

MAMBA-Frost: A Lattice KEM from Learning With Quantization

Shanxiang Lyu, Ke Ma, and Junzuo Lai

College of Cybersecurity, Jinan University, Guangzhou, China

Email: lxx07@jnu.edu.cn, Make2024@stu2024.jnu.edu.cn, laijunzuo@gmail.com

Abstract—The Learning With Errors (LWE) problem provides a conservative and well-established security foundation for lattice-based cryptography, while Learning With Rounding (LWR) improves bandwidth efficiency through deterministic rounding. However, the rounding noise in LWR is inherently correlated with the hidden linear term, preventing tight and sample-preserving reductions to standard LWE for polynomial moduli.

Prior work on the Learning With Quantization (LWQ) problem established a tight security reduction from LWE, where standard additive noise is intrinsically replaced by quantization error. We specialize this framework by explicitly instantiating the quantization lattice as a scaled integer lattice. Under this geometric choice, specifically utilizing aligned power-of-two moduli, we derive the exact finite-support error distribution and construct an explicit bijection between transmitted split samples and normal-form LWE samples. These ingredients yield MAMBA-Frost, a plain LWQ-based key encapsulation mechanism that seamlessly pairs unstructured LWE security with quantization-based compression. Because its effective error stems entirely from public dithered quantization, MAMBA-Frost operates strictly with hardware-friendly power-of-two arithmetic. Furthermore, the construction employs an E_8 -coded message embedding to optimize the correctness margin without altering the underlying LWQ hardness argument.

We instantiate Frost at NIST security levels 1, 3, and 5 and provide a complete software implementation. Its power-of-two design realizes the core arithmetic with shifts and masks, keeping the implementation simple and amenable to constant-time engineering. At level 1, Frost-128 reduces the combined public-key and ciphertext size by about 40% compared with FrodoKEM-640 and by about 8% compared with SCloud⁺-128. On an AVX2 platform, Frost-128 achieves speedups of 2.3×, 1.8×, and 1.5× for key generation, encapsulation, and decapsulation over FrodoKEM-640, and 2.1×, 1.5×, and 1.3× over SCloud⁺-128.

Index Terms—Post-Quantum Cryptography, Lattice-Based Cryptography, Key Encapsulation Mechanism, Learning With Quantization

1. Introduction

Since Regev introduced the *Learning With Errors* problem [1], LWE has become one of the central assumptions in lattice based cryptography. Its strength comes from worst case hardness guarantees [2] and search to decision reductions [1], which give a clean and stable basis for security proofs. Algebraic variants such as Ring LWE and Module LWE [3] further improve efficiency and support leading post quantum standards, including Kyber (ML-KEM) [4] and Dilithium (ML-DSA) [5]. The most conservative line still uses plain, unstructured LWE, as in FrodoKEM [6], SCloud [7], and SCloud⁺ [8]. For deployments that prioritize long term confidentiality and minimal algebraic assumptions, this conservative design point remains important. These schemes use large unstructured matrices and explicit noise sampling, which bring substantial randomness, computation, and bandwidth costs.

Learning With Rounding (LWR) was introduced for constructing pseudorandom functions [9]. It replaces sampled error with deterministic rounding, removing explicit noise sampling and simplifying implementation. LWR has since appeared in a broad line of encryption and KEM schemes, including Saber [10], Round5 [11], TiGER [12], and Lizard [13]. Even Kyber, whose hardness assumption is MLWE, uses rounding based compression to reduce bandwidth.

The determinism that makes LWR attractive also governs its main theoretical difficulty. The rounding error is determined by the hidden linear value and is therefore input dependent. As a result, an LWR transcript follows an input-dependent rounding law outside the independently perturbed linear form used in standard LWE reductions. The initial LWR reduction requires super polynomial moduli [9]. Subsequent reductions improved the bounds for bounded sample counts [14]. For polynomial moduli, any reduction supporting unbounded samples incurs exponential loss [15]. Practical LWR based schemes therefore lack the same direct reduction route enjoyed by LWE. This gap suggests a structural design goal. A compression mechanism should keep the bandwidth advantage of rounding while yielding an explicit additive noise form whose induced error distribution is provably independent of the underlying linear input.

