection: Prior Work

Applebaum et al. [24] showed that the LWE secrets $\boldsymbol{s}$ can be drawn according to the same distribution as the errors without impacting hardness. [25] 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 [9], [10] to propose schemes where the LWE secrets are sparse and trinary. NTRUPrime [26] also utilizes rounding and sparse-trinary secrets. The decryption/decapsulation failure probability can be further reduced by using error-correcting codes. The analysis in [27] 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 [4]. Other proposals [5]–[7] 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 Euclidean lattice-based schemes that must generate n^2 elements. We offer solutions to this problem in our work.

D. 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 II-C.) 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.** As in [28], NTRU-KEM [22], and [20], 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 *largeer design space*, allowing better parameter optimization.

- **Designed for Performance.** Round5 is designed to be highly practical: its use of rounding results in some of the smallest key and ciphertext sizes in lattice-based cryptography [29], and requires less randomness. As the moduli q and p are powers of two, modular operations can be implemented efficiently. Furthermore the use of trinary secrets and error correction codes leads to significant reduction in failure rate without compromising performance.

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

II. Preliminaries

Let $\mathbb{Z}$ and $\mathbb{Z}_a$ denote respectively the ring of rational integers, and for an integer $a \geq 1$ 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 $x \equiv x' \pmod{\alpha}$.

A. 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 $\mathcal{R}_n$ *trinary* 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 $\mathcal{R}_n$.

B. 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 trinary 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 trinary 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 trinary 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 the distribution χ_S defined over the set $\mathcal{H}_{n,d/n}(h)$.

C. Underlying Problem and Hardness Assumption

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 D_s be a probability distribution on $\mathcal{R}_n^{d/n}$.*

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

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

When $n = 1$, the GLWR problem is equivalent to the LWR problem [15] with dimension d, large modulus q, and rounding modulus p. Setting the distribution $D_s = \mathcal{U}(\mathcal{H}_{1,d}(h))$ further specializes the GLWR problem to the LWR problem with sparse-trinary secrets, denoted as $\mathtt{LWR}_{\mathrm{spt}}$. The hardness of the LWR problem has been studied in [15], [30]–[32] and established based on the hardness of the Learning with Errors (LWE) problem [33]. The most recent reductions are due to [32, Theorem 6.4] (that preserves the dimension n between the two problems) and [31, 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 V-C.

When $n = d \geq 1$ is such that $n+1$ is prime, and $\mathcal{R}_{n,q} = \mathbb{Z}_q[x]/(\Phi_{n+1}(x))$, the GLWR problem is equivalent to the Ring LWR problem defined on $\Phi_{d+1}(x)$, dimension d, large modulus q and rounding modulus p. Setting $D_s = \mathcal{U}(\mathcal{H}_{d,1}(h))$ further specializes it to the RLWR problem with sparse-trinary secrets, denoted as $\mathtt{RLWR}_{\mathrm{spt}}$. We are only aware of a reduction from Decision-RLWE to Decision-RLWR due to [15, 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., $q \geq pBn^{\omega(1)}$. The last condition may be too restrictive for practical schemes. Hence, although [15, Theorem 3.2] is relevant for the 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:

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

The decision-GLWR problem with sparse-trinary secrets 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

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

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:

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

The advantage of an adversary for this extended form of the decision-GLWR problem is defined in a similar manner as above. For brevity, when $D_s = \mathcal{U}\left(\mathcal{H}_{n,d/n}(h)\right)$, we denote the $\mathtt{GLWR}_{d,n,m,q,p}\left(\mathcal{U}\left(\mathcal{H}_{n,d/n}(h)\right)\right)$ problem (with sparse-trinary secrets), as $\mathtt{GLWR}_{\mathrm{spt}}$. When the secret distribution D_s is the uniform one over $\mathcal{R}_{n,q}^{d/n}$, it is omitted from notation.

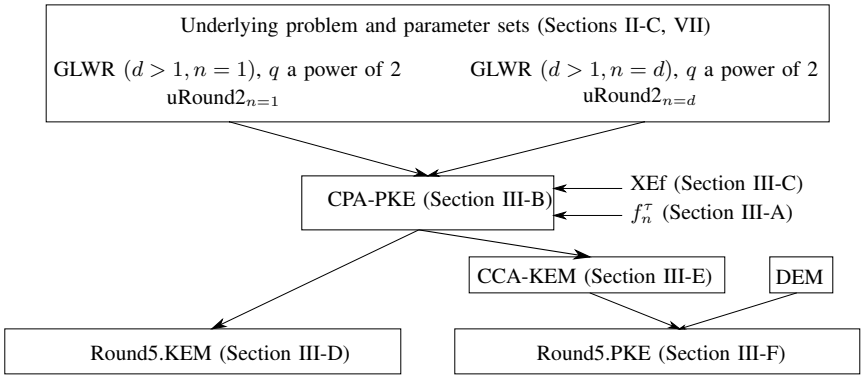

Figure 1: Overview of Round5.

III. 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 III-D), an IND-CPA secure key-encapsulation mechanism (KEM).
- **Round5.PKE** (Section III-F), an IND-CCA secure public-key encryption algorithm (PKE).

