

Radical Ring-LWR: Efficient Key Encapsulation and Signatures from Structured Rounding

Joost Renes¹ , Joppe W. Bos² , Haochen Huang³ , Selim Kirbiyik⁴ ,
Alberto Ovena⁴ , Sujoy Sinha Roy⁴ , Frederik Vercauteren⁵ ,
Peng Wang³ , Fangyu Zheng³  and Chenxin Zhong³ 

¹ NXP Semiconductors, Austin, United States

² NXP Semiconductors, Leuven, Belgium

³ University of Chinese Academy of Sciences, School of Cryptology, Beijing, China

⁴ Graz University of Technology, Graz, Austria

⁵ KU Leuven, Leuven, Belgium

Abstract. State-of-the-art lattice-based cryptography requires a power-of-two cyclotomic field that limits the attainable security levels, or a module structure for which the cost grows quadratically in the module rank. Radical rings were recently proposed as a solution in the context of Learning With Errors (LWE) based Key Encapsulation Mechanisms (KEMs) with heuristic hardness arguments for the Ring-LWE security and failure probability. We develop the Learning With Rounding counterpart, *Radical Ring-LWR* (RR-LWR). We give a *proved* closed-form bound on the distortion incurred from the error sampling independent of the radical ring parameters: this places RR-LWR inside the regime identified by Peikert as safe for instantiating Ring-LWE/LWR. We instantiate two schemes: Mithril, an IND-CCA KEM, and Octarine, an EF-CMA signature scheme. They are built on a single radical-ring arithmetic foundation based on powers of two (favorable for sampling, rounding and masking) and we provide security reductions in the QROM to RR-LWR and SelfTarget-RR-SIS, a radical-ring variant of SIS. We show that the KEM failure probability estimators used for power-of-two cyclotomics are insufficient and provide an exact solution tailored to the RR-LWR setting. Finally, we instantiate Mithril and Octarine with concrete parameters and benchmark optimized AVX2 and Arm Cortex-M4 implementations, showing that both are competitive with (and in some cases significantly faster than) the ML-KEM and ML-DSA standards.

Keywords: Post-quantum cryptography · Radical rings · Learning with rounding · Key encapsulation · Digital signatures.

1 Introduction

The anticipated arrival of a cryptographically relevant quantum computer has driven a global migration to *post-quantum cryptography* (PQC). The U.S. National Institute of Standards and Technology (NIST) standardized the module-lattice key encapsulation mechanism (KEM) ML-KEM [Nat24b] (previously known as CRYSTALS-Kyber [BDK⁺18]) and the module-lattice signature ML-DSA [Nat24a] (previously known as CRYSTALS-Dilithium [DKL⁺18]), with the process continuing through the additional-signatures “on-ramp” [Nat22]. The effort is now global: ISO/IEC is evaluating further KEMs [ISO26], South Korea concluded its KpqC competition [KN25], and China has launched a next-generation commercial-cryptography program soliciting PQC proposals [Ins25]. Across all

E-mail: joost.renes@nxp.com (Joost Renes), joppe.bos@nxp.com (Joppe W. Bos)



of these efforts a single tension recurs: the algebraic structure that makes lattice schemes fast also dictates which security levels are reachable.

Fine-grained parameter selection. For Ring-LWE [LPR10] and Ring-LWR [BPR12] the ring of choice is the power-of-two cyclotomic $\mathbf{Z}_q[x]/(x^n + 1)$, which enables the Number-Theoretic Transform (NTT) but ties the security level to a degree n that must be a power of two. Module-LWE/LWR [Reg05, LPR10, BGV12, LS15, BPR12] recover flexibility by placing a rank- k module over a smaller ring, tuning the security through the rank k . This flexibility comes at a cost: the public matrix holds k^2 ring elements, so its expansion via a XOF such as SHAKE (often the performance bottleneck) scales *quadratically*. Halving n and doubling k thus doubles the matrix size while barely moving the security level, and a smaller degree n forces multiple message bits per coefficient, raising the decryption-failure probability. The Kyber team explicitly notes this relationship: small n forces multiple key bits per coefficient (lowering security), while large n removes the ability to scale security via k [SAB⁺20].

Radical rings. An approach using radical rings [BRVW26] offers a middle ground carrying *more* structure than a module while ensuring a large power-of-two cyclotomic subring for NTT-based arithmetic. This has the advantage that the public matrix only needs k rather than k^2 uniform ring elements. One replaces $x^n + 1$ by the irreducible polynomial $f(x) = (x^k - 2)^n + 1$ (n a power of two, $k \geq 1$), which defines a degree- k radical extension of the $2n$ -th cyclotomic field. The enabling identity is the ring isomorphism $\mathbf{S}_{q,n,k} := \mathbf{Z}_q[x]/((x^k - 2)^n + 1) \cong (\mathbf{Z}_q[y]/(y^n + 1))[x]/(x^k - y - 2)$, under which k acts as a module rank for scaling security, while the product $\mathbf{A} \cdot \mathbf{s}$ is constructed from only k ring elements with all arithmetic in the power-of-two subring $\mathbf{Z}_q[y]/(y^n + 1)$.

The example in Figure 1 compares the matrix structures from LWR, Module-LWR, Radical Ring-LWR and power-of-two cyclotomic Ring-LWR for $n = 2$ and $k = 2$, requiring k^2n^2 , k^2n , kn and kn randomly generated elements respectively for the matrix representation.

Radical-ring-based cryptography. Bos, Renes, Vercauteren and Wang [BRVW26] demonstrated the flexibility of radical rings by presenting a Ring-LWE based KEM. They provided a heuristic argument for Radical Ring-LWE with respect to requirements for secure instances from Peikert [Pei16] and (optimistically) assumed that failure probability would not be affected. In this setting, their implementations outperformed state-of-the-art ML-KEM implementations for an AVX2-based architecture. Instead, we prove a closed-form distortion bound (Proposition 1) and give an exact failure analysis for the correlated rounding errors that structured rounding introduces.

1.1 Mithril and Octarine

The goal of this work is to develop Radical Ring Learning With *Rounding* from its mathematical foundations to a concrete KEM scheme (Mithril) and a signature scheme (Octarine). Our objective is not merely to obtain high performance on a particular platform. Rather, we use the additional structure of radical rings to combine fine-grained security scaling with conservative security margins, simple power-of-two arithmetic, predictable signing, and flexibility in the choice of implementation technique. As a result, we obtain cryptographic designs that are competitive with state-of-the-art cryptosystems (or faster in various cases) on modern architectures while increasing security margins for the 128-bit KEM and improving implementation aspects, specifically for high-assurance implementations where side-channel and fault attacks have to be mitigated. Our contributions are as follows:

- We instantiate two RR-LWR schemes: Mithril (KEM via FO [HHK17]) and Octarine (digital signature scheme via Fiat–Shamir with Aborts [Lyu09, Lyu12]). Both share

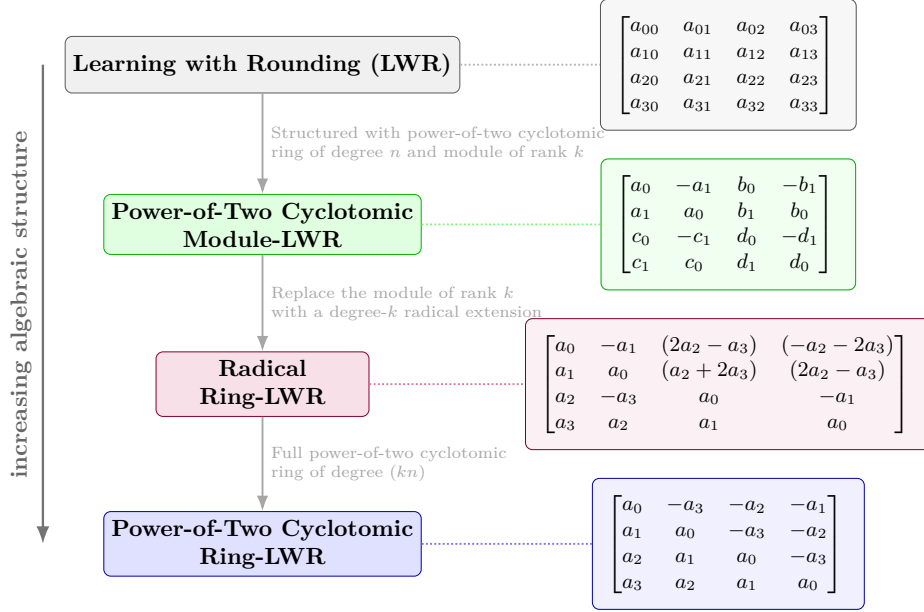


Figure 1: Examples of $(kn) \times (kn)$ matrices for $n, k = 2$, showing the increasing matrix structure in the multiplication $\mathbf{A}\mathbf{s}$ for generic LWR, Module-LWR over $\mathbf{Z}_q[x]/(x^2 + 1)$ with module rank 2, Radical Ring-LWR for ring $\mathbf{Z}_q[y]/(y^2 + 1)[x]/(x^2 - y - 2)$, and Ring-LWR for $\mathbf{Z}_q[x]/(x^4 + 1)$ with negacyclic-circulant multiplication matrix, respectively.

the radical-ring representation, uniform samplers, packing principles, and polynomial-multiplication techniques, which creates a common framework for the KEM and signature scheme.

- We provide formal security proofs for both Mithril and Octarine. First, we *prove* in Proposition 1 the exact distortion that our sampling domain and monomial basis $x^i y^j$ induce on the error in the canonical embedding, and we tightly bound it to be less than 3 for all n and k . This replaces the experimental bound of [BRVW26, Section 4.1] and *proves* that RR-LWR is a *well-instantiated* Ring-LWR problem in the sense of Peikert [Pei16].
- We provide proofs for IND-CCA security of Mithril (Proposition 2) and EF-CMA security of Octarine (Proposition 3) in the QROM, reducing to RR-LWR and SelfTarget-RR-SIS, a radical-ring variant of SIS. We provide a thorough analysis of the impact of the radical ring on failure probabilities and demonstrate that this should carefully be accounted for. For Octarine we show that the min-entropy reduces significantly compared to ML-DSA due to the ring structure with power-of-two modulus, see Table 3, leading to a *tight* security reduction.
- We instantiate Mithril (Table 1) and Octarine (Table 2) with parameters targeting 128-bit, 256-bit and 512-bit security levels. We use the flexibility of the radical ring parameter k to increase security margins while limiting the impact. For example, Mithril-128 increases the classical Core-SVP complexity to 129 bits for dual-hybrid attacks compared to 115 bits for ML-KEM-512 (using the lattice-estimator framework of Albrecht, Player and Scott [APS15]). For all resulting parameter sets Mithril/Octarine-128, 256, 512 we provide core-SVP estimates, correctness or abort probabilities, and concrete key, ciphertext, and signature sizes.
- We provide AVX2 and Cortex-M4 implementations that match or outperform ML-

KEM/ML-DSA in various cases. In particular, on the Arm Cortex-M4 core Octarine is faster than ML-DSA across all other parameter sets with a notable 31% reduction in the signature verification time compared to ML-DSA-87. For the 256-bit security level both Mithril and Octarine achieve similar or better performance than ML-KEM-1024 and ML-DSA-87 respectively for both AVX2 and Cortex-M4 implementation, with the only exception being decapsulation for AVX2 where Mithril-256 is 8% slower.

Related Work. RR-LWR builds on the structured-module viewpoint of [BRVW26]. Algebraically structured LWE/LWR has been studied extensively, including Polynomial-LWE [SSTX09, RSW18], order-LWE [BBPS19], and middle-product LWE [RSSS17], unified by Peikert and Pepin [PP24]. Our hardness argument follows the canonical-embedding methodology of [LPR10]. Since $f(x) = (x^k - 2)^n + 1$ is irreducible and defines a degree- kn number field, RR-LWR is *not* the multivariate m -RLWE [PUTPPG15, PUTPPG16, PUTPPG17, PUTPG⁺21] which was broken for certain parameters in [BCV20] and those attacks do not apply. Constructively, Mithril follows the Saber/ML-KEM [DKRV18, Nat24b] lineage and Octarine the ML-DSA lineage [Nat24a, DKL⁺18].

2 Background

We fix the notation and recall the background on Learning With Rounding and structured rings needed throughout the paper. We also recall the rounding and decomposition operations and the Fiat–Shamir with Aborts paradigm used by our constructions.

Notation and rings. For a positive integer q , let $\mathbf{Z}_q = \mathbf{Z}/q\mathbf{Z}$. Unless stated otherwise, elements of $\mathbf{Z}_q$ are represented by their centered representatives in $[-q/2, q/2)$ when q is even. We extend this convention coefficient-wise to vectors, polynomials, and ring elements. Throughout, n denotes a power of two and $\mathbf{R}_{q,n} := \mathbf{Z}_q[y]/(y^n + 1)$ denotes the degree- n power-of-two cyclotomic ring. More generally, for a monic polynomial $f \in \mathbf{Z}[x]$, we write $\mathbf{R}_q := \mathbf{Z}_q[x]/(f(x))$. Regular lower-case letters denote integers or elements of $\mathbf{Z}_q$, bold lower-case letters denote vectors or ring elements, and bold upper-case letters denote matrices. Throughout we write $\mathbf{A}$ for the public object of an LWR instance, whether it is a vector, a ring element, or the matrix it induces (see Figure 1). For $\mathbf{A} = \sum_{i=0}^{d-1} \sum_{j=0}^{n-1} a_{i,j} y^j x^i$ with centered coefficients $a_{i,j}$, we write $\|\mathbf{A}\|_\infty := \max_{i,j} |a_{i,j}|$. The notation $\mathcal{U}_x$ denotes the uniform distribution over $\{-x, \dots, x-1\}$. We use $D_{\mathbf{A}}(R)$ for a distribution used to sample $\mathbf{A}$ from a ring or module R , and $D_{\mathbf{A}}(R; \rho)$ when this sampling is generated deterministically from a seed ρ . For moduli $q > p \geq 2$, we define coefficient-wise rounding from $\mathbf{Z}_q$ to $\mathbf{Z}_p$ by $\lfloor a \rfloor_p := \lfloor \frac{p}{q} a \rfloor \bmod p$, where ties are resolved according to a fixed convention. We use the same notation for the coefficient-wise extension of this map to vectors and ring elements.

Learning with Rounding. Learning With Rounding (LWR) [BPR12] replaces the explicit error of Learning With Errors (LWE) [Reg05] by the deterministic error introduced when mapping an inner product from a modulus q to a smaller modulus p . For $q > p$, a secret $\mathbf{s} \in \mathbf{Z}_q^d$, and uniformly random $\mathbf{A} \in \mathbf{Z}_q^d$, an LWR sample is $(\mathbf{A}, \lfloor \frac{p}{q} \langle \mathbf{A}, \mathbf{s} \rangle \rfloor) \in \mathbf{Z}_q^d \times \mathbf{Z}_p$. Module-LWR is defined analogously over $\mathbf{R}_q^d$: for $\mathbf{A}, \mathbf{s} \in \mathbf{R}_q^d$, the rounded value is $\lfloor \frac{p}{q} \langle \mathbf{A}, \mathbf{s} \rangle \rfloor$. Thus Ring-LWR corresponds to module rank one, whereas larger module ranks provide an additional parameter with which to scale the underlying lattice dimension. The hardness of LWR is related to that of LWE through a sequence of reductions [BPR12, AKPW13, BGM⁺16]. Of particular relevance to our setting are results allowing power-of-two moduli and a bounded number of samples [BGM⁺16]. For Ring-LWR, however, the concrete choice of number ring and the basis in which rounding is performed require additional care. In Section 4.1, we address this issue for the radical rings used in this work.

Fiat–Shamir with Aborts. Our signature construction follows the Fiat–Shamir with Aborts paradigm [Lyu09, Lyu12]. We briefly recall the abstraction needed for the security analysis. Consider a three-move identification scheme in which the prover first generates a commitment $\mathbf{w}$, receives a challenge c , and produces a response $\mathbf{z}$. Verification checks a relation between $(pk, \mathbf{w}, c, \mathbf{z})$. The Fiat–Shamir transform makes the protocol non-interactive by deriving the challenge from a random oracle, typically as $c = H(\mu, \mathbf{w})$, where μ binds the transcript to the message and public key. For lattice-based protocols, the response commonly takes the form $\mathbf{z} = \mathbf{y} + c\mathbf{s}$, where $\mathbf{y}$ is freshly sampled masking randomness and $\mathbf{s}$ contains secret-key material. Directly releasing $\mathbf{z}$ may leak information about $\mathbf{s}$ because its distribution depends on the secret-dependent shift $c\mathbf{s}$. Fiat–Shamir with Aborts therefore applies rejection sampling: the prover outputs a transcript only when $\mathbf{z}$, and possibly further intermediate quantities, satisfy prescribed bounds. Parameters are selected so that the distribution of accepted responses is independent, or statistically close to independent, of the secret. For our security analysis we use the framework of [BBD⁺23], which captures the required honest-verifier zero-knowledge and commitment-entropy properties and provides transformations from the corresponding identification scheme to Fiat–Shamir-with-Aborts signatures in both the ROM and QROM. We instantiate this framework for Octarine in its security analysis.