Learning With Quantization (LWQ) was recently introduced as a powerful cryptographic framework that unifies

noise addition and ciphertext compression into a single quantization operation [16]. Crucially, by leveraging public dithering, the induced quantization error becomes provably independent of the hidden linear input. This elegant decoupling yields an additive-noise representation that establishes a tight security reduction from standard LWE, even for polynomial moduli—resolving the fundamental security-efficiency trade-off that has long hindered LWR. However, despite this robust theoretical foundation, a significant gap remains between the macroscopic abstraction of general LWQ and the precise algebraic modeling required to build and evaluate a concrete cryptosystem. Consequently, no practical Public-Key Encryption (PKE) or Key Encapsulation Mechanism (KEM) has been constructed directly from the LWQ framework.

1.1. Our Contributions

We make the following contributions.

- **Formalization of LWQ- $\mathbb{Z}$ and its uniform-error LWE correspondence.** While [16] establishes the foundational statistical equivalence for general LWQ via macroscopic group-theoretic properties (the Dithering lemma), we formalize LWQ- $\mathbb{Z}$ for aligned power-of-two moduli with a fully self-contained algebraic proof. By explicitly uncovering the low-level algebraic mechanics (shift-invariance and modulo arithmetic), our approach simplifies verification and bridges the gap to exact low-level modeling. We advance beyond broad notions of “uniform noise” by deriving the exact finite-support discrete probability mass function (PMF) $\chi_{q,p}$. Furthermore, addressing the lack of explicit algebraic reductions in [16], we construct an exact, zero-loss public bijection between the split representation used for transmission and the LWQ normal form. This linear-time map, together with a public translation of the error interval, rigorously converts any distinguisher for either representation into a distinguisher for uniform-error LWE with the same sample count and advantage.
- **MAMBA-Frost construction and lattice codes for error correction.** Building on these concrete results, we construct MAMBA-Frost.PKE and MAMBA-Frost.KEM (hereafter referred to simply as Frost for brevity)¹ over unstructured matrices using split LWQ transcripts. Managing decryption failure rates (DFR) for bounded uniform noise is fundamentally harder than for Gaussian noise. To solve this, we introduce an E_8 -coded message embedding engineered specifically to optimize the correctness margin for our exact uniform noise profile, establishing whole-ciphertext failure bounds using precise Voronoi-facet Chernoff analysis. The PKE proof uses two normal-form LWE instances: one for the public key and one

1. Note on nomenclature: We denote our scheme as MAMBA-Frost, part of the broader MAMBA suite of LWQ-based cryptographic schemes. This should not be confused with the FROST threshold signature scheme (Flexible Round-Optimized Schnorr Threshold signatures).

TABLE 1. LEVEL-1 COMPARISON WITH REPRESENTATIVE LWE AND LWR BASED KEMS. ERR. INDEP. INDICATES WHETHER THE EFFECTIVE ERROR IS INDEPENDENT OF THE HIDDEN LINEAR TERM. BANDWIDTH IS IN BYTES. PERFORMANCE IS IN AVX2 KCYCLES.

Scheme	Type	Err. Indep.	pk	ct	KG	Enc	Dec
Frost-128	LWQ	Yes	5792	5856	483	690	807
Frodo-640 [6]	LWE	Yes	9616	9752	1123	1271	1193
SCloud ⁺ -128 [8]	LWE	Partial	7200	5456	999	1062	1014
R5N1-1 [11]	LWR	No	5740	5788	470	770	857
CCALizard-128 [13]	Hybrid	Partial	524320	955	41113	33	56

joint row-wise mixed-error instance for the ciphertext. Fujisaki-Okamoto with implicit rejection gives a CCA-secure KEM, with the stated loss depending on the PKE advantage, decryption-failure probability, and FO plaintext-space size.