Section III-A describes options for generating Round5's public parameter $\boldsymbol{A}$. The public schemes are derived from internal building blocks; We fist describe *CPA-PKE* (Section III-B), and the error correction code XEf (Section III-C), leading to Round5.KEM (Section III-D). We then apply a KEM variant [34] of the Fujisaki-Okamoto transform to CPA-PKE, to obtain a key encapsulation mechanism *CCA-KEM* (Section III-E), that is IND-CCA secure in the classical and quantum ROM model [35], [36]. Round5.PKE is obtained by combining CCA-KEM with a secure one-time symmetric-key encryption scheme (Section III-F). Details of the IND-CPA and IND-CCA [37] security properties of CPA-PKE, CCA-KEM, and Round5.PKE are discussed later in Section V.

A. Internal building block: Definitions of $f_n^\tau(\sigma)$

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 σ. Round5 gives three options for $f_n^\tau(\sigma)$, the function used to compute $\boldsymbol{A}$ in CPA-PKE.Keygen and CPA-PKE.Encrypt. Given positive integers d, n, q, μ, B and $\tau \in \{0, 1, 3\}$, f_n^τ is a mapping from $\{0,1\}^{\mu B}$ to $\mathcal{R}_{n,q}^{d/n \times d/n}$. The functions f_n^1, f_n^2 are applied in the cases when $n = 1$; the function f_n^3 is only applied if $n = d$. To emphasize this, the notation $f_{n=1}^\tau$ is used for $\tau = 0, 1$, and $f_{n=d}^3$ instead of f_n^3 will be used.

The three options for f_n^τ are defined as follows:

1) In $f_{n=1}^1(\sigma)$, a new $\boldsymbol{A}$ is derived using a seed σ from a DBRG for each protocol instantiation, similar to [5].
2) In $f_{n=1}^2(\sigma)$, a new $\boldsymbol{A}$ is derived using permutations on a long-term matrix $\boldsymbol{A}_{\text{master}} \in \mathbb{Z}_q^{d \times d}$.
3) In $f_{n=d}^3(\sigma)$ creates $\boldsymbol{A}$ from a vector $\boldsymbol{a}_{\text{master}} \in \mathbb{Z}_q^d$ specific to each protocol interaction, similar to [6].

All three f_n^τ options stop both backdoor-like and precomputation attacks. Section V-A contains a discussion on the role of f_n^τ in the provable security of Round5.

B. 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 trinary 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 $\boldsymbol{R}$ 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 secure (see Section III-D), and for IND-CCA secure (see Section III-E) cryptographic constructions. Furthermore, CPA-PKE uses five rounding constants. These, combined with rounding downwards, implement all of the actual rounding operations in its algorithms. The rounding constants are:

1) Matrix $\boldsymbol{H}_1 \in \mathcal{R}_{n,q}^{d/n \times \overline{n}}$ whose coefficients are set to $q/2p$. This constant is equivalent standard rounding in terms of rounding downwards since by definition for $x \in \mathbb{Z}_q$, $\lfloor (p/q) \cdot \{x + (q/2p)\} \rfloor \equiv \lfloor (p/q) \cdot x \rceil$. This is done to simplify the proof of IND-CPA security for CPA-PKE and Round5.KEM. See Section V.

2) Matrix $\boldsymbol{H}_2 \in \mathcal{R}_{n,q}^{d/n \times \overline{m}}$ and vector $\boldsymbol{h}_3 \in \mathbb{Z}_q^{\mu}$ whose coefficients are set 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 V).
3) Matrix $\boldsymbol{H}_4 \in \mathcal{R}_{n,q}^{d/n \times \overline{m}}$ and vector $\boldsymbol{h}_5 \in \mathbb{Z}_q^{\mu}$. Coefficients of $\boldsymbol{H}_4$ are set to $\left(\frac{q}{2p} - \frac{q}{2z}\right)$ and coefficients of $\boldsymbol{h}_5$ are set to $\left(\frac{q^2}{2pz} - \frac{q}{2t} - \frac{q}{2^{B+1}}\right)$, both for avoiding bias in the decryption/decapsulation error in Round5.

Algorithm 1: CPA-PKE.Keygen()

Choose $\tau \in \{0, 1, 3\}$
$\sigma \xleftarrow{\$} \{0,1\}^{\kappa}$
$\boldsymbol{A} = f_n^{\tau}(\sigma)$
$\boldsymbol{S} \leftarrow \chi_S^{\overline{n}}$
$\boldsymbol{B} = \langle \left\lfloor \frac{p}{q} \cdot \langle \boldsymbol{AS} + \boldsymbol{H_1} \rangle_q \right\rfloor \rangle_p$
$pk = (\tau, \sigma, \boldsymbol{B})$
$sk = \boldsymbol{S}$
return (pk, sk)

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

$\boldsymbol{A} = f_n^{\tau}(\sigma)$
$\boldsymbol{R} = f_R(\rho)$
$\boldsymbol{U} = \langle \left\lfloor \frac{p}{q} \cdot \langle \boldsymbol{A}^T \boldsymbol{R} + \boldsymbol{H_2} \rangle_q \right\rfloor \rangle_p$
$\boldsymbol{v} =$
$\langle \left\lfloor \frac{t}{p} \cdot \left\langle \text{Sample}_{\mu}(\boldsymbol{B^T R}) + \boldsymbol{h}_3 \right\rangle_p \right\rfloor + \frac{t}{2^B} ECC\_Enc_y(m) \rangle_t$
$c = (\boldsymbol{U}, \boldsymbol{v})$
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 $\hat{m}$

C. 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 [27], 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 [38] (2-bit correction) and HILA5 [39] (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 $kappa$. 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 $\text{lcm}(l_i, l_j) \geq \kappa$ 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.