Radical Ring-LWR. Radical rings were introduced by Bos et al. [BRVW26]. Instead of a power-of-two cyclotomic $f(x) = x^n + 1$, one uses $f(x) = (x^k - 2)^n + 1$ for a power of two n and positive integer k , which is irreducible [BRVW26, Lemma 1] and defines a number field $\mathbf{Q}[x]/(f(x))$. The resulting Ring-LWR instance is called *radical* Ring-LWR (RR-LWR). The isomorphism

$$\mathbf{S}_{q,n,k} := \mathbf{Z}_q[x]/((x^k - 2)^n + 1) \cong (\mathbf{Z}_q[y]/(y^n + 1)) [x]/(x^k - y - 2), \quad (1)$$

$x^k \mapsto y + 2$, exposes a large power-of-two cyclotomic subring $\mathbf{Z}_q[y]/(y^n + 1)$ in which all arithmetic (including NTTs) can be performed. The degree- k radical extension plays the role of a module rank for scaling the lattice dimension without growing n . Compared to a generic rank- k module it reduces cost and randomness, as the public element $\mathbf{A}$ consists of only k rather than k^2 base-ring elements. Here, $\mathbf{A} \cdot \mathbf{s}$ is a matrix–vector product over $\mathbf{Z}_q[y]/(y^n + 1)$ whose wrap-around terms are multiplied by $y + 2$ (see Figure 1 for an example matrix).

3 Mithril and Octarine

We instantiate the RR-LWR framework in two complementary post-quantum constructions: the key encapsulation mechanism Mithril and the digital signature scheme Octarine. They share the same underlying hardness assumption, ring representation, sampling procedures, and polynomial arithmetic approach. Their cryptographic designs are otherwise distinct: Mithril follows the standard LWR-based encryption paradigm and is converted into an IND-CCA-secure KEM using a Fujisaki–Okamoto transformation, whereas Octarine follows the Fiat–Shamir with Aborts paradigm for lattice-based signatures.

3.1 A Unified Design Framework

Mithril and Octarine share a common design framework built around RR-LWR and arithmetic in radical rings. This section describes the choices shared by both schemes: the choice of ring and moduli, the instantiation of random oracles, and the sampling and representation and arithmetic of ring elements. These choices provide a common implementation foundation while allowing the KEM and signature scheme to select different dimensions and parameters according to their respective correctness, security, and performance requirements.

Ring selection. Both schemes operate over the radical ring $\mathbf{S}_{q,n,k}$ introduced in Section 2, where we additionally take the modulus q to be a power of two. We instantiate **RR-LWR** with coefficient-wise rounding in the monomial basis $x^i y^j$, with secrets sampled directly from $\mathbf{S}_{q,n,k}$ in the monomial basis rather than from the dual. Under the isomorphism from Equation (1) this is a Ring-LWR instance over the number field $\mathbf{Q}[x]/(f(x))$, whose security we analyse in Proposition 1. Since all arithmetic can be carried out in the power-of-two cyclotomic subring, standard implementation techniques such as NTTs or other polynomial-multiplication routines are available if desired.

Choice of LWR and power-of-two moduli. Radical-ring schemes can be instantiated with either LWE or LWR. While the efficiency of arithmetic operations can benefit from NTT-friendly primes that are typically associated with LWE, the advantage of LWR with a power-of-two modulus is that sampling, rounding, packing, decomposition, and bound checks can be expressed using shifts, masks, and carry checks. This choice is also favorable for arithmetic-to-Boolean and Boolean-to-arithmetic conversions that underpin high-assurance masked implementations [CJRR99, PR13, CGV14, BBE⁺18]. Since NTT-friendly primes can still be used for power-of-two moduli (by roughly doubling the bit length, see [CHK⁺21]) and as lattice-based cryptography is generally already very fast compared to contemporary cryptographic schemes such as RSA or ECC, we believe this trade-off is favorable. Hence Mithril and Octarine are based on **RR-LWR**.

Random oracles. Mithril and Octarine require hash functions, or more specifically eXtendable Output Functions (XOFs), for instantiating random oracles, message digest creation, as well as generating pseudo-random bits for key material and ephemeral secrets. Different choices of random oracle can be made. For fair comparison with state-of-the-art cryptosystems that are based on SHA-3 or SHAKE, we select the instances in our implementations that match the scheme we compare with most closely. In the general descriptions of Mithril and Octarine we leave the choice unspecified and refer to H_L as the random oracle domain separated with label L .

Sampling distributions. We sample secret and public values only in uniform distributions of the form $[-2^\alpha, 2^\alpha - 1]$ for non-negative integers α , which can be implemented by uniformly randomly generating exactly $\alpha + 1$ bits. For both Mithril and Octarine this is used to sample the public matrices $\mathbf{A}$ with coefficients in $[-\frac{q}{2}, \frac{q}{2} - 1]$ and secrets $\mathbf{s}_1$ with coefficients in $[-\eta, \eta - 1]$. For Octarine we also sample the coefficients of the ephemeral secret $\mathbf{y}$ in the interval $[-\gamma_1, \gamma_1 - 1]$. All of $p, q, \eta, \gamma_1, \gamma_2$ are chosen to be powers of two. The deterministic error from the rounding operation $\mathbf{s}_2 = \mathbf{A}\mathbf{s}_1 - \frac{q}{p}\mathbf{b}$ is also uniformly randomly distributed in $[-\frac{q}{2p}, \frac{q}{2p} - 1]$, under the assumption that $\mathbf{A}\mathbf{s}_1$ has a uniform distribution. We select our parameters such that $\frac{q}{2p} = \eta$, making it so that the secret and error distributions are equal. In fact, we set $\eta = 2$ for all parameter sets except Octarine-128, where $\eta = 4$, to unify packing and unpacking routines for secrets.

If a polynomial requires many pseudo-randomly sampled bits, one can opt to use multiple random oracle instances (e.g., a XOF with independent input seeds) to allow for parallel implementation. For example, for our Octarine reference implementations we sample $n/4 = 256$ polynomial coefficients independently to allow easy use of 4-times parallel Keccak implementations.

Rounding and dividing. Integers modulo 2^α are represented in the signed interval $[-2^{\alpha-1}, 2^{\alpha-1} - 1]$. For any power of two $m \mid 2^\alpha$ and reduced integer x , we can uniquely write $x = x_0 + x_1 m \bmod 2^\alpha$, where $x_0 = x \bmod m \in [-m/2, m/2 - 1]$ and $x_1 = (x - x_0)/m \bmod 2^\alpha/m \in [-2^{\alpha-1}/m, 2^{\alpha-1}/m - 1]$, defining $\text{LowBits}_q(x, m) := x_0$ and $\text{HighBits}_q(x, m) := x_1$.

Parameter sets. The parameter selection for Mithril and Octarine is very flexible due to the radical ring parameter k . For this work we target 128-bit, 256-bit and 512-bit security

$(pk, sk) \leftarrow \text{Mithril.PKE.KeyGen}()$ 1: $\rho_{\mathbf{A}}, \rho_{\mathbf{s}} \leftarrow \{0, 1\}^{512}$ 2: $\mathbf{A} \leftarrow D_{\mathbf{A}}(\mathbf{S}_{q,n,k}; \rho_{\mathbf{A}})$ 3: $\mathbf{s} \leftarrow D_{\mathbf{s}}(\mathbf{S}_{q,n,k}; \rho_{\mathbf{s}})$ 4: $\mathbf{b} = \lfloor \frac{p}{q} \mathbf{A} \mathbf{s} \rfloor \in \mathbf{S}_{p,n,k}$ 5: return $(pk = (\rho_{\mathbf{A}}, \mathbf{b}), sk = \mathbf{s})$	$(pk, sk) \leftarrow \text{Mithril.KEM.KeyGen}()$ 17: $(pk_{\text{pke}}, sk_{\text{pke}}) \leftarrow \text{PKE.KeyGen}()$ 18: $z \leftarrow \{0, 1\}^{\ell n}$ 19: $hpk \leftarrow H_{hpk}(pk_{\text{pke}}, \lambda/8)$ 20: return $(pk = pk_{\text{pke}},$ 21: $sk = (sk_{\text{pke}}, pk_{\text{pke}}, hpk, z))$
$c \leftarrow \text{Mithril.PKE.Enc}((\rho_{\mathbf{A}}, \mathbf{b}), m)$ 6: $\rho_{\mathbf{s}'} \leftarrow \{0, 1\}^{512}$ 7: $\mathbf{s}' \leftarrow D_{\mathbf{s}}(\mathbf{S}_{q,n,k}; \rho_{\mathbf{s}'})$ 8: $\mathbf{A} \leftarrow D_{\mathbf{A}}(\mathbf{S}_{q,n,k}; \rho_{\mathbf{A}})$ 9: $\mathbf{b}' = \lfloor \frac{p}{q} \cdot (\mathbf{A} \mathbf{s}') \rfloor \in \mathbf{S}_{p,n,k}$ 10: $v' = (\mathbf{b} \mathbf{s}')^{[k-\ell:k-1]} + \frac{q}{2 \cdot p} \in \mathbf{R}_{p,n}^{\ell}$ 11: $c_m = \lfloor \frac{t}{p} ((v' + \frac{p}{2} m) \bmod p) \rfloor \in \mathbf{R}_{t,n}^{\ell}$ 12: return $c = (c_m, \mathbf{b}')$	$(c, ss) \leftarrow \text{Mithril.KEM.Encaps}(pk)$ 22: $m \leftarrow \{0, 1\}^{\ell n}$ 23: $hpk \leftarrow H_{hpk}(pk, \lambda/8)$ 24: $(ss, \rho_{\mathbf{s}'}) \leftarrow H_{ss}(hpk, m, \lambda/8 + 64)$ 25: $c \leftarrow \text{PKE.Enc}(pk, m; \rho_{\mathbf{s}'})$ 26: return (c, ss)
$m' \leftarrow \text{Mithril.PKE.Dec}(\mathbf{s}, (c_m, \mathbf{b}'))$ 13: $v = (\mathbf{b}' \mathbf{s})^{[k-\ell:k-1]} \in \mathbf{R}_{p,n}^{\ell}$ 14: $m' = \lfloor \frac{2}{p} \cdot ((v - \frac{p}{2} \cdot c_m +$ 15: $(\frac{q}{2 \cdot p} - \frac{p}{2 \cdot t})) \bmod p) \rfloor \in \mathbf{R}_{2,n}^{\ell}$ 16: return m'	$ss \leftarrow \text{Mithril.KEM.Decaps}(sk, c)$ 27: $m' \leftarrow \text{PKE.Dec}(sk_{\text{pke}}, c)$ 28: $(ss', \rho_{\mathbf{s}'}) \leftarrow H_{ss}(hpk, m', \lambda/8 + 64)$ 29: $c' \leftarrow \text{PKE.Enc}(pk, m'; \rho_{\mathbf{s}'})$ 30: $\overline{ss} \leftarrow H_{\overline{ss}}(c, z, \lambda/8)$ 31: if $c = c'$ then $ss \leftarrow ss'$ 32: else $ss \leftarrow \overline{ss}$ 33: return ss

Figure 2: The KeyGen, Encryption and Decryption algorithms for Mithril.PKE and the KeyGen, Encaps and Decaps algorithms for Mithril.KEM.

levels. The 128-bit and 256-bit targets correspond to NIST Security Level I and V [Nat16] and will allow direct comparison against existing proposals or standards. The 512-bit security level is a requirement in the PQC call for proposals by the Institute of Commercial Cryptography Standards (ICCS) in China [Ins25], so it may be of relevance for future designs. We show in Section 4 that Mithril and Octarine reduce to variants of Radical Ring-LWR and Ring-SIS. We further assume that the ring structure cannot be exploited in the attacks, as is common in parameter selection for rings based on power-of-two cyclotomics, hence we generate parameter sets based on LWR and SIS problems of column dimensions on the order of kn (see Proposition 2 and 3 for more detailed statements). Section 4.1 discusses what this assumption entails for radical rings specifically, and why the distortion bound of Proposition 1 (which concerns the canonical embedding) is independent of the concrete estimates below, which are computed in the monomial basis where the secret and error distributions are exactly independent and uniform. We report two metrics: the Core-SVP hardness as a conservative and comparable baseline, and the refined rop-based estimates from the lattice-estimator framework of Albrecht, Player and Scott [APS15] as an indication of the current best understanding of concrete attack costs. This dual view reflects the standard practice in the analysis of lattice-based schemes, where coarse but robust metrics are complemented by more detailed, yet assumption-dependent, evaluations. We summarize the parameter sets of Mithril in Table 1 and for Octarine in Table 2.

3.2 The Mithril Key Encapsulation Mechanism

In this section we describe all the components that are needed to build the Mithril key encapsulation mechanism (KEM). The key encapsulation mechanism Mithril.KEM is derived from Mithril.PKE using an implicit-rejection Fujisaki–Okamoto transform. The construction is similar to the lattice-based KEM Saber, so we align notation as much as possible.

Message space. In the setting of Module-LWE or Module-LWR, the message is embedded as an element of $\mathbf{R}_{2,n}$. As this consists of n coefficients of a single bit each, it naturally encodes n -bit messages. For Ring-LWR the message is encoded as an element of $\mathbf{S}_{2,n,k}$ instead, allowing nk -bit messages to be encoded. Since we do not need the full kn -bit message space, we embed our messages in the subspace where ℓ of the coefficients are non-zero, and the other $k - \ell$ coefficients are 0. As $n = 128$, we choose $\ell = 1, 2, 4$ for the respective parameter sets to make sure that $n\ell = \lambda$. To optimize the performance of the scheme, we select the ℓ *most* significant coefficients of the ring element. This minimizes the number of reductions modulo $x^k - (y + 2)$ and therefore saves some multiplications. Additionally, it significantly reduces the failure probability as we show in Section 4.2. In the implementation, all multiplications by zero coefficients can be omitted. More precisely, we use the notation $\mathbf{g}^{[k-\ell:k-1]}$ for the polynomial $\mathbf{g} = \sum_{i=0}^{k-1} \mathbf{g}_i(y)x^i \in \mathbf{S}_{2,n,k}$ where $\mathbf{g}_0(y), \dots, \mathbf{g}_{k-\ell-1}(y) = 0$, which we interpret as a vector $\mathbf{v} \in \mathbf{R}_{2,n}^k$ such that $\mathbf{v}_i = \mathbf{g}_{k-1-i}$.

Ciphertext compression. To reduce the size of the ciphertext, we apply the lossy function $\lfloor \frac{t}{p} \cdot \rfloor$, where the value of t depends on the parameter set. While this decreases the ciphertext size, it increases the failure probability (see Section 4.2). We do not apply compression to the public key.

Fujisaki–Okamoto transformation. Our construction uses the Hofheinz–Hövelmanns–Kiltz modular Fujisaki–Okamoto transform with implicit rejection $\mathbf{U}_m^\chi$ [HHK17] to the deterministic re-encryption form of Mithril.PKE. In this transform, the ciphertext is excluded from the encapsulation-time derivation of the candidate shared secret. We refer to Section 4.2 for the IND-CCA security discussion and the quoted QROM bounds.

3.3 The Octarine Digital Signature Scheme

The Fiat–Shamir with Aborts paradigm was originally proposed in [Lyu09] and forms the basis of several practical lattice signature schemes. We first give a high-level description of how the approach works in our setting. The construction is similar to other lattice-based signature schemes such as ML-DSA, so we align notation as much as possible. We motivate the design choices that are specific to Octarine. The complete algorithm specification is available in Figure 3.

Key generation. The signer samples a public matrix $\mathbf{A}$ and draws a secret vector $\mathbf{s}_1$ with small coefficients. The LWR rounding $\mathbf{b} = \lfloor \frac{p}{q} \mathbf{A} \mathbf{s}_1 \rfloor$ forms the main component of the public key. The rounding error can be deterministically computed as $\mathbf{s}_2 = \mathbf{A} \mathbf{s}_1 - \frac{q}{p} \mathbf{b}$ which is stored as part of the secret key. Intuitively, $\mathbf{s}_2$ is analogous to the explicit rounding error introduced in LWE contexts (with opposite sign). That is, we have the relation $\frac{q}{p} \mathbf{b} = \mathbf{A} \mathbf{s}_1 - \mathbf{s}_2$ that mimics Ring-LWE.

Signing. The signer draws a ring element $\mathbf{y}$ with uniformly random coefficients bounded by γ_1 , computes $\mathbf{w} = \mathbf{A} \mathbf{y}$, and extracts its high bits $\mathbf{w}_1 = \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$ as the commitment. Both γ_1 and γ_2 are powers of two so that sampling does not require rejection sampling and rounding is a simple (signed) bit shift. The challenge seed $\tilde{c}$ is derived by hashing the message digest μ together with $\mathbf{w}_1$; the challenge polynomial $c = \text{SampleInBall}(\tilde{c})$ is computed, and a candidate response $\mathbf{z} = \mathbf{y} + c \mathbf{s}_1$ is formed. Note that for Module-LWE the challenge c is an element of the ring with n coefficients, not a

Table 1: An overview of the parameter sets of Mithril. In parentheses we show the relative change compared to ML-KEM-512 and ML-KEM-1024, respectively.