- **Concrete parameterization and implementation.** Leveraging LWQ- $\mathbb{Z}$ along with our explicit PMF and precise mean/variance endpoints, we securely instantiate Frost at NIST security levels 1, 3, and 5 under rigorous Core-SVP estimates and whole-ciphertext decryption-failure bounds. This design yields substantial practical benefits: at NIST level 1, Frost-128 reduces the combined public-key and ciphertext size by approximately 40% compared to FrodoKEM-640, while achieving AVX2 speedups of $2.3\times$, $1.8\times$, and $1.5\times$ for key generation, encapsulation, and decapsulation, respectively. We report full bandwidth and KEM timings on the evaluated platform, with the level-1 comparison summarized in Table 1.

1.2. Related Work

Quantization in cryptography. The classical theory of dithered quantization [17] shows that suitable dithers produce input independent quantization noise. In lattice cryptography, related ideas appear in reconciliation mechanisms [18] and approximate homomorphic encryption, for example CKKS [19]. More recently, lattice quantizers have been used to improve bandwidth and failure tradeoffs in Kyber type constructions [20]. These works use quantization as an auxiliary tool inside an existing scheme. We use the quantize and reconstruct channel as the interface for both the hardness statement and the public key transcript.

Choice of quantizer. From a classical viewpoint, denser lattices such as A_n , D_4 , E_8 , and Barnes Wall lattices are attractive for quantization efficiency, and efficient nearest point algorithms are available for several families [21]. Recent cryptographic works have explored lattice coded PKE and Kyber type constructions based on dense lattices in order to exploit better coding gain or packing properties [22], [23], [24]. Our construction requires a public quantize and reconstruct channel that is coordinate wise, fits plain matrices over $\mathbb{Z}_q$, and supports a precise hardness route. Dense lattice decoding couples coordinates. Constant time implementations of these dense-lattice decoders require sublattice reduction and exhaustive coset search, with costs far above scalar

decoding over $\mathbb{Z}$ [24]. This motivates our use of LWQ- $\mathbb{Z}$ in Frost.

Hardness with nonstandard noise. Once the quantizer is fixed, security depends on the distribution of the induced quantization error. Hardness results for LWE with small parameters extend beyond the usual asymptotic noise regime [25]. Binary error LWE illustrates the need for dedicated analysis of highly structured errors [26]. A lossy-code approach establishes hardness for bounded-sample LWE with uniform interval errors [27]. These works show that one can move beyond textbook sampled noise families when the error distribution is controlled quantitatively. This viewpoint is directly relevant to our LWQ instantiation, whose error is induced by the public quantization channel.

Entropy based hardness results reinforce the same perspective. Boundedness and entropy govern hardness guarantees for LWE and Ring LWE with general entropic distributions [28], [29]. This line of analysis also extends to Module LWE [30], [31], [32], [33]. These results give the conceptual language for nonstandard error distributions. We use the same language of boundedness, min entropy, and effective error scale to analyze the explicit quantization error distribution induced by our channel.

Concrete cryptanalysis. Recent works have refined the modelling of lattice reduction and primal attacks against MLWE instances [34], [35], provided systematic benchmarking of LWE attacks [36], and strengthened the provable foundation of dual attacks [37], [38]. These developments support a concrete analysis of LWQ through its explicit quantization error distribution.

2. Preliminaries

2.1. Notation and Conventions

Arithmetic and Rounding. For $x \in \mathbb{R}$, $\lfloor x \rfloor \triangleq \lfloor x + 1/2 \rfloor$ denotes nearest-integer rounding with ties broken upward. For an integer $q \geq 1$, $\mathbb{Z}_q$ denotes the ring of integers modulo q , canonically represented by $\{0, \dots, q-1\}$. For $x \in \mathbb{Z}$, $\lfloor x \rfloor_q \in \mathbb{Z}_q$ and $\lfloor x \rfloor_q^\pm \in \{-\lfloor q/2 \rfloor, \dots, \lfloor q/2 \rfloor - 1\}$ denote its canonical and centered reductions, respectively. For integers $p < q$, the modular rounding map $\lfloor \cdot \rfloor_{q \rightarrow p} : \mathbb{Z}_q \rightarrow \mathbb{Z}_p$ is defined as $\lfloor x \rfloor_{q \rightarrow p} \triangleq \lfloor (p/q) \cdot x \rfloor \bmod p$, where x is evaluated via its canonical integer representative. All operations extend entrywise to vectors and matrices.