D. Round5.KEM

Round5.KEM samples a fresh, uniformly random κ-long bitstring and encapsulates it in a ciphertext by directly using CPA-PKE.Encrypt (Algorithm 2) in a black-box manner. A hash function $H : \{0,1\}^* \to \{0,1\}^{\kappa}$ (e.g. SHAKE256 [40]) is then used to hash this bitstring, along with the ciphertext (ensuring that the final key has contributions from both parties), into the final key. For key-generation and decapsulation, Round5.KEM directly reuses the corresponding algorithms from CPA-PKE (Algorithms 1 and 3).

E. Internal building block: CCA-KEM

On a high level, CCA-KEM, a building block for Round5. PKE, also samples a fresh, random bitstring that it then encapsulates in a ciphertext, and from both of which it derives the final key. It is actively secure, being constructed by applying a standard KEM variant [34] of the Fujisaki-Okamoto transform on CPA-PKE, similarly as in [7, Sec. 4], for example. In addition to the hash function H from Round5.KEM, CCA-KEM uses another $G : \{0,1\}^* \to \{0,1\}^{\kappa} \times \{0,1\}^{\kappa} \times \{0,1\}^{\kappa}$, also usually derived from SHAKE256 [40]. On decapsulation failure, CCA-KEM returns a pseudorandom key, causing later protocol steps to implicitly fail. Explicit failure would complicate analysis, especially in the QROM case.

F. Round5.PKE

Round5.PKE is constructed by combining CCA-KEM with a data encapsulation mechanism (DEM), in the canonical way proposed by Cramer and Shoup [41]. 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. During decryption, CCA-KEM is used to decapsulate K, which is then used by the DEM to decrypt and authenticate the plaintext. We omit the details – applications may choose the symmetric algorithms for implementing DEM based on their particular needs.

IV. 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

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

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, $\boldsymbol{B} = \langle (p/q)\,(\boldsymbol{A}\boldsymbol{S} + \boldsymbol{H_1}) - (p/q)\boldsymbol{I}_B \rangle_p$, 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

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

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

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

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 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 trinary 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 trinary polynomial, $w_{j,k}(s) \in \{-2,-1,0,1,2\}$. For a trinary 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

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

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 sastisfy

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

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

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 $|\mu_Y| = |\frac{1}{2}\sum_i iw_i| \le 2h$, 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

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

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

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

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

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 indpendently 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 $\mu \cdot p_f$.

V. 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 V-A. Section V-B 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 V-C, we give a proof of hardness of Round5's underlying problem – the decision GLWR problem with sparse-trinary secrets, assuming the hardness of decision Learning with Errors with uniform secrets and Gaussian errors [33].

A. Deterministic generation of $\boldsymbol{A}$

The General Learning with Rounding (GLWR) public parameter $\boldsymbol{A}$ in Round5 is generated using the function f_n^τ from a short random seed (see Section III-A). The core component in f_n^τ responsible for deterministically expanding this short random seed into a longer random sequence is either AES256 or SHAKE256. In order to relate Round5's security to the hardness of the GLWR problem, we reuse Naehrig et al.'s argument in [42] 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 [43], [44, Def. 3], in exactly the same manner as in [42, Sec. 5.1.4].

In the case of SHAKE256, the proof of security applies to all instantiations of f_n^τ. In case of AES256, the proof applies directly to the instantiations f_n^0 and f_n^3, it applies also to f_n^1 when the function permutes complete AES blocks. We refer to [42, Sec. 5.1.4] for details.

B. 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-trinary 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-trinary secrets, assuming $t|p|z|q$ for $z = \max(p, tq/p)$. More precisely, for every IND-CPA adversary $\mathcal{A}$, if $Adv_{\text{CPA-PKE}}^{\text{IND-CPA}}(\mathcal{A})$ is the advantage in winning the IND-CPA game, then there exist adversaries $\mathcal{D}$ and reduction algorithms $\mathcal{C}', \mathcal{E}'$ such that*

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

$Adv_{d,n,m,q_1,q_2}^{dGLWR_{spt}}(\mathcal{Z})$ is the advantage of adversary $\mathcal{Z}$ in distinguishing m GLWR samples (with sparse-trinary 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: $\text{Adv}_{\text{CPA-PKE}}^{\text{IND-CPA}}(\mathcal{A}) = |\Pr(S_0) - 1/2|$.

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: $|\Pr S_0 - \Pr(S_1)| \le \overline{n} \cdot \text{Adv}_{d,n,\frac{d}{n},q,p}^{\text{dGLWR}_{\text{spt}}}(\mathcal{A} \circ \mathcal{C}')$.

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: $\text{Adv}^{f_R}(\mathcal{D}) \ge |\Pr(S_1) - \Pr(S_2)|$.

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 [45].