	Mithril-128	Mithril-256	Mithril-512
n	128	128	128
k	5	9	18
ℓ	1	2	4
ϵ_q	13	13	14
ϵ_p	11	11	12
ϵ_t	3	4	6
$D_{\mathbf{A}}$	$\mathcal{U}_{\frac{q}{2}}$	$\mathcal{U}_{\frac{q}{2}}$	$\mathcal{U}_{\frac{q}{2}}$
D_s	$\mathcal{U}_2$	$\mathcal{U}_2$	$\mathcal{U}_2$
KEM secret key (bytes)	1 136 (−30%)	2 000 (−37%)	4 224
KEM public key (bytes)	944 (+18%)	1 648 (+5%)	3 520
KEM ciphertext (bytes)	928 (+21%)	1 712 (+9%)	3 840
Shared secret (bytes)	16 (−50%)	32 (−)	64
Failure probability ($\log_2$)	−603	−278	−569
BKZ block size (b)	443 (+12%)	879 (+6%)	1 802
Core-SVP (Classical) (bits)	129	257	526
Core-SVP (Quantum) (bits)	117	233	478
Refined estimate (gates)	153 (+9%)	277 (+6%)	539

vector of ring elements. In the context of Ring-LWR this would naïvely correspond to a full ring element in $\mathbf{S}_{q,n,k}$ with a total of kn coefficients. Instead, c is embedded into $\mathbf{S}_{q,n,k}$ as the constant component, i.e., $c := c(y) + \sum_{i=1}^{k-1} 0 \cdot x^i$. To prevent $\mathbf{z}$ and $\mathbf{w} + c\mathbf{s}_2$ from leaking information about the secret key we apply rejection sampling as usual. The parameters are chosen so that the expected number of repetitions is small: the analysis at the end of this section shows on average 2.1, 2.3, and 3.6 repetitions for Octarine-128, Octarine-256, and Octarine-512, respectively.

Verification. Verification is the standard Fiat–Shamir with Aborts signature verification instantiated with a radical ring that derives the (approximate) commitment $\mathbf{w}_1$ and challenge seed $\tilde{c}$ from $\mathbf{Az} - c \cdot \frac{q}{p} \mathbf{b}_1 \cdot 2^d$ and checks that it matches the challenge seed received in the signature.

Decomposition and public-key compression. We apply a similar public-key compression mechanism as is used by ML-DSA. However, the treatment of hints in Octarine can be simplified compared to ML-DSA. Since q and $2\gamma_2$ are both powers of two with $2\gamma_2 \mid q$, the decomposition into high-order and low-order parts is a clean bit extraction. Unlike the prime-modulus case, there is no boundary anomaly at $q - 1$ and the Decompose function is replaced by separate calls to HighBits_q and LowBits_q . This is also the case for the Power2Round function. The $\text{MakeHint}_q(z, r, m)$ function is equal to the one from ML-DSA, namely it returns 1 if $\text{HighBits}_q(r, m) = \text{HighBits}_q(r + z, m)$ and 0 otherwise. The UseHint_q function accumulates the hint bit into the high bits of a coefficient, adding or subtracting if the coefficient is strictly positive or non-positive, respectively. Note that due to the choice of $n = 1024$ we require 10 bits to encode each hint position in the polynomial, as opposed to 8 bits for ML-DSA, increasing the hint size.

Challenge generation. The function $\text{SampleInBall}(\tilde{c})$ deterministically maps a hash digest $\tilde{c}$ to a challenge polynomial c in B_τ , which is defined as the space of polynomials in

$(pk, sk) \leftarrow \text{Octarine.KeyGen}()$ 1: $\xi \leftarrow U(\mathbb{B}^{64})$ 2: $(\rho, \rho', K) \leftarrow H_{\text{kdf}}(\xi; \rho + \rho' + K)$ 3: $\mathbf{A} \leftarrow D_{\mathbf{A}}(\mathbf{S}_{q,n,k}; \rho)$ 4: $\mathbf{s}_1 \leftarrow D_{\mathbf{s}}(\mathbf{S}_{q,n,k}; \rho')$ 5: $\mathbf{t} = \mathbf{A}\mathbf{s}_1 \in \mathbf{S}_{q,n,k}$ 6: $\mathbf{b} = \lfloor \frac{p}{q}\mathbf{t} \rfloor \in \mathbf{S}_{p,n,k}$ 7: $\mathbf{s}_2 = \mathbf{t} - \frac{q}{p}\mathbf{b} \in \mathbf{S}_{q/p,n,k}$ 8: $(\mathbf{b}_1, \mathbf{b}_0) \leftarrow \text{Power2Round}_p(\mathbf{b}, 2^d)$ 9: $pk \leftarrow (\rho, \mathbf{b}_1)$ 10: $tr \leftarrow H_{tr}(pk; tr)$ 11: $sk \leftarrow (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{b}_0)$ 12: return (pk, sk) $\text{bool} \leftarrow \text{Octarine.Verify}(pk, M, \sigma)$ 13: $(\rho, \mathbf{b}_1) \leftarrow pk$ 14: $(\tilde{c}, \mathbf{z}, \mathbf{h}) \leftarrow \sigma$ 15: $\mathbf{A} \leftarrow D_{\mathbf{A}}(\mathbf{S}_{q,n,k}; \rho)$ 16: $tr \leftarrow H_{tr}(pk; tr)$ 17: $\mu \leftarrow H_{\text{mu}}(tr, M; \mu)$ 18: $c \leftarrow \text{SampleInBall}(\tilde{c})$ 19: $\mathbf{w}'_{\text{Approx}} \leftarrow \mathbf{A}\mathbf{z} - c\frac{q}{p}\mathbf{b}_1 2^d$ 20: $\mathbf{w}'_1 \leftarrow \text{UseHint}_q(\mathbf{h}, \mathbf{w}'_{\text{Approx}}, 2\gamma_2)$ 21: $\tilde{c}' \leftarrow H_{\text{ch}}(\mu, \mathbf{w}'_1; \tilde{c})$ 22: return $[[\ \mathbf{z}\ _{\infty} < \gamma_1 - \beta]]$ 23: and $[[\tilde{c} = \tilde{c}']]$ and $[[\text{HW}(\mathbf{h}) \leq \omega]]$	$\sigma \leftarrow \text{Octarine.Sign}(sk, M, rnd)$ 1: $(\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{b}_0) \leftarrow sk$ 2: $\mathbf{A} \leftarrow D_{\mathbf{A}}(\mathbf{S}_{q,n,k}; \rho)$ 3: $\mu \leftarrow H_{\text{mu}}(tr, M; \mu)$ 4: $\rho'' \leftarrow H_{\text{rhopp}}(K, rnd, \mu; \rho'')$ 5: $\kappa \leftarrow 0$ 6: $(z, h) \leftarrow \perp$ 7: while $(z, h) = \perp$ do 8: $\mathbf{y} \leftarrow D_{\mathbf{y}}(\mathbf{S}_{q,n,k}; \rho'', \kappa)$ 9: $\mathbf{w} = \mathbf{A} \cdot \mathbf{y}$ 10: $\mathbf{w}_1 = \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$ 11: $\tilde{c} = H_{\text{ch}}(\mu, \mathbf{w}_1; \tilde{c})$ 12: $c = \text{SampleInBall}(\tilde{c})$ 13: $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$ 14: $\mathbf{r}_0 = \text{LowBits}_q(\mathbf{w} + c\mathbf{s}_2, 2\gamma_2)$ 15: if $\ \mathbf{r}_0\ _{\infty} \geq \gamma_2 - \beta$ or 16: $\ \mathbf{z}\ _{\infty} \geq \gamma_1 - \beta$ then 17: $(\mathbf{z}, \mathbf{h}) \leftarrow \perp$ 18: else 19: $\mathbf{h} = \text{MakeHint}_q(-c\frac{q}{p}\mathbf{b}_0,$ 20: $\mathbf{w} + c\mathbf{s}_2 + c\frac{q}{p}\mathbf{b}_0, 2\gamma_2)$ 21: if $\ c\frac{q}{p}\mathbf{b}_0\ _{\infty} \geq \gamma_2$ or 22: $\text{HW}(\mathbf{h}) > \omega$ then 23: $(\mathbf{z}, \mathbf{h}) \leftarrow \perp$ 24: $\kappa = \kappa + 1$ 25: return $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$
--	--

Figure 3: The KeyGen, Sign, and Verify algorithms for the signature scheme Octarine.

$\mathbf{R}_{q,n}$ with exactly τ non-zero coefficients, each in $\{-1, +1\}$. The algorithm is an inside-out Fisher–Yates shuffle applied to the n integer coefficients of a polynomial in $\mathbf{R}_{q,n}$, as is used for ML-DSA. In the signature algorithms, c is embedded into $\mathbf{S}_{q,n,k}$ as the constant component, i.e., $c := c(y) + \sum_{i=1}^{k-1} 0 \cdot x^i$. This mimics the case of Module-LWE where c is a single ring element, not a full module element. Thus, for $\mathbf{s} = \sum_{i=0}^{k-1} \mathbf{s}_i x^i$, multiplication by c acts component-wise as $c\mathbf{s} = \sum_{i=0}^{k-1} (c\mathbf{s}_i)x^i$.

The challenge entropy is $\log_2 \binom{n}{\tau} + \tau$ bits. This is the entropy of the uniform distribution over all signed Hamming-weight- τ challenges. As we choose $n = 1024$, we can select smaller τ than Module-LWE proposals with $n = 256$, allowing to reduce the size of β and reducing the number of aborts in the signing loop. We select τ to be the smallest value such that the exact challenge entropy is at least λ ; Table 2 reports the corresponding entropy values rounded down to whole bits.

Fiat–Shamir with Aborts loop. The Fiat–Shamir with Aborts loop rejects when any of the following conditions holds: $\|\mathbf{z}\|_{\infty} \geq \gamma_1 - \beta$, $\|\mathbf{r}_0\|_{\infty} \geq \gamma_2 - \beta$, $\|c\frac{q}{p}\mathbf{b}_0\|_{\infty} \geq \gamma_2$, or $\text{HW}(\mathbf{h}) > \omega$. The first two conditions dominate the rejection probability and can be estimated analytically as follows. One can show (e.g., see [LDK⁺22, Section 3.4]) that each coefficient z_i of $\mathbf{z}$ satisfies $|z_i| < \gamma_1 - \beta$ if and only if $-\gamma_1 + \beta - \sigma_i < y_i < \gamma_1 - \beta - \sigma_i$, where y_i and σ_i are the corresponding coefficients of $\mathbf{y}$ and $c\mathbf{s}_1$, respectively. As each of

Table 2: An overview of the parameter sets of **Octarine**. In parentheses we show the relative change compared to ML-DSA-44 and ML-DSA-87, respectively.

	Octarine-128	Octarine-256	Octarine-512
n	1024	1024	1024
k	1	2	5
$\epsilon_q = \log_2 q$	24	24	24
$\epsilon_p = \log_2 p$	21	22	22
η	4	2	2
d	11	13	14
τ	16	36	87
λ	128	256	512
γ_1	2^{18}	2^{19}	2^{21}
γ_2	2^{17}	2^{18}	2^{20}
$\beta = \tau\eta$	64	72	174
ω	79	210	399
Challenge entropy	131	257	512
# Repetitions	2.1	2.3	3.6
$D_{\mathbf{A}}$	$\mathcal{U}_{\frac{q}{2}}$	$\mathcal{U}_{\frac{q}{2}}$	$\mathcal{U}_{\frac{q}{2}}$
$D_{\mathbf{s}_1}$	$\mathcal{U}_\eta$	$\mathcal{U}_\eta$	$\mathcal{U}_\eta$
Private key (bytes)	2 432 (−5.0%)	4 608 (−5.9%)	11 776
Public key (bytes)	1 344 (+2.4%)	2 368 (−8.7%)	5 184
Signatures (bytes)	2 564 (+6.0%)	5 449 (+17.8%)	14 713
<u>LWR Hardness (BDD)</u>			
BKZ block size (b)	424 (−3.1%)	924 (+4.6%)	2181
Classical Core-SVP (0.292 b)	124	270	637
Quantum Core-SVP (0.265 b)	112	245	578
Refined estimate (gates)	150	291	686
<u>SIS Hardness (EF-CMA)</u>			
BKZ block size (b)	439 (+3.8%)	910 (+0.1%)	2340
Classical Core-SVP (0.292 b)	128	266	683
Quantum Core-SVP (0.265 b)	116	241	620
Refined estimate (gates)	156	288	691

the kn coefficients of $\mathbf{y}$ is uniformly distributed in $[-\gamma_1, \gamma_1 - 1]$, the probability that $\mathbf{z}$ passes the bound check is

$$\Pr[||\mathbf{z}||_\infty < \gamma_1 - \beta] = \left(\frac{2(\gamma_1 - \beta) - 1}{2\gamma_1} \right)^{kn} = \begin{cases} 0.777 & \text{for Octarine-128,} \\ 0.753 & \text{for Octarine-256,} \\ 0.653 & \text{for Octarine-512.} \end{cases}$$

This probability does not depend on any secret data. Similarly, with the assumption that the centered low bits $\mathbf{r}_0 = \text{LowBits}_q(\mathbf{w} + c\mathbf{s}_2, 2\gamma_2)$ are uniformly distributed in $[-\gamma_2, \gamma_2 - 1]$, the probability of passing the bound check is

$$\Pr[||\mathbf{r}_0||_\infty < \gamma_2 - \beta] = \left(\frac{2(\gamma_2 - \beta) - 1}{2\gamma_2} \right)^{kn} = \begin{cases} 0.604 & \text{for Octarine-128,} \\ 0.568 & \text{for Octarine-256,} \\ 0.427 & \text{for Octarine-512.} \end{cases}$$

As the success rate for $\mathbf{r}_0$ is lower than for $\mathbf{z}$, it is more efficient to do the rejection based on $\mathbf{r}_0$ first in the implementation. This way fewer computations have been completed when rejection occurs. We empirically confirm that the bounds checks for $\mathbf{z}$ and $\mathbf{r}_0$ dominate the rejection probability. In a run of the reference implementation generating 5000 signatures each for Octarine-128, Octarine-256, and Octarine-512, the numbers of rejected iterations caused by the $c_p^2 \mathbf{b}_0$ -bound test were 0, 30, and 1, respectively, and the corresponding numbers caused by the hint-weight test were 36, 32, and 8. The measured average number of loop iterations was 2.10, 2.29, and 3.54, respectively, in close agreement with the predicted 2.1, 2.3, and 3.6.

4 Radical Ring Security

The main hard problem underlying Mithril and Octarine is Ring-LWR over a ring $\mathbf{R}_q$ which we recall in Definition 1. In case of Mithril and Octarine the ring $\mathbf{R}_q$ is instantiated with the radical ring $\mathbf{S}_{q,n,k}$, and we refer to this instance as the **RR-LWR** problem. We shall write $\text{Adv}_{k,D_s,N}^{\text{R-LWR}}(\mathcal{A}) := \text{Adv}_{k,q,p,D_s,N}^{\text{R-LWR}}(\mathcal{A})$ when it is clear we are working over the ring $\mathbf{S}_{q,n,k}$ with rounding parameter p .

Definition 1 (Decision Ring-LWR). Let $q > p \geq 2$ and define the rings $\mathbf{R}_{\nu,k} := \mathbf{Z}_\nu[x]/(f_k(x))$ where k is an arbitrary positive integer and $\nu \in \{p, q\}$, and let D_s be a secret distribution over $\mathbf{R}_{q,k}$. For a fixed secret $\mathbf{s} \in \mathbf{R}_{q,k}$ sampled from D_s , the decision Ring-LWR problem is to distinguish the following two distributions given a number of samples N :

$$\begin{aligned} \text{Real : } & [(\mathbf{A}_i, \lfloor \frac{p}{q} \mathbf{A}_i \mathbf{s} \rfloor)]_{i=0}^{N-1}, \mathbf{A}_i \leftarrow \mathbf{R}_{q,k} \text{ uniformly random for all } i; \\ \text{Rand : } & [(\mathbf{A}_i, \mathbf{b}_i)]_{i=0}^{N-1}, \mathbf{A}_i \leftarrow \mathbf{R}_{q,k}, \mathbf{b}_i \leftarrow \mathbf{R}_{p,k} \text{ uniformly random for all } i. \end{aligned}$$

One defines

$$\text{Adv}_{k,q,p,D_s,N}^{\text{R-LWR}}(\mathcal{A}) = |\Pr[\mathcal{A}(\text{Real}) = 1] - \Pr[\mathcal{A}(\text{Rand}) = 1]|$$

to be the advantage for a distinguisher $\mathcal{A}$. The decision Ring-LWR assumption states that this advantage is negligible for all efficient distinguishers $\mathcal{A}$.

4.1 Security of the RR-LWR Instance

As discussed in [BRVW26, Section 4.1], a secure instance of Ring-LWR [Pei16] requires the secrets to be sampled in $\mathcal{O}_K^\vee$ and the error in a spherical distribution in the canonical embedding of K (or actually $K \otimes_{\mathbb{Q}} \mathbb{R}$, but we omit $\mathbb{R}$ as it does not impact our analysis).