Linear Algebra and Distributions. We use bold upper-case letters (e.g., $\mathbf{A}$) for matrices and bold lower-case letters (e.g., $\mathbf{v}$) for column vectors. For a vector $\mathbf{v}$, $\|\mathbf{v}\|_2$ and $\|\mathbf{v}\|_\infty$ denote its Euclidean and infinity norms, respectively. For a finite set S , $\mathcal{U}(S)$ denotes the uniform distribution over S . For a distribution χ , $\text{supp}(\chi)$ denotes its support, and χ^k denotes the k -fold independent product distribution. Sampling of vectors/matrices (written as $\mathbf{s} \leftarrow \chi$ or $\mathbf{A} \leftarrow \mathcal{U}(S)$) is performed independently entrywise. We write $\Pr[\cdot]$ for probability and $\mathbb{E}[\cdot]$ for expectation. For a discrete random variable X , its min-entropy in bits is $H_\infty(X) \triangleq -\log_2 \max_x \Pr[X = x]$.

Asymptotics and Lattice Parameters. We adopt standard asymptotic notation ($O, \Omega, \Theta, o, \omega$). An algorithm is

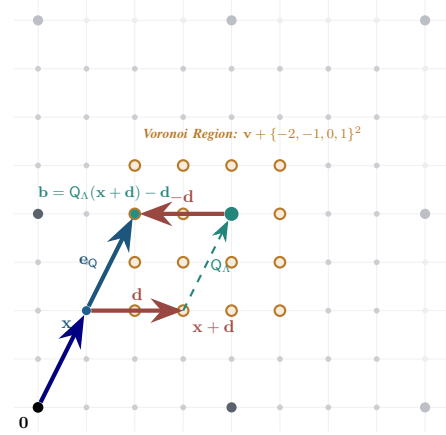


Figure 1. Geometric illustration of dithered quantization under $\Lambda = 4\mathbb{Z}^2$ with $\mathbf{d} = (2, 0)$. The shaded 4×4 region represents the Voronoi cell $\mathbf{v} + \{-2, -1, 0, 1\}^2$ anchored at $\mathbf{v} = (4, 4)$ under nearest-integer rounding with ties rounded upward. Vectors show the mapping from $\mathbf{x}$ to $\mathbf{x} + \mathbf{d}$, its quantization to $\mathbf{v}$, and the final compensated error $\mathbf{e}_Q$. Points at $x = 8$ or $y = 8$ are faded to indicate mod 8 folding after storage.

probabilistic polynomial-time (PPT) if its runtime is bounded by $\text{poly}(\lambda)$, where λ is the security parameter. A function $\text{negl}(\lambda)$ is negligible if it decays faster than any inverse polynomial.

Centered binomial distribution. For an integer $\eta \geq 1$, the centered binomial distribution $\mathcal{B}_\eta$ over $\mathbb{Z}$ is defined as the distribution of

$$X = \sum_{i=1}^{\eta} (a_i - b_i), \quad (1)$$

where $a_i, b_i \leftarrow \mathcal{U}(\{0, 1\})$ are sampled independently. Thus $\text{supp}(\mathcal{B}_\eta) = \{-\eta, \dots, \eta\}$, $\mathbb{E}[X] = 0$, and $\text{Var}(X) = \eta/2$. For vectors and matrices, $\mathcal{B}_\eta^N$ and $\mathcal{B}_\eta^{r \times c}$ denote entrywise independent sampling from $\mathcal{B}_\eta$.