Game 4 For $z = max(p, tq/p)$, we define $\boldsymbol{U}' = \left\langle \left\lfloor \frac{z}{q}\cdot\left\langle \boldsymbol{A}^T\boldsymbol{R}+\boldsymbol{H}_2\right\rangle_q\right\rfloor\right\rangle_z$ and $\boldsymbol{v}'' = \left\langle \text{Sample}_\mu(\left\lfloor \frac{z}{q}\cdot\left\langle \boldsymbol{B}_q^T\boldsymbol{R}+\boldsymbol{H}_3\right\rangle_q\right\rfloor)+\frac{pz}{q2^B}\cdot m_b\right\rangle_z$.

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. Distinguishing between this and game 5 leads to a distinguisher $\mathcal{E}'$ between the uniform and GLWR distribution (with parameters as follows): $|\Pr(S_5)-\Pr(S_6)| \le \overline{m}\cdot \mathrm{A}^{\mathrm{dGLWR_{spt}}}_{d,n,\frac{d}{n}+\overline{n},q,z}(\mathcal{A}\circ\mathcal{E}')$. 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.

□

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, when the hash functions H and G are modeled as random oracles, the IND-CCA security of CCA-KEM under the decision GLWR assumption with sparse-trinary secrets, follows directly from the IND-CPA security of CPA-PKE that is used to construct it using the standard KEM variant of the Fujisaki-Okamoto transform, in both the classical and quantum random oracle model (see for example, [7, Th. 4.1, 4.2]). Note however, that the latter reduction is not tight; this is an open problem. Finally, as Round5.PKE is constructed from the key-encapsulation mechanism CCA-KEM and a secure data-encapsulation mechanism in the canonical way [41], its IND-CCA security follows directly from that of CCA-KEM.

We omit details for reasons of space.

C. Hardness of Sparse-Trinary LWR

In this section, we prove that the Decision-LWR problem with sparse-trinary 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 2. *Let $k,p,q\ge 1$ and $m\ge n\ge h\ge 1$ be integers such that p divides q, and $k\ge m' = \frac{q}{p}\cdot m$. Let $\epsilon\in(0,\frac{1}{2})$, and $\alpha,\delta>0$ such that*

$$\alpha \ge q^{-1}\sqrt{\left(\frac{2}{\pi}\right)\ln(2n(1+\epsilon^{-1}))},\ \binom{n}{h}2^h \ge q^{k+1}\cdot\delta^{-2},\quad m = O\left(\frac{\log n}{\alpha\sqrt{10h}}\right). \tag{14}$$

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 $(\zeta-\delta)/(3m') - 41\epsilon/2 - \sum_{s|q,s\text{ prime}} s^{-k-1}$. 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 2 and Lemma 5 with $\alpha' = \alpha\sqrt{10h}$.

Step 1: Reduction from LWE with secrets in $\mathbb{Z}_q$ and Gaussian errors to Sparse-trinary LWE.: In [10, Theorem 1], specializing [9, 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 [25, Theorem 4.1], the following holds.

Lemma 2. *Let $k,q\ge 1$ and $m\ge n\ge h\ge 1$ be integers, and let $\epsilon\in(0,\frac{1}{2})$, and $\alpha,\delta>0$ such that $\alpha \ge q^{-1}\sqrt{(2/\pi)\ln(2n(1+\epsilon^{-1}))}$, and $\binom{n}{h}2^h \ge q^{k+1}\cdot\delta^{-2}$. 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 $(\zeta-\delta)/(3m) - 41\epsilon/2 - \sum_{s|q,s\text{ prime}} s^{-k-1}$.*

Step 2: Reduction from Sparse-trinary LWE to Sparse-trinary LWR.: Bai et al. provide in [32, Theorem 6.4] a reduction from LWE with Gaussian noise to LWR, that is based on two independent reductions. One of these reductions [32, 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 [32, 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 [32, Theorem 5.1] that describes a reduction from LWE with Gaussian noise and sparse trinary secrets to LWR with sparse-trinary secrets. U_B denotes the continuous uniform distribution in $[-B,\ldots,B]$.

Lemma 3 (Adapted from [32, 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 $m > \log\left(\binom{n}{h}2^h\right)/\log\left(\alpha+B\right)^{-1} \geq 1$. 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 [32, 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 [32, 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 [32, Theorem 5.1] cannot be applied. Instead, we extend an argument due to Regev (see, e.g, [33]) to prove the search-to-decision reduction, which requires that Bq is an integer. We first state an easy lemma.

   Lemma 4. *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 $\sum_{keven}\phi(k) = \sum_{kodd}\phi(k) = \frac{1}{2}$.*

   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-trinary secret distribution.

Lemma 5. *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-trinary 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 3. *Second*, a reduction from $\mathtt{dLWE}_{n,m',q,\phi}$ to $\mathtt{dLWR}_{n,m,q,p}$, due to [32, 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 3 indeed is applicable.

□

□

Note that the conditions imposed by Lemma 3 imply that $1/\alpha$ must at least grow linearly in n. This is a common bottleneck in known LWE to LWR reductions [15], [31], [32].

VI. 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 [6, 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*. 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 [6]), enhanced with Grover's quantum search algorithm [46], [47] to fit a post-quantum scenario. We optimize Round5 parameters such that the best known attacks result in at least a minimum targeted cost, following which we choose parameters that result in minimum bandwidth requirements.