We improve on the analysis from [BRVW26, Section 4.1] by proving in Proposition 1 that the total distortion on the error is less or equal than $3^{(k-1)/k} < 3$, in particular demonstrating that the distortion from sampling in the ring of integers and in monomial basis $x^i y^j$ partially cancel each other out. In the terminology of Proposition 1 we show that sampling the secret in the ring $\mathcal{R}$ with monomial basis $x^i y^j$ as opposed to $\mathcal{O}_K^\vee$ incurs a distortion on the error samples with respect to a spherical Gaussian error in $\sigma(K)$. This distortion comes from the linear transformation $D_t^{-1} \circ L$ and we show that it is less than or equal to $3^{(k-1)/k} < 3$, approaching 3 as k grows.

Proposition 1. *Let $f = (x^k - 2)^n + 1 \in \mathbf{Z}[x]$ for a power-of-two $n \geq 2$ and positive integer k , and $\mathcal{O}_K$ be the ring of integers of the number field $K = \mathbb{Q}[x]/(f(x)) = \mathbb{Q}[\alpha]$, and let $\mathcal{O}_K^\vee = \frac{1}{t} \mathcal{O}_K$ be the dual of $\mathcal{O}_K$ for $t = nk\alpha^{(k-1)}$. Let $\xi_{2n}, \xi_k \in \mathbb{C}$ be primitive $2n$ -th and k -th roots of unity, respectively, and write $r_{i,j} = (\xi_{2n}^{2j+1} + 2)^{1/k} \cdot \xi_k^i$ for the roots of f for $0 \leq i < k$ and $0 \leq j < n$. Denote by $\sigma = (\sigma_{i,j})_{i,j} : g \mapsto (g(r_{i,j}))$ the canonical embedding of K into $\mathbb{C}^{nk}$. The diagram*

$$\begin{array}{ccc}
 \mathcal{O}_K^\vee & \xrightarrow{\cdot t} & \mathcal{O}_K = \mathbb{Z}[x]/((x^k - 2)^n + 1) \\
 \sigma \downarrow & & \downarrow \phi : x^k \mapsto y + 2 \\
 & & \mathcal{R} = (\mathbb{Z}[y]/(y^n + 1))[x]/(x^k - y - 2) \\
 \mathbb{C}^{nk} & \xrightarrow{\cdot D_t} & \mathbb{C}^{nk} \\
 & & \downarrow L : (I_n \otimes F_k)PD(I_k \otimes F_n)
 \end{array}$$

is commutative, where

- F_n is the matrix $(\xi_{2n}^{(2i+1)j})_{i,j}$ for $0 \leq i, j < n$,
- F_k is the k -point Fourier matrix $(\xi_k^{ij})_{i,j}$ for $0 \leq i, j < k$,
- $D = \text{diag}(\Delta^0, \Delta^1, \dots, \Delta^{k-1})$ for $\Delta = \text{diag}(r_{0,0}, \dots, r_{0,n-1})$,
- $D_t = \text{diag}(\sigma_{i,j}(t))$ for $0 \leq i < k$ and $0 \leq j < n$,
- P is the permutation matrix that swaps row $\ell n + m$ with row $mk + \ell$ for $0 \leq \ell < k$ and $0 \leq m < n$.

The distortion of $D_t^{-1}L : \mathcal{R} \rightarrow \mathbb{C}^{nk}$, defined as $\max S(D_t^{-1}L) / \min S(D_t^{-1}L)$ for the set $S(D_t^{-1}L)$ of singular values of $D_t^{-1}L$, is equal to $\max_j (|\xi_{2n}^{2j+1} + 2|)^{(k-1)/k}$ and is less or equal than $3^{(k-1)/k}$.

Proof. The main observation is that the map $\mathcal{R} \rightarrow \mathbb{C}^{nk}$ is constructed by evaluating (x, y) at all of the $((\xi_{2n}^{2j+1} + 2)^{1/k} \cdot \xi_k^i, \xi_{2n}^{2j+1})$. Commutativity of the diagram follows from the fact that $\mathcal{O}_K \rightarrow \mathbb{C}^{nk}$ is the canonical embedding given by evaluating x at the values $r_{i,j} = (\xi_{2n}^{2j+1} + 2)^{1/k} \cdot \xi_k^i$, so that y in $\mathcal{R}$ should be evaluated at $r_{i,j}^k - 2 = \xi_{2n}^{2j+1}$. The multiplication by t corresponds to D_t under the canonical embedding of K , completing the commutativity.

We show that L can be decomposed as a product of matrices. For each i, j , the map $\mathcal{R} \rightarrow \mathbb{C}^{nk}$ can be decomposed as the three mappings $(x, y) \mapsto (x, \xi_{2n}^{2j+1})$, $x \mapsto (\xi_{2n}^{2j+1} + 2)^{1/k} x$ and $x \mapsto \xi_k^i$. For $g = \sum_i g_i(y)x^i \in \mathcal{R}$ the first map equates to $I_k \otimes F_n$ by letting row $in + j$ correspond to the evaluation $g_i(\xi_{2n}^{2j+1})$. The second map multiplies row $in + j$ by $r_{0,j}^i$ which corresponds to left multiplication by the diagonal matrix D . Finally, we compute the evaluation at ξ_k^i for the coefficients from row $in + j$ for $0 \leq i < k$ and fixed j . To achieve this, we do the permutation P and apply F_k to each of the n blocks of k rows each, i.e., multiply by $I_n \otimes F_k$.

It remains to compute the singular values and distortion of $D_t^{-1} \circ L$. Since $F_k F_k^* = kI_k$ and $F_n F_n^* = nI_n$ all their singular values are $\sqrt{k}$ and $\sqrt{n}$ respectively. Similarly, the singular values of P are all trivial. A priori, D_t^{-1} does not commute with $I_n \otimes F_k$ as the diagonal entries are of the form $\sigma_{i,j}(t)^{-1} = \frac{1}{nk} (\xi_{2n}^{2j+1} + 2)^{-(k-1)/k} \xi_k^{-(k-1)i}$. However, we observe that $D_t^{-1}(I_n \otimes F_k) = (D'_t \otimes I_k)(I_n \otimes F_k S) = (I_n \otimes F_k S)(D'_t \otimes I_k)$ for a permutation matrix S and $D'_t = \text{diag}(\sigma_{0,j}(t)^{-1})$. This works since the multiplication by $\xi_k^{-(k-1)i}$ effectively rotates the elements of F_k , leading to the permutation S .

Now $(D'_t \otimes I_k)PD$ is a diagonal matrix (up to permutation with P) with entries of the form $\sigma_{0,j}(t)^{-1} r_{0,j}^i = \frac{1}{nk} (\xi_{2n}^{2j+1} + 2)^{(i-(k-1))/k}$. Therefore, the singular values of $D_t^{-1} \circ L$ are exactly the $\frac{1}{\sqrt{nk}} |\xi_{2n}^{2j+1} + 2|^{(i-(k-1))/k}$ by accounting for the factor $\sqrt{nk}$ from F_k and F_n . The largest singular value is $1/\sqrt{nk}$ for $i = k - 1$, and the smallest is $\frac{1}{\sqrt{nk}} \max_j (|\xi_{2n}^{2j+1} + 2|)^{-(k-1)/k}$ for $i = 0$. Hence the distortion is given by their quotient $\max_j (|\xi_{2n}^{2j+1} + 2|)^{(k-1)/k}$ which is strictly less than $3^{(k-1)/k}$, as $1 < |\xi_{2n}^{2j+1} + 2| < 3$. $\square$

The exact distortion. Since $\xi_{2n}^{2j+1} = e^{i\pi(2j+1)/n}$ we have $|\xi_{2n}^{2j+1} + 2|^2 = 5 + 4\cos(\pi(2j+1)/n)$, which is maximal at $j = 0$. The distortion of Proposition 1 is therefore known in closed form as

$$\max_j (|\xi_{2n}^{2j+1} + 2|)^{(k-1)/k} = (5 + 4\cos(\pi/n))^{\frac{k-1}{2k}}$$

and approaches $3^{(k-1)/k}$ as n increases. For $n = 128$ it equals $2.9998^{(k-1)/k}$, and for $n = 1024$ it equals $2.99997^{(k-1)/k}$. Concretely, the distortion is 2.41, 2.66 and 2.82 for Mithril-128, Mithril-256 and Mithril-512, and 1, 1.73 and 2.41 for Octarine-128, Octarine-256 and Octarine-512. The case $k = 1$ is degenerate: the radical ring collapses to the power-of-two cyclotomic ring $\mathbf{Z}_q[y]/(y^n + 1)$ under $x \mapsto y + 2$, and the distortion is exactly 1. Thus Octarine-128 is a plain Ring-LWR instance over a degree-1024 cyclotomic ring, with no distortion whatsoever.

Relation to the bounds of [BRVW26]. We note that [BRVW26, Section 4.1] reports two bounds, first a bound $|r_{i,j}^{(k-1)}|$ on the distortion incurred from sampling the secret in $\mathcal{O}_K$ and second a bound $\beta_j \sqrt{nk}$ on the singular values introduced by sampling in the monomial basis as opposed to the canonical basis, where they define $\beta = \max_j \beta_j$ with $\beta < 2.79$ for $k \leq 15$ with empirical analysis. The first bound is derived from a transformation that corresponds to our map D_t while the second is derived from a linear transformation that corresponds to $L \circ \phi$. What Peikert’s criterion constrains, however, is the error relative to the dual $\mathcal{O}_K^\vee = \frac{1}{t}\mathcal{O}_K$, which is different from the bounds in [BRVW26, Section 4.1] and corresponds to the distortion on the map $D_t^{-1} \circ L$. Proposition 1 bounds the distortion of the correct transformation $D_t^{-1} \circ L$ and the proof reveals cancellations that occur which are not captured by bounding the maps D_t and $L \circ \phi$ independently, leading to a tighter overall bound on the distortion.

Impact of the distortion κ . Proposition 1 bounds the shape of the error but does not by itself quantify a security loss, so we record the standard consequence. Let $\kappa = \max S / \min S$ be the distortion and Σ the covariance of the error in the canonical embedding, so that $s_{\min}^2 I \preceq \Sigma \preceq s_{\max}^2 I$ with $s_{\max}/s_{\min} = \kappa$. Writing $\Sigma = s_{\min}^2 I + (\Sigma - s_{\min}^2 I)$ with $\Sigma - s_{\min}^2 I \succeq 0$, a Gaussian skewed error is a spherical Gaussian of width $s_{\min}$ plus an independent term that the reduction can sample itself. Hardness of RR-LWR therefore follows from hardness of the spherical Ring-LWR instance whose width is smaller by a factor at most $\kappa < 3$, a loss of at most $\log_2 3 < 1.59$ bits of error width, uniformly in n and k , and nothing at all for $k = 1$.

This statement lives in the Gaussian model of Peikert’s criterion and concerns instantiation; it is not the route by which we select concrete parameters, and the estimator runs do not depend on the distortion. Parameter selection is carried out entirely in the monomial basis, where the secret $\mathbf{s}_1$ and the rounding error $\mathbf{s}_2 = \mathbf{A}\mathbf{s}_1 - \frac{q}{p}\mathbf{b}$ are independent and uniform on $[-\eta, \eta - 1]$ and $[-\frac{q}{2p}, \frac{q}{2p} - 1]$ respectively (the latter under the standard assumption that $\mathbf{A}\mathbf{s}_1$ is uniform, cf. Section 3); we instantiate the lattice estimator with an unstructured LWE instance of dimension kn , modulus q , and precisely these distributions. Detailed script which run these experiments will be released soon. A larger distortion would therefore not invalidate our concrete estimates, only the claim that RR-LWR is a sensible instantiation of Ring-LWR.

Structural attacks. Since concrete parameters are selected as though the instance were unstructured, we argue briefly that the extra structure of $\mathbf{S}_{q,n,k}$ does not appear to help an attacker. As f is irreducible [BRVW26, Lemma 1], K is a number field of degree kn and RR-LWR is a genuine Ring-LWR instance, in particular it is not the multivariate m -RLWE problem [PUTPPG15, PUTPPG16, PUTPPG17, PUTPG⁺21], whose ring decomposes as a tensor product, so the attacks of [BCV20] do not apply. Under (1), K contains $L = \mathbf{Q}(\xi_{2n})$ with $[K : L] = k$, but the subfield attacks on overstretched NTRU [ABD16, KF17] act on a *multiplicative* relation, whereas an RR-LWR sample is additive and $N_{K/L}$ inflates the error

to norms growing like $\|\mathbf{s}\|^k$. The L -linear projections onto the x^m -components do commute with the rounding, but together they return the rank- k Module-LWR instance over L of the same total dimension kn : descending to the subfield recovers the module viewpoint without gaining anything. The public matrix is a twisted circulant drawn from a kn -dimensional subspace of the $(kn)^2$ -dimensional matrix space, the same reduction of randomness that Ring-LWE/LWR exhibits over plain LWE. Each generator is individually uniform, as $y + 2$ has odd norm $2^n + 1$ in $\mathbf{R}_{q,n}$ and is thus a unit modulo any power of two; moreover K/L is Galois only when $k \mid 2n$, so K admits fewer automorphisms than a cyclotomic field of the same degree. We are not aware of any attack exploiting this structure beyond the constant-factor speedups already available for Ring- and Module-LWE/LWR.

4.2 Security of Mithril

The security proof of Mithril consists of three independent steps. First, we reduce the security of Mithril.PKE to a similar PKE design based on RR-LWR. Specifically, we show that the security of $\text{Mithril.PKE} := \text{Mithril}_\ell.\text{PKE}$ reduces to the security of $\text{Mithril}_k.\text{PKE}$, while the failure probability is no larger than that of $\text{Mithril}_k.\text{PKE}$, where k is the degree of the radical ring extension. We prove this in Lemma 1. This handles the main difference between standard Ring-LWR based PKEs that use all shared secret bits (like Mithril_k) and Mithril which uses only ℓ out of k coefficients.

Second, the reduction from $\text{Mithril}_k.\text{PKE}$ to RR-LWR is then a standard result from D’Anvers [D’A21, Theorem 2–3]. Third, our construction follows the modular FO-with-implicit-rejection transform U_m^χ of Hofheinz–Hövelmanns–Kiltz [HHK17] applied to the deterministic re-encryption form of Mithril.PKE. The reduction from the IND-CCA security of Mithril.KEM to Mithril.PKE in the QROM uses the reduction from Kuchta, Sakzad, Stehlé, Steinfeld and Sun [KSS⁺20, Corollary 4.7]. In summary, we prove the security of Mithril with the following reductions

$$\begin{array}{ccc}
 \text{IND-CCA}(\text{Mithril}) & \xrightarrow{[\text{KSS}^+20, \text{Corollary 4.7}]} & \text{IND-CPA}(\text{Mithril}_\ell) \\
 \downarrow & & \downarrow \text{Lemma 1} \\
 \text{RR-LWR} & \xleftarrow{[\text{D’A21, Theorem 2–3}]} & \text{IND-CPA}(\text{Mithril}_k)
 \end{array}$$

and state the full security reduction in Proposition 2.

IND-CPA security of Mithril.PKE. We begin by reducing the IND-CPA security of Mithril.PKE to $\text{Mithril}_k.\text{PKE}$.

Lemma 1. *For any adversary $\mathcal{A}$ running the IND-CPA experiment for Mithril_ℓ there exists an adversary $\mathcal{B}$ running the IND-CPA experiment for Mithril_k such that*

$$\text{Adv}_{\text{Mithril}_\ell.\text{PKE}}^{\text{IND-CPA}}(\mathcal{A}) = \text{Adv}_{\text{Mithril}_k.\text{PKE}}^{\text{IND-CPA}}(\mathcal{B})$$

with approximately the same running time. Moreover, writing δ_ℓ for the failure probability of Mithril_ℓ , we have $\delta_\ell \leq \delta_k$.

Proof. Let $pk = (\rho_{\mathbf{A}}, \mathbf{b})$ be the public key generated by the challenger of $\mathcal{B}$. The adversary $\mathcal{B}$ forwards this public key to $\mathcal{A}$. $\mathcal{A}$ returns challenge messages m_0 and m_1 , which are (ℓn) -bit messages. $\mathcal{B}$ embeds m_0 and m_1 into the top ℓn bits of larger (kn) -bit messages $\mu_0 = 0^{(k-\ell)n} \parallel m_0$ and $\mu_1 = 0^{(k-\ell)n} \parallel m_1$ and sends them to its challenger. The challenger returns a ciphertext $c = (c_\mu, \mathbf{b}')$ which $\mathcal{B}$ converts to $c^* = (c_\mu^{[k-\ell:k-1]}, \mathbf{b}')$, which is a valid ciphertext for Mithril_ℓ . Finally $\mathcal{A}$ returns the bit b' upon receiving c^* , which $\mathcal{B}$ forwards to its challenger. As c^* corresponds to the encryption of m_b with Mithril_ℓ and c corresponds to the encryption of μ_b with Mithril_k , the advantages of $\mathcal{A}$ and $\mathcal{B}$ are exactly the same.