2.2. Lattices and Hardness Problems

An n -dimensional lattice $\Lambda \subset \mathbb{R}^n$ is a discrete additive subgroup. When a full-rank basis matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ is specified, we write $\mathcal{L}(\mathbf{B}) \triangleq \{\mathbf{B}\mathbf{z} : \mathbf{z} \in \mathbb{Z}^n\}$ for the lattice generated by $\mathbf{B}$. The nearest-neighbor lattice quantizer is

$$Q_\Lambda(\mathbf{x}) \triangleq \arg \min_{\boldsymbol{\lambda} \in \Lambda} \|\mathbf{x} - \boldsymbol{\lambda}\|, \quad (2)$$

with ties broken by a fixed translation-consistent rule. Given a dither vector $\mathbf{d}$, the dithered quantizer is $Q_{\Lambda, \mathbf{d}}(\mathbf{x}) \triangleq Q_\Lambda(\mathbf{x} + \mathbf{d})$, yielding the compensated error

$$\mathbf{e}_Q = Q_\Lambda(\mathbf{x} + \mathbf{d}) - (\mathbf{x} + \mathbf{d}). \quad (3)$$

Fig. 1 visualizes this process for a concrete two-dimensional setting with $\Lambda = 4\mathbb{Z}^2$ and dither $\mathbf{d} = (2, 0)$, where the lattice points, the shifted input, and the resulting quantization error are shown explicitly.

For any distribution $\mathcal{D}$ over a set Ω and a PPT distinguisher $\mathcal{A}$, we compactly denote its distinguishing advantage against the uniform distribution $\mathcal{U}(\Omega)$ as:

$$\text{Adv}_{\mathcal{D}}^{\text{dist}}(\mathcal{A}) \triangleq \left| \Pr_{\mathbf{x} \leftarrow \mathcal{D}}[\mathcal{A}(\mathbf{x}) = 1] - \Pr_{\mathbf{x} \leftarrow \mathcal{U}(\Omega)}[\mathcal{A}(\mathbf{x}) = 1] \right|. \quad (4)$$

Definition 1 (LWE, [1]). Let $n, m, q \in \mathbb{N}$, and χ_s, χ_e be distributions over $\mathbb{Z}_q$. The LWE distribution $\text{LWE}_{q, \chi_s, \chi_e}^m$ outputs $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, where $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$, $\mathbf{s} \leftarrow \chi_s^n$, and $\mathbf{e} \leftarrow \chi_e^m$.

- **Search-LWE:** Given $(\mathbf{A}, \mathbf{b}) \leftarrow \text{LWE}_{q, \chi_s, \chi_e}^m$, find $\mathbf{s}$.
- **Decision-LWE:** Distinguish $\text{LWE}_{q, \chi_s, \chi_e}^m$ from $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$. We denote this concrete advantage by $\text{Adv}_{q, n, m, \chi_s, \chi_e}^{\text{LWE}}(\mathcal{A})$.

The following LWQ and split LWQ definitions specialize the formulations in [16] to an integer product lattice, using the convention of adding the dither before quantization and subtracting it afterward.

Definition 2 (LWQ, [16]). Let $n, m, q, p \in \mathbb{N}$ with $p \mid q$, $\Delta \triangleq q/p$, $\Lambda \triangleq \Delta\mathbb{Z}^m$, and $\mathcal{D}_\Delta \triangleq \{0, \dots, \Delta - 1\}$. The LWQ distribution $\text{LWQ}_{q, p, \chi_s}^m$ outputs $(\mathbf{A}, \mathbf{b} = \mathbf{Q}_{\Lambda, \Delta}(\mathbf{A}\mathbf{s}) - \mathbf{d})$, where $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$, $\mathbf{s} \leftarrow \chi_s^n$, and $\mathbf{d} \leftarrow \mathcal{U}(\mathcal{D}_\Delta^m)$.