A. Weighted Primal and Dual lattice reduction-based attacks

It is possible to express a Learning with Rounding problem instance $\boldsymbol{B} = \left\langle \left\lfloor \frac{p}{q} \cdot \langle \boldsymbol{AS} \rangle_q \right\rceil \right\rangle_p$ as a Learning with Errors problem instance with bounded noise, as follows:

$$\frac{q}{p}\boldsymbol{B} \equiv \boldsymbol{AS} + \boldsymbol{E'} \pmod q \text{ with } \boldsymbol{E'} \in \left(-\frac{q}{2p}, \frac{q}{2p}\right] \cap \mathbb{Z} \quad (15)$$

We then consider the primal [6] and dual [48] attacks' effectiveness against this problem. We optimize parameters with respect to the primal attack's success criteria from [6, Sec. 6.3], which we adopt to also account for attack speedups exploiting the fact that Round5 secret-keys have smaller variance than the noise. Lattice rescaling allows the attacker to take advantage of this, by considering the scaled version $\Lambda_\omega = \{(\omega\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \in (\omega\mathbb{Z})^d \times \mathbb{Z}^m \times \mathbb{Z} : \left(\boldsymbol{A}_m | \boldsymbol{I}_m | -\boldsymbol{b}\right)\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{z}\end{pmatrix} = \boldsymbol{0} \pmod q\}$ of the primal lattice (A similar rescaling applies for the dual lattice in the dual attack). Adopting the attack success criterion to account for this involves updating the norm of the projection of the short vector (that the attack searches for) in the success criterion [6, Eq. 1], following which Round5 parameters can be chosen such that the resulting attack fails.

B. Hybrid lattice-reduction and Meet-in-the-Middle attack

The combinatorial hybrid attack originally due to [49] applies to schemes where the entropy of the secret-key distribution is not very high, especially in cases where the secret is sparse and trinary, as in Round5, and the schemes in [21], [26]. Although the recent work of Wünderer [50] indicate that the hybrid attack *may* not be as competitive as previously thought, we choose to remain conservative and account for it in our parameter selection (especially in choosing the Hamming weight of secret-keys), where it typically (assuming a conservative attack cost analysis) turns out to be the most effective attack.

We note that our current analysis of the attack could be further extended following Wünderer's work [50], which involves a more accurate analysis of the attack. 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.

C. Attacks targeting Sparse Secrets

Finally, we consider an attack due to Albrecht *et al.* [48] exploiting the fact that a significant number of Round5's secret-key coefficients are zero. This leads to the realization that the corresponding columns of the GLWR public matrix $\boldsymbol{A}$ can be ignored in the resulting lattice-reduction based attack, reducing the dimension of the underlying lattice and thus the attack cost. This technique holds for both the primal and dual attack. We optimize over the Hamming weight of secret-keys to account for this attack, both in its standard and an adaptive embodiment [48, Sec. 5].

VII. Parameter Selection and Performance

The security of Round5 depends, among other parameters, on 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 [26], [54].

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 f_n^3 is always used to generate the public value $\boldsymbol{A}$ in ring setting. In non-ring setting $n = 1$ we have two options, f_n^1 and f_n^2, so the designator takes the form `R5FN`$\{\tau\}$`_`$\{l\}$`KEM` and `R5FN`$\{\tau\}$`_`$\{l\}$`PKE`, where $\tau \in \{1, 2\}$ is the function f_n^τ, $l \in \{1, 3, 5\}$ is the security level, and KEM/PKE has the same meaning as before.

Table I 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 III-C), that is the instantiation of the (generic) error-correction mechanism ECC_Enc_y used in the core Round5 building block CPA-PKE (see Section III-B, Algorithm 2). Our security estimates are conservative, as verified in an independent analysis [55] of various lattice-based proposals to the NIST standardization process.

A. Comparison and Discussion

Table II leads to the following observations.

LWE vs LWR: As expected, LWR leads to lower bandwidth requirements, as observed, e.g., when comparing $uRound.PKE_{n=1}$ with Frodo [42], or Saber [8] with Kyber [53].

Prime cyclotomic rings with q power of two allow for fine-tuning of parameters in Round5. For instance, NewHope [51] 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.

RLWR vs MLWR: Saber [8] offers three configurations corresponding to ranks $\{2, 3, 4\}$ in a module lattice. In contrast, the scalability of Round5 allows finding finely-tuned parameters to fit *any* security target.