Since $\ell \leq k$ by the design of the PKE, the ciphertext of Mithril_ℓ is a subset of the ciphertext of Mithril_k . Therefore if decryption fails for Mithril_ℓ it also fails for Mithril_k , and it follows that $\delta_\ell \leq \delta_k$. $\square$

IND-CCA security of Mithril.KEM. Now we state the main result of this section. We observe that the reduction depends on both the failure probability δ_ℓ of Mithril as well as η , which is chosen so that Mithril.PKE is η -injective according to [KSS⁺20, Definition 4.3]. As a result parameters should be selected such that the failure probability is negligible and we show that this is the case for Mithril in the next section. For Proposition 2 to yield a concrete security bound for Mithril , the value of η must be quantified. Bindel, Hamburg, Hövelmanns, Hülsing and Persichetti [BHH⁺19, Appendix D] provide a heuristic analysis for LWE, which applies equally to LWR, that η should be on the order of q^{-kn} and is therefore negligible. Ding, Esgin, Sakzad and Steinfeld [DESS22] compute the exact value of η in the context of ML-KEM. We leave a detailed analysis of η for Mithril for future work.

Proposition 2 (QROM IND-CCA security of Mithril). *Let $\mathcal{A}$ be an adversary against the IND-CCA security of Mithril issuing at most q_{ss} (resp. q_{hpk}) quantum queries to H_{ss} (resp. H_{hpk}) with query depth at most d_{ss} (resp. d_{hpk}) and at most q_{dec} classical queries to the decapsulation oracle of Mithril . Then we can construct a Radical-Ring LWR distinguisher $\mathcal{B}_1$ and a PRF adversary $\mathcal{B}_2$ against H_{ss} issuing at most q_{dec} queries, of approximately the same running time as $\mathcal{A}$, and satisfying*

$$\begin{aligned} \text{Adv}_{\text{Mithril.KEM}}^{\text{IND-CCA}}(\mathcal{A}) &\leq 8d_{hpk}(d_{ss} + 1) \left(2 \cdot \text{Adv}_{k, D_s, 2}^{\text{RR-LWR}}(\mathcal{B}_1) + 8(3q_{ss} + 1)/|M| \right) \\ &\quad + 6(3q_{ss} + q_{dec}) \cdot ((8d_{ss} + 1) \cdot \delta_\ell + \sqrt{3\eta}) \\ &\quad + (4d_{hpk} + 12)\eta + 2 \cdot \text{Adv}_{H_{ss}}^{\text{PRF}}(\mathcal{B}_2), \end{aligned}$$

where η is such that Mithril.PKE is η -injective and δ_ℓ is the failure probability.

Proof. The main result reducing the IND-CCA security of Mithril.KEM to the IND-CPA security of Mithril_ℓ .PKE follows from [KSS⁺20, Corollary 4.7] using the measure-rewind-measure technique, instantiating it with parameters from Mithril (i.e., Mithril_ℓ by definition), in particular its failure probability δ_ℓ . We combine this result with Lemma 1 to reduce IND-CPA security of Mithril_ℓ to IND-CPA security of Mithril_k , which is a standard PKE based on Ring-LWR instantiated with a radical ring. This allows us to apply the results from the thesis of D’Anvers [D’A21, Theorem 2-3] to reduce to a Ring-LWR problem. While [D’A21, Theorem 2] reduces to the sum of two distinct Ring-LWR problems $\text{Adv}_{k, q, p, D_s, 1}^{\text{R-LWR}}(\mathcal{B}_2^{(1)})$ and $\text{Adv}_{k, q, p, D_s, 2}^{\text{R-LWR}}(\mathcal{B}_2^{(2)})$, we slightly simplify (and weaken) the statement by reducing to $2 \cdot \text{Adv}_{k, q, p, D_s, 2}^{\text{R-LWR}}(\mathcal{B}_2)$. $\square$

Decryption failures. It remains to show that the decryption failure rate δ_ℓ is small for concrete parameter sets. As with most LWE/LWR-based schemes, decryption is not always perfectly correct. A decryption failure occurs when the deterministic rounding errors, combined with the interaction between the secret key and ciphertext coefficients, accumulate sufficiently to shift a decrypted coefficient across the decoding boundary.

The decryption failures for Mithril arise in a similar fashion as other Ring- or Module-LWR schemes, such as Saber. We use a Saber-style heuristic estimator for the failure probability, following the calculation framework of [DKR⁺20, Theorem 6.2]. Translated to our notation, the estimator is based on the event

$$\delta_\ell = \Pr[\|(\mathbf{e}\mathbf{s}')^{[k-\ell:k-1]} - (\mathbf{e}'\mathbf{s})^{[k-\ell:k-1]} + \mathbf{e}_r\|_\infty > \frac{q}{4}]$$

where $\mathbf{e}_r$ is represented in $\mathbf{R}_q^\ell$ with centered coefficients in $[-q/(2t), q/(2t) - 1]$. Note that one may expect the bound $p/4$ due to reduction modulo p in Line 14 of Figure 2, but this is converted to $q/4$ when $\mathbf{bs}'$ and $\mathbf{b}'\mathbf{s}$ are lifted back to $\mathbf{R}_q$ by a multiplication with q/p to compute the error terms.

We define

$$\Delta_m(y) = \sum_{i=0}^m \mathbf{e}_{m-i}(y) \mathbf{s}'_i(y) + \sum_{i=m+1}^{k-1} (y+2) \mathbf{e}_{m-i+k}(y) \mathbf{s}'_i(y),$$

where Δ_m represents the coefficient of x^m of $\mathbf{es}'$. We observe that for each i , each coefficient of $\mathbf{e}_{m-i}(y) \mathbf{s}'_i(y)$ is a sum of products of two unique coefficients of $\mathbf{e}_{m-i}(y)$ and $\mathbf{s}'_i(y)$ without overlap in the coefficient indices. Hence, for each coefficient of $\mathbf{e}_{m-i}(y) \mathbf{s}'_i(y)$ the distribution can be modeled as the sum of n independent random variables, each of which is the product of two uniform distributions (equal to $[-q/(2p), q/(2p) - 1]$ for the choice of our parameter sets). We denote by X_0 the distribution of a coefficient of $\mathbf{e}_{m-i}(y) \mathbf{s}'_i(y)$. In the case of Saber, or Ring-LWR based on power-of-two cyclotomics, X_0 is the main distribution used to compute the failure probability.

While one may be tempted to similarly model $(y+2) \mathbf{e}_{m-i+k}(y) \mathbf{s}'_i(y)$ as the sum of two random variables representing a product $2 \mathbf{e}_{m-i+k}(y) \mathbf{s}'_i(y)$ and $y \mathbf{e}_{m-i+k}(y) \mathbf{s}'_i(y)$ (a nega-cyclic rotation preserving the coefficient distributions up to sign) respectively, their distributions do not represent *independent* random variables. As a result of the multiplication with y , each coefficient is a sum of products of elements of $\mathbf{e}_{m-i+k}(y)$ and $\mathbf{s}'_i(y)$ where indices do overlap. This introduces covariances that can have a significant impact on the failure probability. To account for this, we have to make sure we track the *joint* probabilities of adjacent terms even when computing the distribution of a single coefficient of $(y+2) \mathbf{e}_{m-i+k}(y) \mathbf{s}'_i(y)$. We refer to the distribution of each coefficient as X_{y+2} . Since only adjacent terms have dependencies and the support of $\mathbf{s}$ and $\mathbf{e}$ is small the full distribution X_{y+2} can still be computed efficiently. We validated the constructed distribution X_{y+2} by comparing against an empirically generated distribution with 10^{11} samples and verified that the mean and variance closely match. Since X_{y+2} has a very long tail distribution, even after 10^{11} samples the empirical distribution still leads to noticeable differences in the failure probabilities so theoretically computing the distribution is critical.

We note that not all coefficients have exactly the same distribution due to the negacyclic nature of $y^n + 1$ in the multiplication, so the other coefficients may a priori also be required. However, coefficients differ only in their mean and not in their variance. Since $\mathbf{es}'$ and $\mathbf{e}'\mathbf{s}$ have identical distributions, differences in means cancel out in the distribution for $\mathbf{es}' - \mathbf{e}'\mathbf{s}$. Therefore, while we compute the distribution on all of the coefficients, we use only the coefficient of y of $(y+2) \mathbf{e}_{m-i+k}(y) \mathbf{s}'_i(y)$ as a representative distribution for all its coefficients to simplify the treatment.

We compute the distribution X_{Δ_m} of Δ_m as $X_{\Delta_m} = (m+1)X_0 + (k-1-m)X_{y+2}$. Since all distributions are known at this point, this can be computed in full. We observe that the variance of X_{y+2} is significantly higher than X_0 so that in practice the failure probability is dominated by the minimum value of m . Since for Radical Ring-LWR we only use a subset of the rows for the shared secret, the minimum value of m is exactly $k - \ell$. Hence for Mithril-128 ($\ell = 1$) there is no effect from X_{y+2} , whereas for Mithril-256 ($\ell = 2$) and Mithril-512 ($\ell = 4$) we have at most 1 and 3 copies, respectively. The distribution Y_m on the m -th coefficient of $(\mathbf{es}')^{[k-\ell:k-1]} - (\mathbf{e}'\mathbf{s})^{[k-\ell:k-1]} + \mathbf{e}_r$ is the convolution of X_{Δ_m} and its negation together with the distribution on $\mathbf{e}_r$. The failure probability then follows as $\delta = 1 - p_Y$, where

$$p_Y = \prod_{m=k-\ell}^{k-1} (\Pr[|Y_m| \leq q/4])^n,$$

under the standard heuristic assumption that the distributions for Y_m are independent for different m .

Quantifying the impact of the covariance. To demonstrate that tracking the joint distribution is not a formality, we also evaluate the estimator with X_{y+2} replaced by the naïve law warned against above: the coefficient of $(y+2)\mathbf{e}_{m-i+k}(y)\mathbf{s}'_i(y)$ is modelled as $2A+B$ with A and B *independent*, each distributed as a single coefficient of $\mathbf{e}\mathbf{s}'$, that is, as a sum of n independent coefficient products. For Mithril-256 we observe that $\log_2(\delta_\ell)$ goes from -320 to -278 when taking covariance into account, while for Mithril-512 it goes from -674 to -569 , a noticeable understatement of 42 and 105 bits respectively for the naïve failure probability estimates.

4.3 Security of Octarine

The main result of this section is Proposition 3 that reduces the security of Octarine to the radical-ring instantiation of Ring-LWR (see Section 4.1 for its security argument) and a radical-ring variant of the self-target Shortest Integer Solution (SIS) that we refer to as SelfTarget-RR-SIS.

Definition 2 (SelfTarget-RR-SIS). Let k be an arbitrary positive integer, M be the message space, B_τ be the challenge space, and let $H : \mathbf{S}_{q,n,k} \times M \rightarrow B_\tau$ be a random oracle. For an adversary $\mathcal{A}$ against SelfTarget-RR-SIS we define

$$\text{Adv}_{H,k,\zeta}^{\text{SelfTarget-RR-SIS}}(\mathcal{A}) := \Pr \left[\|\mathbf{r}\|_\infty \leq \zeta, H([\mathbf{I} \mid \mathbf{A}_0 \mid \frac{q}{p}\mathbf{A}_1] \cdot \mathbf{r}, \mu) = \mathbf{r}[2] \right. \\ \left. \mid (\mathbf{A}_0, \mathbf{A}_1) \leftarrow \mathbf{S}_{q,n,k} \times \mathbf{S}_{p,n,k}; (\mathbf{r}, \mu) \leftarrow \mathcal{A}^H(\mathbf{A}_0, \mathbf{A}_1) \right].$$

The definition is close to the SelfTarget-MSIS from ML-DSA [LDK⁺22, Section 6.1] but there are notable differences. An LWE-based signature scheme such as ML-DSA samples $\mathbf{A}$ in $\mathbf{R}_q^{m \times k}$ since both $\mathbf{A}$ and the public key $\mathbf{t}$ are (indistinguishable from) uniformly randomly distributed over $\mathbf{R}_q$. However, in the case of LWR the public key $\mathbf{b}$ is distributed (indistinguishable from) uniformly randomly over $\mathbf{R}_p$, and hence we sample and lift it back to $\mathbf{R}_q$ via a multiplication with $\frac{q}{p}$. Moreover, the column-dimension of $\mathbf{t}$ in the context of Module-LWE (or Module-LWR) is n , leading to a column dimension of $\mathbf{I} \mid \mathbf{A}$ of $(2\ell+1)n$. In a ring setting the column dimension of $\mathbf{b}$ (corresponding to $\mathbf{A}_1$) increases to k with a total dimension of $\mathbf{I} \mid \mathbf{A}_0 \mid \mathbf{A}_1$ of $3kn$. We adjust for this increase in the dimension in the generation of our parameter sets, though the impact on security is minimal. So far all changes were due to switching from LWE to LWR. We also make the hardness assumption specific to radical rings by instantiating $\mathbf{R}_q$ by $\mathbf{S}_{q,n,k}$.

A standard argument is to reduce from the SelfTarget problem with bound ζ to the Module-SIS or Ring-SIS problem with bound 2ζ . This uses the forking lemma to generate two samples $\mathbf{y}, \mathbf{y}'$ satisfying $\|\mathbf{y}\|_\infty, \|\mathbf{y}'\|_\infty < \zeta$ such that $(\mathbf{I} \mid \mathbf{A}_0 \mid \mathbf{A}_1) \cdot (\mathbf{y} - \mathbf{y}') = 0$. As $\|\mathbf{y} - \mathbf{y}'\|_\infty < 2\zeta$ the reduction follows. Since this is based on the forking lemma it only works in the classical setting. Moreover, it is unclear if the increase in the bound size is strictly necessary or an artifact of the proof. To allow for direct comparison with ML-DSA we make analogous assumptions: parameter selection for Octarine assumes hardness equivalent to the SIS problem of column dimension $3kn$ with bound ζ .

Security proof. Having defined the hard problems, we continue with the security proof of Octarine. The main complexity lies in the reduction from Existential Forgery under Chosen Message Attack (EF-CMA) to Existential Forgery under No Message Attack (EF-NMA) for classical and quantum adversaries. This was originally based on a proof from Kiltz, Lyubashevsky and Schaffner [KLS18], though a gap was identified in [BBD⁺23] who subsequently fixed the gap and provided a formal proof for signature schemes based on Fiat-Shamir with Aborts.

<u>Octarine.KeyGen(par)</u> 1: $\rho \leftarrow \mathbb{B}^{64}$ 2: $\mathbf{A} \leftarrow D_{\mathbf{A}}(\mathbf{S}_{q,n,k}; \rho)$ 3: $\mathbf{s}_1 \leftarrow D_{\mathbf{s}}(\mathbf{S}_{q,n,k})$ 4: $\mathbf{t} = \mathbf{A}\mathbf{s}_1 \in \mathbf{S}_{q,n,k}$ 5: $\mathbf{b} = \lfloor \frac{p}{q}\mathbf{t} \rfloor \in \mathbf{S}_{p,n,k}$ 6: $\mathbf{s}_2 = \mathbf{t} - \frac{q}{p}\mathbf{b} \in \mathbf{S}_{q/p,n,k}$ 7: $(\mathbf{b}_1, \mathbf{b}_0) \leftarrow$ 8: $\text{Power2Round}_p(\mathbf{b}, 2^d)$ 9: $pk \leftarrow (\rho, \mathbf{b}_1, \mathbf{b}_0)$ 10: $sk \leftarrow (\rho, \mathbf{s}_1, \mathbf{s}_2, \mathbf{b}_0)$ 11: return (pk, sk)	<u>Octarine.Com(sk)</u> 12: $\mathbf{A} \leftarrow D_{\mathbf{A}}(\mathbf{S}_{q,n,k}; \rho)$ 13: $\mathbf{y} \leftarrow D_{\mathbf{y}}(\mathbf{S}_{q,n,k})$ 14: $\mathbf{w} = \mathbf{A} \cdot \mathbf{y}$ 15: $\mathbf{w}_1 = \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$ 16: return $(W = \mathbf{w}_1, St = (\mathbf{w}, \mathbf{y}))$ <u>Octarine.Resp((sk, W = \mathbf{w}_1, c, St = (\mathbf{w}, \mathbf{y})))</u> 17: $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$ 18: $\mathbf{r}_0 = \text{LowBits}_q(\mathbf{w} + c\mathbf{s}_2, 2\gamma_2)$ 19: if $\ \mathbf{r}_0\ _{\infty} \geq \gamma_2 - \beta$ or 20: $\ \mathbf{z}\ _{\infty} \geq \gamma_1 - \beta$ then $(\mathbf{z}, \mathbf{h}) \leftarrow \perp$ 21: else 22: $\mathbf{h} = \text{MakeHint}_q(-c\frac{q}{p}\mathbf{b}_0,$ 23: $\mathbf{w} + c\mathbf{s}_2 + c\frac{q}{p}\mathbf{b}_0, 2\gamma_2)$ 24: if $\ c\frac{q}{p}\mathbf{b}_0\ _{\infty} \geq \gamma_2$ then $(\mathbf{z}, \mathbf{h}) \leftarrow \perp$ 25: return $(\mathbf{z}, \mathbf{h})$ <u>Octarine.Verify(pk, W = \mathbf{w}_1, c, Z = (\mathbf{z}, \mathbf{h}))</u> 26: return $[\ \mathbf{z}\ _{\infty} < \gamma_1 - \beta] \text{ and } [\mathbf{w}_1 = \text{UseHint}_q(\mathbf{h}, \mathbf{A}\mathbf{z} - c\frac{q}{p}\mathbf{b}_12^d, 2\gamma_2)]$
--	---

Figure 4: The ID scheme ID_{oct} that is transformed into the signature scheme **Octarine** with the Fiat–Shamir with Aborts paradigm. The sampling details are omitted as they are not relevant to the proof. Similar to the treatment in [KLS18] we assume $\mathbf{b}_0$ to be part of the public key, though the real instantiations will include it only with the private key. The challenge is assumed to be sampled in B_{τ} and commitment recovery is not included (but accounted for in the proof).