- **Search-LWQ:** Given $(\mathbf{A}, \mathbf{b}) \leftarrow \text{LWQ}_{q, p, \chi_s}^m$, find $\mathbf{s}$.
- **Decision-LWQ:** Distinguish $\text{LWQ}_{q, p, \chi_s}^m$ from $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$. We denote this concrete advantage by $\text{Adv}_{q, n, m, p, \chi_s}^{\text{LWQ}}(\mathcal{A})$.

Definition 3 (Split-LWQ, [16]). Let parameters be as in Definition 2. The Split-LWQ distribution $\text{SLWQ}_{q, p, \chi_s}^m$ outputs $(\mathbf{A}, \mathbf{c} = \mathbf{Q}_{\Lambda, \Delta}(\mathbf{A}\mathbf{s}), \mathbf{d})$, where $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$, $\mathbf{s} \leftarrow \chi_s^n$, and $\mathbf{d} \leftarrow \mathcal{U}(\mathcal{D}_\Delta^m)$.

- **Search Split-LWQ:** Given $(\mathbf{A}, \mathbf{c}, \mathbf{d}) \leftarrow \text{SLWQ}_{q, p, \chi_s}^m$, find $\mathbf{s}$.
- **Decision Split-LWQ:** Distinguish $\text{SLWQ}_{q, p, \chi_s}^m$ from $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times (\Lambda \cap \mathbb{Z}_q^m) \times \mathcal{D}_\Delta^m)$, where $\Lambda \cap \mathbb{Z}_q^m = \{0, \Delta, \dots, q - \Delta\}^m$. We denote this concrete advantage by $\text{Adv}_{q, n, m, p, \chi_s}^{\text{Split-LWQ}}(\mathcal{A})$.

2.3. Cryptographic Definitions

A public-key encryption scheme PKE over message space $\mathcal{M}$ consists of PPT algorithms (KeyGen, Enc, Dec) mapping to keys (pk, sk) , ciphertexts ct , and decrypted outputs $m' \in \mathcal{M} \cup \{\perp\}$.

- **δ -Correctness:** PKE is δ -correct if $\Pr[\text{Dec}(\text{sk}, \text{Enc}(\text{pk}, m)) \neq m] \leq \delta$ for all $m \in \mathcal{M}$, over the internal coins of KeyGen and Enc.
- **IND-CPA Security:** For a PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, its IND-CPA advantage is defined via the experiment where $(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(1^\lambda)$, $(m_0, m_1) \leftarrow \mathcal{A}_1(\text{pk})$, and $\text{ct}^* \leftarrow \text{Enc}(\text{pk}, m_b)$ for $b \leftarrow \mathcal{U}(\{0, 1\})$:

$$\text{Adv}_{\text{PKE}}^{\text{IND-CPA}}(\mathcal{A}) \triangleq \left| \Pr[\mathcal{A}_2(\text{pk}, \text{ct}^*) = b] - \frac{1}{2} \right|. \quad (5)$$

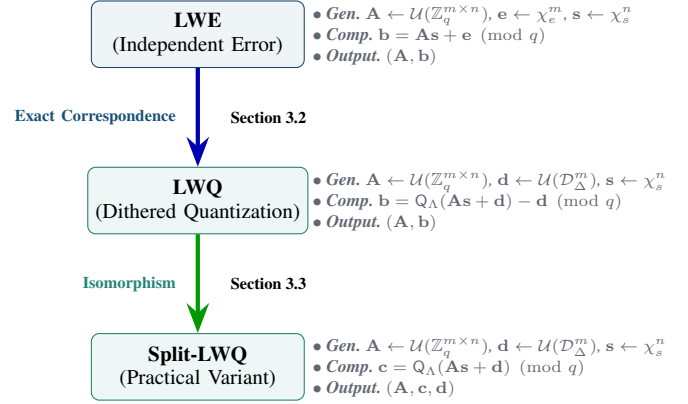


Figure 2. The hardness reduction chain of LWQ.