Generation of $\boldsymbol{A}$: We observe that f_n^2 allows for a 10x computational speed-up compared with f_n^1 in $uRound5.PKE_{n=1}$ when AES is used as the pseudo-random number generator. Thus, f_n^2 allows us to achieve almost the same performance [29] as Frodo [42] – whose method to obtain $\boldsymbol{A}$ is similar to f_n^2 – when relying on AVX instructions. We note that Frodo [42] reports a key generation of 111424 kilo CPU cycles when

Table I: Round5 parameters sets

		Round5.KEM			Round5.PKE		
	Parameters	**NIST1**	**NIST3**	**NIST5**	**NIST1**	**NIST3**	**NIST5**
$n = 1$	d, n, h	500, 1, 130	885, 1, 176	1128, 1, 564	502, 1, 124	860, 1, 430	1128, 1, 564
	q, p, t	$2^{15}, 2^{11}, 2^{7}$	$2^{15}, 2^{11}, 2^{9}$	$2^{14}, 2^{12}, 2^{8}$	$2^{13}, 2^{11}, 2^{9}$	$2^{14}, 2^{12}, 2^{8}$	$2^{14}, 2^{12}, 2^{8}$
	$B, \bar{n}, \bar{m}, f$	4, 6, 6, 1	4, 6, 8, 1	4, 8, 8, 1	4, 5, 7, 1	4, 6, 8,1	4, 8, 8, 1
	μ	32	48	64	32	48	64
	Public key	4142 B	7327 B	13569 B	3469 B	7765 B	13569 B
	Ciphertext	4153 B	9789 B	13600 B	4912 B	10420 B	13660 B
	PQ security	2^{84}	2^{159}	2^{218}	2^{84}	2^{159}	2^{218}
	Failure rate	2^{-113}	2^{-98}	2^{-121}	2^{-159}	2^{-161}	2^{-121}
	Version (f_n^1)	`R5FN1_1KEM`	`R5FN1_3KEM`	`R5FN1_5KEM`	`R5FN1_1PKE`	`R5FN1_3PKE`	`R5FN1_5PKE`
	Version (f_n^2)	`R5FN2_1KEM`	`R5FN2_3KEM`	`R5FN2_5KEM`	`R5FN2_1PKE`	`R5FN2_3PKE`	`R5FN2_5PKE`
$n = d$	d, n, h	420, 420, 66	700, 700, 196	946, 946, 208	418, 418, 100	756, 756, 136	1018, 1018, 222
	q, p, t	$2^{15}, 2^{7}, 2^{5}$	$2^{15}, 2^{8}, 2^{4}$	$2^{15}, 2^{8}, 2^{4}$	$2^{14}, 2^{8}, 2^{4}$	$2^{15}, 2^{8}, 2^{5}$	$2^{14}, 2^{9}, 2^{3}$
	$B, \bar{n}, \bar{m}, f$	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	$128 + 91$	$192 + 103$	$256 + 121$	$128 + 91$	$192 + 103$	$256 + 121$
	Public key	384 B	724 B	978 B	434 B	780 B	1178 B
	Ciphertext	505 B	848 B	1135 B	544 B + DEM	965 B + DEM	1320 B + DEM
	PQ security	2^{84}	2^{160}	2^{218}	2^{84}	2^{159}	2^{218}
	Failure rate	2^{-66}	2^{-70}	2^{-64}	2^{-141}	2^{-131}	2^{-153}
	Version (f_n^3)	`R5ND_1KEM`	`R5ND_3KEM`	`R5ND_5KEM`	`R5ND_1PKE`	`R5ND_3PKE`	`R5ND_5PKE`

no AVX operations are used. This is a factor of 40 lower than when using AVX operations. This configuration is not reported in [29].

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 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.

VIII. 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 the 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.

We have shown that the ring variant is faster that most comparable RLWE schemes. In the general lattice case the function f_n^2 allows for quick generation of the public parameter $\boldsymbol{A}$, while both stopping precomputation and backdoor-like attacks, and ensuring provable security.

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] National Institute of Standards and Technology, "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.

Table II: Performance of Round5 C implementation on Intel Xeon Platinum 8168. For each scheme, columns from left to right represent respectively, IND security claim (IND-CPA or IND-CCA), the underlying hardness assumption, claimed quantum security level, failure probability (FP) during decryption, sizes of public-key (PK) and ciphertext (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 Bytes		Kilo CPU Cycles		
Scheme	**IND**	**Prob.**	**Sec.**	**FP**	**PK**	**CT**	**KG**	**Enc**	**Dec**
R5ND_1KEM	CPA	RLWR	2^{84}	2^{-66}	384	505	19.7	31.1	17.0
R5ND_3KEM	CPA	RLWR	2^{160}	2^{-70}	724	848	40.9	61.0	31.8
R5ND_5KEM	CPA	RLWR	2^{218}	2^{-64}	978	1135	54.2	69.7	39.9
NewHope512-CPA-KEM [51]	CPA	RLWE	2^{101}	2^{-213}	928	1088	96.9	147.8	54.0
NewHope1024-CPA-KEM [51]	CPA	RLWE	2^{233}	2^{-216}	1824	2176	222.1	323.3	71.5
R5ND_1PKE	CCA	RLWR	2^{84}	2^{-141}	434	466	20.1	35.5	46.9
R5ND_3PKE	CCA	RLWR	2^{159}	2^{-131}	780	965	36.2	63.7	98.7
R5ND_5PKE	CCA	RLWR	2^{218}	2^{-153}	1178	1320	55.5	97.0	131.2
LAC128 [52]	CCA	RLWE	2^{133}	2^{-240}	544	1024	86.7	198.0	284.7
NewHope1024-CCA-KEM [51]	CCA	RLWE	2^{233}	2^{-216}	928	1120	284.5	359.2	414.0
Saber [8]	CCA	MLWR	2^{180}	2^{-136}	992	1088	156.4	238.9	284.9
CRYSTALS-Kyber768 [53]	CCA	MLWE	2^{161}	2^{-142}	1088	1152	232.0	318.0	391.1
Frodo976 [42]	CCA	LWE	2^{150}	2^{-200}	15632	15768	99184	96976	99090

[2] European Telecommunications Standards Institute, ""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.