The proof for **Octarine** proceeds as follows. First, we define the identification scheme ID_{oct} in Figure 4 that is transformed into **Octarine** using a standard Fiat–Shamir with Aborts transformation $\text{Octarine} = \text{FSwA}[\text{ID}_{\text{oct}}, H]$ with random oracle H . We show in Lemma 2 that ID_{oct} is **achVZK** (see [BBD⁺23, Definition 1]), which enables the use of [BBD⁺23, Theorem 2–3] to reduce from EF-CMA to EF-NMA. Finally, in Lemma 3 we reduce to **RR-LWR** and **SelfTarget-RR-SIS** by following a proof strategy close to [BBD⁺23, Lemma 7]. The full security claim is summarized in Proposition 3.

Lemma 2. *If $\beta \geq \max_{c \in B_{\tau}, \mathbf{s}_1 \in D_{\mathbf{s}_1}} \|\mathbf{c}\mathbf{s}_1\|_{\infty}$, then ID_{oct} is correct with error*

$$1 - \nu^{nk} \Pr[\|c\frac{q}{p}\mathbf{b}_0\|_{\infty} < \gamma_2], \text{ where } \nu = \frac{2(\gamma_1 - \beta) - 1}{2\gamma_1} \cdot \frac{2(\gamma_2 - \beta) - 1}{2\gamma_2},$$

and is perfectly achVZK.

Proof. The proof of the **achVZK** property closely follows the proof for No-Absort HVZK (**naHVZK**) of [KLS18, Lemma 4.3], since by virtue of rejection sampling and with the assumption that $\beta \geq \max_{c \in B_{\tau}, \mathbf{s}_1 \in D_{\mathbf{s}_1}} \|\mathbf{c}\mathbf{s}_1\|_{\infty}$, $\mathbf{z}$ has uniformly random coefficients in $\{-\gamma_1 + \beta + 1, \dots, \gamma_1 - \beta - 1\}$ if no abort takes place.

The simulator **ZKSim** generates $(c, \mathbf{z})$ randomly and then proceeds to generate $\mathbf{r}_0 = \text{LowBits}_q(\mathbf{A}\mathbf{z} - c\frac{q}{p}\mathbf{b}, 2\gamma_2)$, which is possible as $\mathbf{b}_0$ is included in the public key. It is

accurate because of the relationship

$$\mathbf{Az} - c\frac{q}{p}\mathbf{b} = \mathbf{A}(\mathbf{y} + c\mathbf{s}_1) - c\frac{q}{p}\mathbf{b} = \mathbf{Ay} + c(\mathbf{As}_1 - \frac{q}{p}\mathbf{b}) = \mathbf{w} + c\mathbf{s}_2,$$

which therefore has the exact same distribution as an honestly generated $\mathbf{r}_0$. All steps afterwards are generated honestly, including the generation of the hint vector, so the distribution of the generated proof matches perfectly with an honestly created one.

As we re-use the UseHint and MakeHint functions from Dilithium, the proof for correctness is also virtually identical to the proof in [KLS18, Lemma 4.4] so we do not repeat it here. In particular, UseHint does not fail as the norm $\|c\frac{q}{p}\mathbf{b}_0\|_\infty < \gamma_2$ is explicitly checked by the prover. As ZKSim models the output of the identification scheme exactly, the correctness error is exactly the probability that ZKSim outputs $\perp$. As is the case for Dilithium, the stated error is under the heuristic assumption that coefficients of $\mathbf{r}_0$ are indistinguishable from independent and uniformly distributed. The error is slightly adjusted as $\mathbf{y}$ and $\mathbf{r}_0$ are uniformly distributed in intervals of size $2\gamma_1$ and $2\gamma_2$ as opposed to $2\gamma_1 - 1$ and $2\gamma_2 - 1$ for Dilithium, respectively. $\square$

Lemma 3. (Derived from [BBD⁺23, Lemma 7]) For every adversary $\mathcal{A}^{\text{Sign}, H}$ breaking NMA security of FSwA[ID_{oct}, H], we can construct an RR-LWR adversary $\mathcal{B}$ and a SelfTarget-RR-SIS adversary $\mathcal{C}$ such that

$$\text{Adv}_{\text{FSwA}[ID_{\text{oct}}, H]}^{\text{EF-NMA}}(\mathcal{A}) \leq \text{Adv}_{k, D_s, 1}^{\text{RR-LWR}}(\mathcal{B}) + \text{Adv}_{(H \circ \text{shift}_{2\gamma_2}), k, \zeta}^{\text{SelfTarget-RR-SIS}}(\mathcal{C}),$$

for $\zeta = \max(\gamma_1 - \beta, 2\gamma_2 + 1 + \frac{q}{p}\tau 2^{d-1})$.

Proof. We also closely follow the proof from [BBD⁺23, Lemma 7], making adjustments as needed. The first step is to replace the public key $\mathbf{b}$ with a uniformly random one and return an undefined private key, and replace H by $H'(\mathbf{w}_1, m) := H(\text{shift}_{2\gamma_2}(\mathbf{w}_1), m)$. The **shift** is needed simply to create the integer representation where the high bits correspond to $\mathbf{w}_1$, which UseHint places in the least significant bits. The resulting game incurs a loss of $\text{Adv}_{k, D_s, 1}^{\text{RR-LWR}}(\mathcal{B})$.

We construct the adversary $\mathcal{C}$ that uses $\mathcal{A}$ to solve SelfTarget-RR-SIS. For an instance $(\mathbf{A}'_0, \mathbf{A}'_1) := (\mathbf{A}, -\mathbf{b}) \in \mathbf{S}_{q,n,k} \times \mathbf{S}_{p,n,k}$, $\mathcal{C}$ provides $\mathbf{b}$ to the EF-NMA adversary. This results in a response $(m, (c, \sigma = (\mathbf{z}, \mathbf{h})))$ from $\mathcal{A}$ which passes signature verification with respect to H' . The adversary $\mathcal{C}$ responds

$$\begin{aligned} \mathbf{r}' &:= (\mathbf{u}_2 - \mathbf{u}_1) \mid \mathbf{z} \mid c, \text{ where } \mathbf{r} = \mathbf{Az} - c\frac{q}{p}\mathbf{b}_1 2^d, \\ \mathbf{u}_1 &:= \mathbf{r} - \text{shift}_{2\gamma_2}(\text{UseHint}_q(\mathbf{h}, \mathbf{r}, 2\gamma_2)), \quad \mathbf{u}_2 := c\frac{q}{p}(\mathbf{b} - \mathbf{b}_1 2^d). \end{aligned}$$

We can check that

$$\begin{aligned} H([\mathbf{I} \mid \mathbf{A}'_0 \mid \frac{q}{p}\mathbf{A}'_1] \mathbf{r}', m) &= H([\mathbf{I} \mid \mathbf{A} \mid -\frac{q}{p}\mathbf{b}] \mathbf{r}', m) \\ &= H((\mathbf{u}_2 - \mathbf{u}_1) + \mathbf{Az} - c\frac{q}{p}\mathbf{b}, m) \\ &= H'(\text{UseHint}_q(\mathbf{h}, \mathbf{Az} - c\frac{q}{p}\mathbf{b}_1 2^d, 2\gamma_2), m), \end{aligned}$$

which equals $c = \mathbf{r}'[2]$ since $(m, (c, \sigma = (\mathbf{z}, \mathbf{h})))$ is a valid signature. Hence $\mathbf{r}'$ is a valid response for $\mathcal{C}$ if $\|\mathbf{r}'\|_\infty \leq \zeta$.

By definition of the signature we have $\|\mathbf{z}\|_\infty \leq \gamma_1 - \beta$ and $\|c\|_\infty = 1$. Finally, it follows from the definition of the hint vector that

$$\|\mathbf{u}_2 - \mathbf{u}_1\|_\infty \leq \|\mathbf{u}_2\|_\infty + \|\mathbf{u}_1\|_\infty \leq \frac{q}{p}\|c\|_1 \cdot \|\mathbf{b} - \mathbf{b}_1 2^d\| + 2\gamma_2 + 1 \leq \frac{q}{p}\tau 2^{d-1} + 2\gamma_2 + 1$$

and the proof is complete. $\square$

Proposition 3 (QROM EF-CMA Security of Octarine). *Let Γ be an event and let $\delta < 1$ be such that*

$$\Pr_{\mathbf{A} \leftarrow D_{\mathbf{A}}}[\neg \Gamma] \leq \delta.$$

Let $\epsilon, p_0 < 1$ be such that for all $\mathbf{A} \leftarrow D_{\mathbf{A}}$ satisfying Γ we have

$$\mathbb{E}_{\mathbf{A} \leftarrow D_{\mathbf{A}}} \left[\max_{\mathbf{w}} \Pr_{\mathbf{y} \leftarrow D_{\mathbf{y}}} [\text{HighBits}_q(\mathbf{A}\mathbf{y}, 2\gamma_2) = \mathbf{w}] \mid \Gamma \right] \leq \epsilon \quad (2)$$

and for $D_{\mathbf{z}}$ the uniform distribution with coefficients in $[-\gamma_1 + \beta + 1, \gamma_1 - \beta - 1]$

$$\Pr_{\mathbf{z} \leftarrow D_{\mathbf{z}}} [\|\text{LowBits}_q(\mathbf{A}\mathbf{z} - c \frac{q}{p} \mathbf{b}, 2\gamma_2)\|_{\infty} \geq \gamma_2 - \beta] \leq p_0, \text{ for all } c \in B_{\tau}, \quad (3)$$

$$\Pr_{c \leftarrow B_{\tau}} [\|c \frac{q}{p} \mathbf{b}_0\|_{\infty} \geq \gamma_2] \leq p_2,$$

for all $\mathbf{b} \leftarrow \mathbf{S}_{p,n,k}$. Then, for every EF-CMA adversary $\mathcal{A}^{\text{Sign}, H}$ against Octarine that makes q_S queries to the signing oracle Sign and q_H queries to the random oracle H , we can construct an RR-LWR adversary $\mathcal{B}$ and an adversary $\mathcal{C}$ against SelfTarget-RR-SIS such that

$$\text{Adv}_{\text{Octarine}}^{\text{EF-CMA}}(\mathcal{A}) \leq \text{Adv}_{k, D_{\mathbf{s}}, 1}^{\text{RR-LWR}}(\mathcal{B}) + \text{Adv}_{(H \circ \text{shift}_{2\gamma_2}), k, \zeta}^{\text{SelfTarget-RR-SIS}}(\mathcal{C}) + \delta + t$$

with approximately the same running time, where $\mathbf{p} = 1 - (1 - p_2)p_1^{kn}(1 - p_0)$ for $p_1 = \frac{2(\gamma_1 - \beta) - 1}{2\gamma_1}$ and $\zeta = \max(\gamma_1 - \beta, 2\gamma_2 + 1 + (q/p)\tau 2^{d-1})$, and for $q'_H = q_H + q_S$ we have where $t = t_Q$ if all adversaries and queries are quantum for

$$t_Q := \frac{2q_S\sqrt{\epsilon}}{(1 - \mathbf{p})} \sqrt{q'_H + 1 + \frac{q_S}{1 - \mathbf{p}}} + 2(q'_H + 1) \sqrt{\frac{q_S\epsilon}{(1 - \mathbf{p})}}$$

and $t = t_C$ if all adversaries and queries are classical for

$$t_C := \frac{2q_S(q'_H + 1)\epsilon}{(1 - \mathbf{p})} + \frac{q_S\epsilon(q_S + 1)}{2(1 - \mathbf{p})^2}.$$

Proof. The proof works by interpreting Octarine as a Fiat–Shamir with Aborts signature scheme $\text{FSwA}[\text{ID}_{\text{oct}}, H]$ based on ID_{oct} from Figure 4 and random oracle H .

We observe that Octarine deviates in a few ways from the standard transformation. Firstly, $\text{FSwA}[\text{ID}_{\text{oct}}, H]$ assumes that the full public key $\mathbf{b}$ is available, while Octarine only uses $\mathbf{b}_1$. A straightforward reduction (i.e., by discarding $\mathbf{b}_0$ from the public key), as done in [BBD⁺23, Section 6.3] for ML-DSA, shows that there is no loss of security (less information is available to the adversaries). Moreover, signatures for Octarine are of the form $(\tilde{c}, \mathbf{z}, \mathbf{h})$ whereas for $\text{FSwA}[\text{ID}_{\text{oct}}, H]$ they are of the form $(\mathbf{w}, \mathbf{z}, \mathbf{h})$. Each signature $(\mathbf{w}, \mathbf{z}, \mathbf{h})$ can be turned into $(\tilde{c}, \mathbf{z}, \mathbf{h})$ with a query to the random oracle per signing query to obtain $\tilde{c}$ from $\mathbf{w}$ and μ . Afterwards c is derived deterministically (in polynomial time) from $\tilde{c}$. Finally, we omit the hint rejection probabilities in signature generation and verification related to the Hamming weight in ID_{oct} , as the parameter ω can be selected so that the impact on $\mathbf{p}$ is negligible; the measurements reported in Section 3 confirm that hint-weight rejections are rare. Combining the above, we reduce the security of Octarine to the security of $\text{FSwA}[\text{ID}_{\text{oct}}, H]$ with q_S signing queries and q'_H random oracle queries.

By Lemma 2, ID_{oct} is perfectly achVZK so Theorem 2 and Theorem 3 from [BBD⁺23] apply with $\zeta_{zk} = 0$. From Equation (2) we obtain exactly the bound ϵ on the entropy of the commitment. As the coefficients of $\mathbf{y}$ are sampled in $[-\gamma_1, \gamma_1 - 1]$, an interval of size $2\gamma_1$, and $\mathbf{z}$ is accepted if and only if $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$ lies in the interval $\{-\gamma_1 + \beta + 1, \dots, \gamma_1 - \beta - 1\}$, the probability of success for $\mathbf{z}$ across its kn coefficients is at least p_1^{kn} . From Equation (3) it follows that the success probability of the second bound check $\Pr[\|\mathbf{r}_0\|_{\infty} < \gamma_2 - \beta] \geq 1 - p_0$,

hence the success probability of full Octarine is at least $(1 - p_2)p_1^{kn}(1 - p_0) =: 1 - \mathfrak{p}$, accounting also for the success probability of the check $\|c_p^q \mathbf{b}_0\|_\infty < \gamma_2$. Therefore, the abort probability is smaller or equal to $\mathfrak{p}$.

Now we can apply Theorem 2 and Theorem 3 from [BBD⁺23] to ID_{oct} to reduce to an EF-NMA adversary against $\text{FSwA}[\text{ID}_{\text{oct}}, H]$ with loss factor $\delta + t$, using t_Q from Theorem 3 for a quantum adversary and t_C from Theorem 2 for a classical adversary. Finally, we apply Lemma 3 to get the full reduction. $\square$

Finally it remains to show that the concrete bounds remain small when instantiating Proposition 3 with our parameter sets. The main challenge is to specify Γ to bound the min-entropy ϵ , which for ML-DSA is done in [BBD⁺23, Appendix A] by letting Γ be the event that a sub-matrix of $\mathbf{A}$ has a certain co-rank and computing the min-entropy ϵ accordingly, finding an optimal trade-off between δ and ϵ . The treatment from [BBD⁺23] specifically relies on the fact that $x^n + 1$ splits in $\mathbf{Z}_q$ for the ML-DSA prime q , which does not apply to our situation.

Instead, we show that the situation for Octarine simplifies due to the use of powers of two. An elementary argument shows that we have a chain of ideals

$$\mathbf{S}_{q,n,k} \supsetneq (\pi) \supsetneq (\pi^2) \supsetneq \dots \supsetneq (\pi^{\log_2(q)nk})$$

for $\pi = (x + 1)$ with $\pi^{\log_2(q)nk} = 0$, each inclusion decreasing the ideal size by a factor 2. We provide a proof in Lemma 4.

Lemma 4. *Let $q = 2^e$ and n be powers of two, let k be a power of two, and set $N = kn$, $L = eN$, $\pi = x + 1$ and $\mathcal{R} = \mathbf{Z}_q[x]/((x^k - 2)^n + 1)$. Then $\mathcal{R}$ is a chain ring with maximal ideal (π) , residue field $\mathbf{F}_2$ and $|(\pi^j)| = 2^{L-j}$. Consequently, for $\mathbf{A} \in \mathcal{R}$ with $J = \nu_\pi(\mathbf{A})$, i.e. J maximal such that $(x + 1)^J \mid \mathbf{A}$,*

$$|\ker(\mathbf{y} \mapsto \mathbf{A}\mathbf{y})| = 2^J, \quad |\text{im}(\mathbf{y} \mapsto \mathbf{A}\mathbf{y})| = 2^{L-J}, \quad \Pr_{\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R})}[J \geq J_0] = 2^{-J_0}.$$

Proof. Modulo 2 we have $f \equiv x^N + 1 = (x + 1)^N$ as N is a power of two, so $\mathcal{R}/2\mathcal{R} \cong \mathbf{F}_2[x]/(\pi^N)$ is local with residue field $\mathbf{F}_2$; as 2 is nilpotent, $\mathcal{R}$ is local with maximal ideal $\mathfrak{m} = (2, \pi)$. Comparing the two monic degree- N polynomials gives $\pi^N = f + 2h = 2h$ in $\mathcal{R}$ with $h(-1) = -f(-1)/2$ odd, since $f(-1) = ((-1)^k - 2)^n + 1$ equals 2 for k even and $3^n + 1 \equiv 2 \pmod{8}$ for $k = 1$. Hence h is a unit, $2 \in (\pi)$, $\mathfrak{m} = (\pi)$ and $\pi^L = 2^e h^e = 0$. Any ideal $I \not\subseteq (\pi^{j+1})$ contains some $z = \pi^j w$ with $w \notin \mathfrak{m}$ a unit, so $I = (\pi^j)$: the ideals form the chain (π^j) . Each quotient $(\pi^j)/(\pi^{j+1})$ is cyclic generated by π^j with annihilator $\mathfrak{m}$, hence of size 2; as $|\mathcal{R}| = 2^L$ the chain has L steps and $|(\pi^j)| = 2^{L-j}$.

Write $\mathbf{A} = \pi^J u$ with u a unit. Then $\ker(\mu_{\mathbf{A}}) = \text{Ann}(\pi^J) = (\pi^{L-J})$: the inclusion $\supseteq$ is clear, and if $\pi^i v$ with v a unit annihilates π^J then $\pi^{i+J} = 0$, so $i \geq L - J$. The three claims follow, the last from $J \geq J_0 \iff \mathbf{A} \in (\pi^{J_0})$. $\square$

As a result, there exists a maximal j such that $\mathbf{A} \in (\pi^j)$, i.e., $(x + 1)^j \mid \mathbf{A}$, which we define to be $\lambda(\mathbf{A})$, and $\mathbf{A}\mathbf{y} = 0$ if and only if $(x + 1)^{\log_2(q)nk-j} \mid \mathbf{y}$. Intuitively, the larger j , the worse the min-entropy for that specific matrix as the kernel of $\mathbf{A}$ is larger.

In this setting, we can define $\Gamma_J = \{\mathbf{A} : \lambda(\mathbf{A}) < J\}$ and it follows straightforwardly that $\Pr[-\Gamma_J] = 2^{-J}$. Moreover, as there are exactly $(q/(2\gamma_2))^{nk}$ total possible commitments, each of which has $(2\gamma_2)^{nk}$ pre-images under the map $\text{HighBits}_q(\mathbf{A}\mathbf{y}; 2\gamma_2)$ and at most $2^{\lambda(\mathbf{A})}$ pre-images from the multiplication with $\mathbf{A}$, it follows that

$$\Pr_{\mathbf{y} \leftarrow D_{\mathbf{y}}}[\text{HighBits}_q(\mathbf{A}\mathbf{y}, 2\gamma_2) = \mathbf{w}] \leq (2\gamma_2)^{kn} / (2\gamma_1)^{kn} 2^{\lambda(\mathbf{A})}$$

over the pre-image set $\mathbf{y}$ uniform in $[-\gamma_1, \gamma_1 - 1]$. To complete the statement from Equation (2) we average over $\mathbf{A}$ and Γ_J to get $E[2^{\lambda(\mathbf{A})} \mid \Gamma_J] = (\sum_{j=0}^{J-1} 2^j \cdot 2^{-j-1}) / (1 - 2^{-J}) =$

Table 3: Concrete security loss of Octarine ($\Gamma = \Gamma_{kn}$) and ML-DSA.

Algorithm	q_S	q_H	$\delta + t_C$	$\delta + t_Q$
ML-DSA-44	2^{64}	2^{128}	2^{-206}	2^{-58}
Octarine-128	2^{64}	2^{128}	2^{-821}	2^{-346}
ML-DSA-87	2^{64}	2^{192}	2^{-847}	2^{-360}
Octarine-256	2^{64}	2^{192}	2^{-1780}	2^{-793}

$J/(2(1 - 2^{-J}))$. Hence, $\epsilon(J) \leq (2\gamma_2)^{kn} J/(2(1 - 2^{-J}) \cdot (2\gamma_1)^{kn})$ and $\delta(J) = 2^{-J}$ as defined earlier. We summarize the concrete bounds and compare against ML-DSA in Table 3, showing that we achieve a *tight* reduction for Octarine.

We note that this argument applies directly to Octarine-128 and Octarine-256, where k is a power of two, which is the setting of Lemma 4. For Octarine-512 we have $k = 5$ and the valuation at $x + 1$ no longer controls the kernel on its own, since $\mathbf{S}_{q,n,k}$ is no longer local: modulo 2 we have $f \equiv (x^5 + 1)^{nk/5} = ((x + 1)(x^4 + x^3 + x^2 + x + 1))^{nk/5}$, and $x^4 + x^3 + x^2 + x + 1$ is irreducible over $\mathbf{F}_2$, so $\mathbf{S}_{q,n,k}$ splits as a product of two local rings with residue fields $\mathbf{F}_2$ and $\mathbf{F}_{2^4}$. The argument above then applies to each factor separately. Writing λ_1, λ_2 for the valuations at $x + 1$ and $x^4 + x^3 + x^2 + x + 1$, and $d_1 = 1, d_2 = 4$ for the corresponding residue degrees, one obtains $\Pr[\lambda_i(\mathbf{A}) \geq J_i] = 2^{-d_i J_i}$ and a kernel of size $2^{d_1 \lambda_1 + d_2 \lambda_2}$, so that $\Gamma_{J_1, J_2} = \{\mathbf{A} : \lambda_1(\mathbf{A}) < J_1, \lambda_2(\mathbf{A}) < J_2\}$ gives $\delta \leq 2^{-J_1} + 2^{-4J_2}$ and a bound on ϵ of the same shape. We omit the details.

5 Implementation and Evaluation

We demonstrate the practical feasibility and advantages of Mithril and Octarine. We select a high-performance target based on the x86 architecture and an Arm-based microprocessor for embedded implementations. Much of the high-level implementation design can be shared between the platforms, but we comment on specific implementation details to improve performance and reduce stack usage for the embedded implementation in Section 5.2.

Setup. All benchmarks for the x86 implementation were performed on an Intel Core i7-10700K processor running Ubuntu 22.04.5 LTS. The implementations were compiled with GCC 11.4.0 using `-O3`, with AVX2 enabled for vectorized implementations. Processor cycles were measured using the hardware performance monitoring unit (PMU). For each operation, we report the median of 1000 measurements following 10 warmup calls. The embedded implementation runs on an Arm Cortex-M4 microcontroller board (STM32F407G-DISC1) using the `pqm4` benchmarking framework [KPR⁺]. All measurements are obtained with the framework’s cycle-accurate harness, and all schemes are built with the identical toolchain and SHA-3/SHAKE backend.

5.1 Implementation Strategy

Arithmetic. A Toom-Cook approach is a natural candidate for the matrix-vector multiplication in Mithril given the small ring degree ($n = 128$) and power-of-two modulus. The lazy Toom-Cook interpolation [BMKV20] reduces the complexity and by combining Toom-4 and Karatsuba one splits each polynomial into 21 sub-polynomials of length 16. However, an NTT approach is also a natural fit by embedding the power-of-two arithmetic [CHK⁺21] into an auxiliary NTT-friendly prime field. To implement NTT for Mithril, we use a 27-bit auxiliary prime to ensure no overflow happens in the multiplication $\mathbf{A}s$ across all parameter sets. Besides improving performance, NTTs can be computed in-place and therefore also

provide benefits for total memory usage which is particularly important for embedded implementations.

Moving to Octarine, the larger ring degree ($n = 1024$) makes NTT-based multiplication even more attractive than Toom-Cook, reducing cycle counts and memory. To use NTTs for $\mathbf{A}\mathbf{y}$ at the 512-bit level, where the coefficients of $\mathbf{A}$ are 24-bit, the coefficients of $\mathbf{y}$ are 22-bit and the radical extension degree is $k = 5$, one needs a 58-bit NTT modulus to prevent any overflow. However, on processors with 32-bit multiplication instructions, such as Arm Cortex-M4 or AVX2 vector instructions for x86, a 58-bit NTT modulus requires multi-word arithmetic. To mitigate this we use a CRT approach with two NTT-friendly 29-bit primes to keep all the arithmetic in-place and within 32-bit bounds. For the 128-bit security level instead, two 27-bit primes are used, since the NTT need to cover only a single polynomial multiplication ($k = 1$). This introduces a structural trade-off: while computing the matrix-vector product over two fields requires extra memory for intermediate results and a final recombination step, it is more efficient than utilizing a single 58-bit modulus, which would affect performance on a 32-bit architecture due to the cost of multi-word arithmetic. Additionally, if one re-uses one of the 27-bit or 29-bit primes for Mithril, the arithmetic building blocks can be re-used to reduce the total code size. Since we report the code size per parameter set, we do not yet include this saving in the code size estimates below.

Both Mithril and Octarine allow several memory-speed implementation trade-offs: one can store ring elements and expanded seeds that are reused across attempts to avoid resampling and save cycles, at the cost of (often substantial) memory increase, or discard and regenerate them on each restart to preserve a minimal stack footprint. While performance is the main goal of the implementations we present, we do attempt to minimize the stack usage and code size for the embedded implementation even if there is a small performance impact.

Random oracles. To allow for a fair comparison with the state-of-the-art algorithms ML-KEM and ML-DSA, we instantiate our random oracles H_L with primitives based on SHA-3. We select SHAKE128 for public variables, specifically the matrix $\mathbf{A}$, and SHAKE256 for secret material. Due to its 512-bit capacity SHAKE256 is not sufficiently secure for Mithril-512 and Octarine-512 and performance evaluations would be misleading. Since there are no existing schemes with comparable security level, we omit them from the evaluation.

Secure implementation. The Mithril and Octarine algorithms do not involve secret-dependent branching and our implementations follow standard constant-time coding practices. Octarine signing follows the Fiat-Shamir-with-Abort procedure and thus requires a variable number of attempts, but this number does not depend on the secret. Constant-time routines do not prevent leakage through power consumption or electromagnetic emanations during Mithril decapsulation or Octarine signing, where the long-term secret keys are processed [RRCB20, XPR⁺22, RCDB24].

Masking countermeasures are crucial to resist differential power analysis [KJJ99] attacks. A detailed investigation of masking and other countermeasures is left for future work. Nevertheless, several features of Mithril and Octarine are favorable for masked implementations. For Mithril, deterministic LWR rounding avoids explicit error sampling, while power-of-two modular reduction, rounding, and Boolean-arithmetic mask conversions can be implemented efficiently using shifts and masks. Masking techniques developed for Saber implementations [BDK⁺21] can be adapted to Mithril. For Octarine, its power-of-two moduli simplify sampling, rounding, decomposition, hint computation and avoid much of the masking complexity introduced by arithmetic modulo an odd prime. Hence, we expect masking overhead to be lower than for ML-DSA, consistent with results showing substantial savings when ML-DSA’s prime modulus is replaced by a power of two [MGTF19].

Table 4: AVX2 performance rounded to the nearest thousand cycles (values in 10^3 cycles; Sec. in bits). Ratio rows give our scheme over the NIST baseline.

KEM: Mithril vs. ML-KEM					Signatures: Octarine vs. ML-DSA				
Algorithm	Sec.	KeyG.	Enc.	Dec.	Algorithm	Sec.	KeyG.	Sign	Vfy.
ML-KEM-512	140	26	27	29	ML-DSA-44	159	89	203	93
Mithril-128	153	36	38	42	Octarine-128	150	83	211	107
Ratio	1.09	1.39	1.38	1.43	Ratio	0.94	0.94	1.04	1.15
ML-KEM-1024	262	59	62	68	ML-DSA-87	285	237	428	234
Mithril-256	277	58	63	73	Octarine-256	288	153	384	193
Ratio	1.06	0.97	1.00	1.08	Ratio	1.01	0.65	0.90	0.82

5.2 Embedded Implementation

Slot-major scheduling. Since the Cortex-M4 has only 14 working registers (r0 through r12 and its link register), operation scheduling has significant impact on the performance. Naïvely, one would compute $\mathbf{b}_i = (\mathbf{A}\mathbf{s})_i = \sum_{j=0}^{k-1} (\mathbf{A}_{i,j}\mathbf{s}_j)$ by computing each term $\mathbf{A}_{i,j}\mathbf{s}_j$ and accumulating into a running result. We refer to this approach as *entry-major*, requiring $n = 128$ coefficient accumulators to be maintained for Mithril such that (approximately) $nk(k-1)$ times a coefficient accumulator is loaded into registers, the addition of a product coefficient is handled, and finally the accumulator is stored back into memory.

However, since we use NTTs (using $\hat{\mathbf{x}}$ to denote $\text{NTT}(\mathbf{x})$), our accumulation can take place in the NTT domain, for each coefficient index m we can compute $\hat{\mathbf{b}}_i[m] = (\hat{\mathbf{A}}\hat{\mathbf{s}})_i[m] = \sum_{j=0}^{k-1} \hat{\mathbf{A}}_{i,j}[m]\hat{\mathbf{s}}_j[m]$. We can load and store the accumulator coefficient only once, and handle the k 32-bit multiplications and additions in integer domain by reducing only after all accumulations have taken place. We refer to this as a *slot-major* implementation. This decreases the number of loads, stores and reductions for accumulation by a factor of $k-1$. Moreover, in integer domain the SMLAL (Signed Multiply Accumulate Long) can be used to do the multiplication and 64-bit accumulation with a single instruction. As a result of the operation sequencing, we are not operating on full polynomials (in NTT domain) at a time. Since instructions involving memory accesses (e.g., LDR/LDM, STR/STM) benefit from contiguous memory accesses, we adjust our data layout such that the k loads and stores for each variable for $\sum_{j=0}^{k-1} \hat{\mathbf{A}}_{i,j}[m]\hat{\mathbf{s}}_j[m]$ are on sequential memory addresses. Whereas for $\hat{\mathbf{s}}$ one would naturally store $\hat{\mathbf{s}}_i[m]$ at address $in + m$, we store it at address $mn + i$ instead, ensuring that $\hat{\mathbf{s}}_0[m], \dots, \hat{\mathbf{s}}_{k-1}[m]$ are contiguous in memory for all m . A similar layout is used for $\hat{\mathbf{A}}$. Considering the register pressure, up to 6 registers can be maintained for operands (3 for the matrix and 3 for the vector), so we perform $2\lceil k/3 \rceil$ LDMs on 3 registers and, if $3 \nmid k$, two further LDMs on $k \bmod 3$ registers per coefficient index m , for a total of $2\lceil k/3 \rceil$ LDMs. Applying all the above, each coefficient index m requires 1 accumulator init, $2\lceil k/3 \rceil$ LDMs, k SMLALs, 2 reductions, 1 subtraction, and 1 store. The final cost per result coefficient goes from the naïve $8k-3$ to $2\lceil k/3 \rceil + k + 5$ operations. The precise saving in cycle counts depends on the specific microarchitecture, but the saving is significant considering that there are nk coefficients in total (or nl for the shared secret computation).

Memory scheduling for Octarine. While we aim for a performance-optimized implementation, we aim to minimize the stack consumption where impact on performance is limited. A column-wise accumulation strategy is adopted to process the matrix-vector product. This allows us to store only one column of the matrix, and loop through the components of $\mathbf{y}$. In each iteration, the NTT-representation of the component $\mathbf{y}_j$ is computed, its contribution to the result $\mathbf{w}$ is added, and then the ring element representing the column is multiplied by $(y+2)$ in NTT-domain. By adopting this processing strategy, there is

Table 5: Cycle count and memory evaluation for the Cortex-M4 implementations comparing Mithril with ML-KEM, using rop-based estimates [APS15] for their security.

Algorithm	Sec.	Size (bytes)		Speed (clock cycles)			Hash
		Stack	Code	Keygen	Encaps	Decaps	
ML-KEM-512	140	5 428	15 748	374 093	371 731	406 241	72%
Mithril-128	153	10 932	9 232	365 219	442 241	543 036	49%
Ratio	1.09	2.01	0.59	0.98	1.19	1.34	0.68
ML-KEM-1024	262	7 500	16 792	969 986	982 488	1 042 121	74%
Mithril-256	277	18 748	9 528	658 121	804 765	1 010 923	45%
Ratio	1.06	2.50	0.57	0.68	0.82	0.97	0.62

no necessity to store the entire NTT of $\mathbf{y}$ in memory. Where the reuse constraints on $\mathbf{A}$ and $\mathbf{y}$ are relaxed, such as during KeyGen or Verify, the coefficients of $\mathbf{y}$ can be sampled on-the-fly and the NTT of $\mathbf{A}$ can be computed in-place.