[3] "Terms of Reference for ETSI TC Cyber Working Group for Quantum-Safe Cryptography (ETSI TC Cyber WG-QSC)," https://portal.etsi.org/TBSiteMap/CYBER/CYBERQSCToR.aspx, accessed: 15-02-2017.

[4] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila, "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*, 2015, pp. 553–570. [Online]. Available: https://doi.org/10.1109/SP.2015.40

[5] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila, "Frodo: Take off the ring! Practical, Quantum-Secure Key Exchange from LWE," Cryptology ePrint Archive, Report 2016/659, 2016, http://eprint.iacr.org/2016/659.

[6] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange - a new hope," Cryptology ePrint Archive, Report 2015/1092, 2015, http://eprint.iacr.org/2015/1092.

[7] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, and D. Stehlé, "CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM," Cryptology ePrint Archive, Report 2017/634, 2017, http://eprint.iacr.org/2017/634.

[8] J.-P. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren, "SABER," National Institute of Standards and Technology, Tech. Rep., 2017, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[9] J. H. Cheon, K. H. Han, J. Kim, C. Lee, and Y. Son, "A Practical Post-Quantum Public-Key Cryptosystem Based on spLWE," Cryptology ePrint Archive, Report 2016/1055, 2016, http://eprint.iacr.org/2016/1055.

[10] J. H. Cheon, D. Kim, J. Lee, and Y. Song, "Lizard: Cut off the Tail! Practical Post-Quantum Public-Key Encryption from LWE and LWR," Cryptology ePrint Archive, Report 2016/1126, 2016, http://eprint.iacr.org/2016/1126.

[11] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, "Lattice Signatures and Bimodal Gaussians," Cryptology ePrint Archive, Report 2013/383, 2013, http://eprint.iacr.org/2013/383.

[12] L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehle, "CRYSTALS – Dilithium: Digital Signatures from Module Lattices," Cryptology ePrint Archive, Report 2017/633, 2017, https://eprint.iacr.org/2017/633.

[13] V. Lyubashevsky, C. Peikert, and O. Regev, "On Ideal Lattices and Learning with Errors Over Rings," Cryptology ePrint Archive, Report 2012/230, 2012, http://eprint.iacr.org/2012/230.

[14] A. Langlois and D. Stehle, "Worst-case to average-case reductions for module lattices," Cryptology ePrint Archive, Report 2012/090, 2012, https://eprint.iacr.org/2012/090.

[15] A. Banerjee, C. Peikert, and A. Rosen, "Pseudorandom functions and lattices," Cryptology ePrint Archive, Report 2011/401, 2011, http://eprint.iacr.org/2011/401.

[16] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Post-quantum cryptography*. Springer, 2009, pp. 147–191.

[17] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *Advances in Cryptology — CRYPTO '91: Proceedings*, J. Feigenbaum, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 433–444. [Online]. Available: http://dx.doi.org/10.1007/3-540-46766-1_35

[18] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," Internet Requests for Comments, RFC Editor, RFC 4301, December 2005, http://www.rfc-editor.org/rfc/rfc4301.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4301.txt

[19] E. Rosen and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)," Internet Requests for Comments, RFC Editor, RFC 4364, February 2006.

[20] N. P. Smart, M. R. Albrecht, Y. Lindell, E. Orsini, V. Osheter, K. Paterson, and G. Peer, "LIMA," National Institute of Standards and Technology, Tech. Rep., 2017, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[21] J. Hoffstein, J. Pipher, J. M. Schanck, J. H. Silverman, W. Whyte, and Z. Zhang, "Choosing Parameters for NTRUEncrypt," pp. 3–18, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-52153-4_1

[22] A. Hülsing, J. Rijneveld, J. M. Schanck, and P. Schwabe, "High-speed key encapsulation from NTRU," in *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, 2017, pp. 232–252. [Online]. Available: https://doi.org/10.1007/978-3-319-66787-4_12

[23] O. Regev, "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*, 2010, pp. 191–204. [Online]. Available: https://doi.org/10.1109/CCC.2010.26

[24] B. Applebaum, D. Cash, C. Peikert, and A. Sahai, *Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 595–618. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03356-8_35

[25] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, "Classical Hardness of Learning with Errors," in *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, ser. STOC '13. New York, NY, USA: ACM, 2013, pp. 575–584. [Online]. Available: http://doi.acm.org/10.1145/2488608.2488680

[26] D. J. Bernstein, C. Chuengsatiansup, T. Lange, and C. van Vredendaal, "NTRU Prime: reducing attack surface at low cost," Cryptology ePrint Archive, Report 2016/461, 2016, https://eprint.iacr.org/2016/461.

[27] T. Fritzmann, T. Pöppelmann, and J. Sepulveda, "Analysis of Error-Correcting Codes for Lattice-Based Key Exchange," Cryptology ePrint-Archive, Report 2018/150, 2018, https://eprint.iacr.org/2018/150.

[28] V. Singh, "A practical key exchange for the internet using lattice cryptography," Cryptology ePrint Archive, Report 2015/138, 2015.