5.3 Performance Evaluation

We compare our schemes against the optimized implementations of the NIST standards, namely ML-KEM and ML-DSA, at the nearest matching security levels. Table 4 shows the performance of Mithril and Octarine compared to the AVX2 optimized implementations of ML-KEM and ML-DSA¹. Table 5 and Table 6 compare Mithril and Octarine against the optimized implementations in pqm4², namely ML-KEM (`m4fspeed`) and ML-DSA (`m4f`).

At the 128-bit level Mithril-128 is generally slower than ML-KEM-512. It is important to note that this is not a like-for-like comparison: Mithril-128 provides higher bit security than ML-KEM-512 as its lattice dimension is increased from 512 to 640. The remaining performance gap is mainly due to the arithmetic being more costly (as the prime size has doubled) while the matrix is not yet sufficiently large for the radical ring to make up for the additional cost. At the 256-bit level the radical ring structure pays off as Mithril-256 *outperforms* ML-KEM-1024 on Cortex-M4 with key generation and encapsulation routines that take just $0.68\times$ and $0.82\times$ compared to ML-KEM, alongside almost identical decapsulation performance. For AVX2 the performance at the 256-bit security level of Mithril and ML-KEM is roughly similar with Mithril having slightly faster key generation but slower decapsulation.

Focusing on the memory size and hashing metrics, two effects stand out. First, Mithril is less bottlenecked by hashing: it spends 49% and 45% of its cycles in Keccak for 128-bit and 256-bit security levels respectively, versus 72% and 74% for ML-KEM, a direct consequence of the compact k -element public object and the power-of-two, rejection-free sampling. Second, the code footprint is about $0.59\times$ of ML-KEM-512 and $0.57\times$ of ML-KEM-1024. The price is a larger stack footprint, reflecting the intermediate radical-ring representations kept in memory. It should be noted that this implementation was not fully optimized for stack usage and we leave it as future work to incorporate optimizations as studied in [BRS22].

The advantages of the radical ring structure become much more apparent for Octarine. While ML-DSA-44 still has faster signing and verification for the AVX2 implementation, Octarine is faster than ML-DSA across all other parameter sets and implementations with a notable 31% reduction in the signature verification time compared to ML-DSA-87 on Cortex-M4. As with the KEM, the embedded implementation of Octarine is far less

¹<https://github.com/pq-crystals>, kyber commit 3edd5af, dilithium commit d35ba3f

²<https://github.com/mupq/pqm4>, commit 5e5cc76

Table 6: Cycle count and memory evaluation for the Cortex-M4 implementations comparing Octarine with ML-DSA, using rop-based estimates [APS15] for their security.

Algorithm	Sec.	Size (bytes)		Speed (clock cycles)			Hash
		Stack	Code	Keygen	Sign	Verify	
ML-DSA-44	159	44 816	19 456	1 359 300	3 679 342	1 428 878	72%
Octarine-128	150	24 224	23 698	939 652	2 867 008	1 189 104	31%
Ratio	0.94	0.54	1.22	0.69	0.78	0.83	0.43
ML-DSA-87	285	107 784	19 460	4 064 972	7 845 465	4 211 268	77%
Octarine-256	288	61 004	26 866	2 335 722	7 478 892	2 755 060	26%
Ratio	1.01	0.57	1.38	0.57	0.95	0.69	0.35

hashing-bound than ML-DSA: the mean hashing share drops to 31% and 26%, versus 72% and 77% for ML-DSA, i.e., a 2.3–3.0 $\times$ reduction. The stack footprint is roughly halved compared to ML-DSA (signing dominates the maximum). The main cost is code size, roughly 1.22 $\times$ and 1.38 $\times$ that of ML-DSA-44 and ML-DSA-87, respectively.

Acknowledgments. AI-based tools (including Claude Opus 4.8) were used for limited editorial assistance and for critically reviewing draft proofs. Suggestions provided by the AI were treated as informal feedback and were independently assessed by the authors. AI was used to assist in the generation of Python and C code for parameter and security estimates and for optimization of the implementations for AVX2 and Arm Cortex-M4. The authors take full responsibility for the correctness of all proofs, results, and conclusions.

References

- [ABD16] Martin R. Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 153–178. Springer, Berlin, Heidelberg, August 2016. doi:10.1007/978-3-662-53018-4_6.
- [AKPW13] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 57–74. Springer, Berlin, Heidelberg, August 2013. doi:10.1007/978-3-642-40041-4_4.
- [APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015. doi:10.1515/jmc-2015-0016.
- [BBD⁺23] Manuel Barbosa, Gilles Barthe, Christian Doczkal, Jelle Don, Serge Fehr, Benjamin Grégoire, Yu-Hsuan Huang, Andreas Hülsing, Yi Lee, and Xiaodi Wu. Fixing and mechanizing the security proof of Fiat-Shamir with aborts and Dilithium. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 358–389. Springer, Cham, August 2023. doi:10.1007/978-3-031-38554-4_12.
- [BBE⁺18] Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Benjamin Grégoire, Mélissa Rossi, and Mehdi Tibouchi. Masking the GLP lattice-based

- signature scheme at any order. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 354–384. Springer, Cham, April / May 2018. doi:[10.1007/978-3-319-78375-8_12](https://doi.org/10.1007/978-3-319-78375-8_12).
- [BBPS19] Madalina Bolboceanu, Zvika Brakerski, Renen Perlman, and Devika Sharma. Order-LWE and the hardness of ring-LWE with entropic secrets. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 91–120. Springer, Cham, December 2019. doi:[10.1007/978-3-030-34621-8_4](https://doi.org/10.1007/978-3-030-34621-8_4).
- [BCV20] Carl Bootland, Wouter Castryck, and Frederik Vercauteren. On the security of the multivariate ring learning with errors problem. In *Algorithmic Number Theory Symposium (ANTS-XIV)*, volume 4 of *Open Book Series*, pages 57–71. Mathematical Sciences Publishers, 2020. doi:[10.2140/obs.2020.4.57](https://doi.org/10.2140/obs.2020.4.57).
- [BDK⁺18] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367. IEEE Computer Society Press, April 2018. doi:[10.1109/EuroSP.2018.00032](https://doi.org/10.1109/EuroSP.2018.00032).
- [BDK⁺21] Michiel Van Beirendonck, Jan-Pieter D’anvers, Angshuman Karmakar, Josep Balasch, and Ingrid Verbauwhede. A side-channel-resistant implementation of SABER. *J. Emerg. Technol. Comput. Syst.*, 17(2), 2021. doi:[10.1145/3429983](https://doi.org/10.1145/3429983).
- [BGM⁺16] Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 209–224. Springer, Berlin, Heidelberg, January 2016. doi:[10.1007/978-3-662-49096-9_9](https://doi.org/10.1007/978-3-662-49096-9_9).
- [BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012. doi:[10.1145/2090236.2090262](https://doi.org/10.1145/2090236.2090262).
- [BHH⁺19] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 61–90. Springer, Cham, December 2019. doi:[10.1007/978-3-030-36033-7_3](https://doi.org/10.1007/978-3-030-36033-7_3).
- [BMKV20] Jose Maria Bermudo Mera, Angshuman Karmakar, and Ingrid Verbauwhede. Time-memory trade-off in toom-cook multiplication: an application to module-lattice based cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(2):222–244, March 2020. doi:[10.13154/tches.v2020.i2.222-244](https://doi.org/10.13154/tches.v2020.i2.222-244).
- [BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Berlin, Heidelberg, April 2012. doi:[10.1007/978-3-642-29011-4_42](https://doi.org/10.1007/978-3-642-29011-4_42).
- [BRS22] Joppe W. Bos, Joost Renes, and Amber Sprenkels. Dilithium for memory constrained devices. In Lejla Batina and Joan Daemen, editors,

- AFRICACRYPT 22*, volume 13503 of *LNCS*, pages 217–235. Springer, Cham, July 2022. doi:[10.1007/978-3-031-17433-9_10](https://doi.org/10.1007/978-3-031-17433-9_10).
- [BRVW26] Joppe W. Bos, Joost Renes, Frederik Vercauteren, and Peng Wang. Structured module lattice-based cryptography: Fine-grained parameter selection without compromising performance. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2026(3):1032–1055, July 2026. doi:[10.46586/tches.v2026.i3.1032-1055](https://doi.org/10.46586/tches.v2026.i3.1032-1055).
- [CGV14] Jean-Sébastien Coron, Johann Großschädl, and Praveen Kumar Vadnala. Secure conversion between Boolean and arithmetic masking of any order. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 188–205. Springer, Berlin, Heidelberg, September 2014. doi:[10.1007/978-3-662-44709-3_11](https://doi.org/10.1007/978-3-662-44709-3_11).
- [CHK⁺21] Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, and Bo-Yin Yang. NTT multiplication for NTT-unfriendly rings. *IACR TCHES*, 2021(2):159–188, 2021. URL: <https://tches.iacr.org/index.php/TCHES/article/view/8791>, doi:[10.46586/tches.v2021.i2.159-188](https://doi.org/10.46586/tches.v2021.i2.159-188).
- [CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *CRYPTO’99*, volume 1666 of *LNCS*, pages 398–412. Springer, Berlin, Heidelberg, August 1999. doi:[10.1007/3-540-48405-1_26](https://doi.org/10.1007/3-540-48405-1_26).
- [D’A21] Jan-Pieter D’Anvers. *Design and Security Analysis of Lattice-based Post-Quantum Encryption*. Phd thesis, KU Leuven, May 2021. URL: <https://lirias.kuleuven.be/3420486>.
- [DESS22] Xiaohui Ding, Muhammed F. Esgin, Amin Sakzad, and Ron Steinfeld. An injectivity analysis of Crystals-Kyber and implications on quantum security. In Khoa Nguyen, Guomin Yang, Fuchun Guo, and Willy Susilo, editors, *ACISP 22*, volume 13494 of *LNCS*, pages 332–351. Springer, Cham, November 2022. doi:[10.1007/978-3-031-22301-3_17](https://doi.org/10.1007/978-3-031-22301-3_17).
- [DKL⁺18] Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018(1):238–268, 2018. URL: <https://tches.iacr.org/index.php/TCHES/article/view/839>, doi:[10.13154/tches.v2018.i1.238-268](https://doi.org/10.13154/tches.v2018.i1.238-268).
- [DKR⁺20] Jan-Pieter D’Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel Van Beirendonck, and Andrea Basso. SABER. Technical report, National Institute of Standards and Technology, 2020. available at <https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions>.
- [DKRV18] Jan-Pieter D’Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 18*, volume 10831 of *LNCS*, pages 282–305. Springer, Cham, May 2018. doi:[10.1007/978-3-319-89339-6_16](https://doi.org/10.1007/978-3-319-89339-6_16).

- [HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 341–371. Springer, Cham, November 2017. doi:[10.1007/978-3-319-70500-2_12](https://doi.org/10.1007/978-3-319-70500-2_12).
- [Ins25] Institute of Commercial Cryptography Standards. Call for proposals for the next-generation public-key cryptographic algorithms. <https://www.niccs.org.cn/en/>, 2025. Part of the Next-generation Commercial Cryptographic Algorithms Program (NGCC), accessed September 2026.
- [ISO26] ISO/IEC. Information technology—security techniques—encryption algorithms—part 2: Asymmetric ciphers—amendment 2. ISO/IEC 18033-2:2006/Amd 2:2026, June 2026.
- [KF17] Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on over-stretched NTRU parameters. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 3–26. Springer, Cham, April / May 2017. doi:[10.1007/978-3-319-56620-7_1](https://doi.org/10.1007/978-3-319-56620-7_1).
- [KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999. doi:[10.1007/3-540-48405-1_25](https://doi.org/10.1007/3-540-48405-1_25).
- [KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 552–586. Springer, Cham, April / May 2018. doi:[10.1007/978-3-319-78372-7_18](https://doi.org/10.1007/978-3-319-78372-7_18).
- [KN25] Korean Information Security Agency and National Security Research Institute. Selected algorithms for the KpqC competition round 2. https://www.kpqc.or.kr/competition_02.html, 2025.
- [KPR⁺] Matthias J. Kannwischer, Richard Petri, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. PQM4: Post-quantum crypto library for the ARM Cortex-M4. <https://github.com/mupq/pqm4>.
- [KSS⁺20] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 703–728. Springer, Cham, May 2020. doi:[10.1007/978-3-030-45727-3_24](https://doi.org/10.1007/978-3-030-45727-3_24).
- [LDK⁺22] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>.
- [LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*,

- volume 6110 of *LNCS*, pages 1–23. Springer, Berlin, Heidelberg, May / June 2010. doi:10.1007/978-3-642-13190-5_1.
- [LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *DCC*, 75(3):565–599, 2015. doi:10.1007/s10623-014-9938-4.
- [Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Berlin, Heidelberg, December 2009. doi:10.1007/978-3-642-10366-7_35.
- [Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Berlin, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_43.
- [MGTF19] Vincent Migliore, Benoît Gérard, Mehdi Tibouchi, and Pierre-Alain Fouque. Masking dilithium - efficient implementation and side-channel evaluation. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *Applied Cryptography and Network Security - 17th International Conference, ACNS 2019, Bogota, Colombia, June 5-7, 2019, Proceedings*, Lecture Notes in Computer Science, pages 344–362. Springer, 2019. doi:10.1007/978-3-030-21568-2_17.
- [Nat16] National Institute of Standards and Technology. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, 2016. NIST PQC Project. URL: <https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization>.
- [Nat22] National Institute of Standards and Technology. Request for additional digital signature schemes for the post-quantum cryptography standardization process. NIST Post-Quantum Cryptography Project, September 2022. URL: <https://csrc.nist.gov/news/2022/request-additional-pqc-digital-signature-schemes>.
- [Nat24a] National Institute of Standards and Technology. Module-lattice-based digital signature standard. Federal Information Processing Standards Publication 204, August 2024. doi:10.6028/NIST.FIPS.204.
- [Nat24b] National Institute of Standards and Technology. Module-lattice-based key-encapsulation mechanism standard. Federal Information Processing Standards Publication 203, August 2024. doi:10.6028/NIST.FIPS.203.
- [Pei16] Chris Peikert. How (not) to instantiate ring-LWE. In Vassilis Zikas and Roberto De Prisco, editors, *SCN 16*, volume 9841 of *LNCS*, pages 411–430. Springer, Cham, August / September 2016. doi:10.1007/978-3-319-44618-9_22.
- [PP24] Chris Peikert and Zachary Pepin. Algebraically structured LWE, revisited. *Journal of Cryptology*, 37(3):28, July 2024. doi:10.1007/s00145-024-09508-3.
- [PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 142–159.

- Springer, Berlin, Heidelberg, May 2013. doi:10.1007/978-3-642-38348-9_9.
- [PUTPG⁺21] Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, Nicolas Gama, Mariya Georgieva, and Fernando Pérez-González. Revisiting multivariate ring learning with errors and its applications on lattice-based cryptography. *Mathematics*, 9(8), 2021. doi:10.3390/math9080858.
- [PUTPPG15] Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Multivariate lattices for encrypted image processing. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1707–1711, 2015. doi:10.1109/ICASSP.2015.7178262.
- [PUTPPG16] Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. On ring learning with errors over the tensor product of number fields, 2016. arXiv:1607.05244. URL: <https://arxiv.org/abs/1607.05244>.
- [PUTPPG17] Alberto Pedrouzo-Ulloa, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Multivariate cryptosystems for secure processing of multi-dimensional signals, 2017. arXiv:1712.00848. URL: <https://arxiv.org/abs/1712.00848>.
- [RCDB24] Prasanna Ravi, Anupam Chattopadhyay, Jan Pieter D’Anvers, and Anubhab Baksı. Side-channel and fault-injection attacks over lattice-based post-quantum schemes (Kyber, Dilithium): Survey and new results. *ACM Trans. Embed. Comput. Syst.*, 23(2), March 2024. doi:10.1145/3603170.
- [Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. doi:10.1145/1060590.1060603.
- [RRCB20] Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. Generic side-channel attacks on cca-secure lattice-based PKE and kems. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(3):307–335, 2020. doi:10.13154/TCHES.V2020.I3.307-335.
- [RSSS17] Miruna Rosca, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. Middle-product learning with errors. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 283–297. Springer, Cham, August 2017. doi:10.1007/978-3-319-63697-9_10.
- [RSW18] Miruna Rosca, Damien Stehlé, and Alexandre Wallet. On the ring-LWE and polynomial-LWE problems. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 146–173. Springer, Cham, April / May 2018. doi:10.1007/978-3-319-78381-9_6.
- [SAB⁺20] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at <https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions>.

- [SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, Berlin, Heidelberg, December 2009. doi:[10.1007/978-3-642-10366-7_36](https://doi.org/10.1007/978-3-642-10366-7_36).
- [XPR⁺22] Zhuang Xu, Owen Pemberton, Sujoy Sinha Roy, David Oswald, Wang Yao, and Zhiming Zheng. Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of Kyber. *IEEE Transactions on Computers*, 71(9):2163–2176, 2022. doi:[10.1109/TC.2021.3122997](https://doi.org/10.1109/TC.2021.3122997).