[29] M. Hamburg, "Graphs of "Estimate all the LWE, NTRU schemes!" indexed to the "PQC Lounge" data." https://bitwiseshiftleft.github.io/estimate-all-the-lwe-ntru-schemes.github.io/graphs.

[30] J. Alwen, S. Krenn, K. Pietrzak, and D. Wichs, "Learning with rounding, revisited: New reduction, properties and applications," Cryptology ePrint Archive, Report 2013/098, 2013, http://eprint.iacr.org/2013/098.

[31] A. Bogdanov, S. Guo, D. Masny, S. Richelson, and A. Rosen, "On the Hardness of Learning with Rounding over Small Modulus," Cryptology ePrint Archive, Report 2015/769, 2015, http://eprint.iacr.org/2015/769.

[32] S. Bai, A. Langlois, T. Lepoint, A. Sakzad, D. Stehle, and R. Steinfeld, "Improved security proofs in lattice-based cryptography: using the Rényi divergence rather than the statistical distance," Cryptology ePrint Archive, Report 2015/483, 2015, http://eprint.iacr.org/2015/483.

[33] O. Regev, "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography," in *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, ser. STOC '05. New York, NY, USA: ACM, 2005, pp. 84–93. [Online]. Available: http://doi.acm.org/10.1145/1060590.1060603

[34] D. Hoffheinz, K. Hövelmanns, and E. Kiltz, "A Modular Analysis of the Fujisaki-Okamoto Transformation," Cryptology ePrint Archive, Report 2017/604, 2017, http://eprint.iacr.org/2017/604.

[35] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, ser. CCS '93. New York, NY, USA: ACM, 1993, pp. 62–73. [Online]. Available: http://doi.acm.org/10.1145/168588.168596

[36] D. Boneh, O. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," Cryptology ePrint Archive, Report 2010/428, 2010, http://eprint.iacr.org/2010/428.

[37] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, *Relations among notions of security for public-key encryption schemes*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 26–45. [Online]. Available: https://doi.org/10.1007/BFb0055718

[38] M. O. Saarinen, "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*, ser. IoTPTS '17. ACM, April 2017, pp. 15–22. [Online]. Available: https://eprint.iacr.org/2016/1058

[39] ——, "HILA5: On reliability, reconciliation, and error correction for Ring-LWE encryption," in *SAC 2017*, ser. Lecture Notes in Computer Science, C. Adams and J. Camenisch, Eds., vol. 10719. Springer, 2018, pp. 192–212.

[40] FIPS, "SHA-3 standard: Permutation-based hash and extendable-output functions," Federal Information Processing Standards Publication 202, August 2015.

[41] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," Cryptology ePrint Archive, Report 2001/108, 2001, http://eprint.iacr.org/2001/108.

[42] M. Naehrig, E. Alkim, J. Bos, L. Ducas, K. Easterbrook, B. LaMacchia, P. Longa, I. Mironov, V. Nikolaenko, C. Peikert, A. Raghunathan, and D. Stebila, "FrodoKEM," National Institute of Standards and Technology, Tech. Rep., 2017, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[43] U. Maurer, R. Renner, and C. Holenstein, "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.

[44] J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya, "Merkle-Damgård Revisited: How to Construct a Hash Function," in *Advances in Cryptology – CRYPTO 2005*, V. Shoup, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 430–448.

[45] J.-P. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren, "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.

[46] T. Laarhoven, "Search problems in cryptography," Ph.D. dissertation, Eindhoven University of Technology, 2015.

[47] T. Laarhoven, M. Mosca, and J. van de Pol, "Finding shortest lattice vectors faster using quantum search," Cryptology ePrint Archive, Report 2014/907, 2014, http://eprint.iacr.org/2014/907.

[48] M. R. Albrecht, "On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL," Cryptology ePrint Archive, Report 2017/047, 2017, http://eprint.iacr.org/2017/047.

[49] N. Howgrave-Graham, "A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU," in *Advances in Cryptology - CRYPTO 2007: 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007. Proceedings*, A. Menezes, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 150–169. [Online]. Available: https://doi.org/10.1007/978-3-540-74143-5_9

[50] T. Wünderer, "Revisiting the hybrid attack: Improved analysis and refined security estimates," Cryptology ePrint Archive, Report 2016/733, 2016, http://eprint.iacr.org/2016/733.

[51] T. Pöppelmann, E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. de la Piedra, P. Schwabe, and D. Stebila, "NewHope," National Institute of Standards and Technology, Tech. Rep., 2017, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[52] X. Lu, Y. Liu, D. Jia, H. Xue, J. He, and Z. Zhang, "LAC," National Institute of Standards and Technology, Tech. Rep., 2017, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[53] P. Schwabe, R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, G. Seiler, and D. Stehle, "CRYSTALS-KYBER," National Institute of Standards and Technology, Tech. Rep., 2017, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[54] D. J. Bernstein, "A subfield-logarithm attack against ideal lattices," February 2014, available from https://blog.cr.yp.to/20140213-ideal.html.

[55] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, and T. Wünderer, "Estimate all the LWE, NTRU schemes!" Cryptology ePrint Archive, Report 2018/331, 2018, https://eprint.iacr.org/2018/331.