A key encapsulation mechanism KEM over key space $\mathcal{K}$ consists of PPT algorithms (KeyGen, Encaps, Decaps) mapping to keys (pk, sk) , ciphertexts ct , and capsule keys $K \in \mathcal{K}$.

- **IND-CCA Security:** Let $\mathcal{A}$ be a PPT adversary with access to a decapsulation oracle $\mathcal{O}_{\text{dec}}(\cdot)$ (which returns $\text{Decaps}(\text{sk}, \cdot)$ for any $\text{ct} \neq \text{ct}^*$). Given target challenge $(\text{pk}, \text{ct}^*, K_b)$, where $(\text{ct}^*, K_0) \leftarrow \text{Encaps}(\text{pk})$, $K_1 \leftarrow \mathcal{U}(\mathcal{K})$, and $b \leftarrow \mathcal{U}(\{0, 1\})$, the IND-CCA advantage is:

$$\text{Adv}_{\text{KEM}}^{\text{IND-CCA}}(\mathcal{A}) \triangleq \left| \Pr[\mathcal{A}^{\mathcal{O}_{\text{dec}}}(\text{pk}, \text{ct}^*, K_b) = b] - \frac{1}{2} \right|. \quad (6)$$

3. On the Hardness of LWQ- $\mathbb{Z}$

In this work, we specialize the LWQ quantization lattice to the standard integer lattice, yielding a concrete instantiation that we denote as LWQ- $\mathbb{Z}$. Let $q = 2^{t_1}$ and $p = 2^{t_2}$ with $0 < t_2 < t_1$. Set the quantization step to $\Delta \triangleq q/p = 2^{t_1 - t_2}$, so the scaled quantization lattice is $\Lambda \triangleq \Delta\mathbb{Z}^m$. Throughout this section, the dither domain for both $\text{LWQ}_{q, p, \chi_s}^m$ and $\text{SLWQ}_{q, p, \chi_s}^m$ is $\mathcal{D}_\Delta^m$, where $\mathcal{D}_\Delta \triangleq \{0, \dots, \Delta - 1\}$. All quantization inputs are represented by their canonical integer representatives in $\{0, \dots, q - 1\}^m$. The sum $\mathbf{x} + \mathbf{d}$ is quantized over $\mathbb{Z}^m$ and is reduced modulo q only when interpreted as a cryptographic sample.

Figure 2 summarizes the proof route. The divisibility condition $p \mid q$ aligns the quantization lattice $\Delta\mathbb{Z}^m$ with the canonical representatives of $\mathbb{Z}_q^m$. Uniform dithering gives input independence, while divisibility makes the quantizer compatible with reduction modulo q . A public dither sampled uniformly from $\mathcal{D}_\Delta^m$ decouples the quantization error from the linear input and identifies the normal form of LWQ- $\mathbb{Z}$ with an LWE normal-form distribution. It also gives unique decomposition. Every normal form coordinate in $\mathbb{Z}_q$ has a unique deterministic decomposition into a quantized coordinate and a dither coordinate, so the split and normal forms are two public encodings of the same sample.

3.1. The Quantization Error Distribution

We first compute the scalar error induced by nearest neighbor quantization onto $\Delta\mathbb{Z}$ with uniform dither. This distribution is the error distribution in the corresponding LWE instance.

Definition 4. Let $q = 2^{t_1}$, $p = 2^{t_2}$, and $0 < t_2 < t_1$. Let $\Delta \triangleq q/p$. For $u \leftarrow \mathcal{U}(\mathcal{D}_\Delta)$, define $e_{q,p}(u) \triangleq Q_{\Delta\mathbb{Z}}(u) - u$. The distribution of $e_{q,p}(u)$ is denoted by $\chi_{q,p}$.

Proposition 1. Let $q = 2^{t_1}$, $p = 2^{t_2}$, and $0 < t_2 < t_1$. Let $\Delta \triangleq q/p$. Then

$$e_{q,p}(u) = \begin{cases} -u, & 0 \leq u < \Delta/2, \\ \Delta - u, & \Delta/2 \leq u < \Delta. \end{cases} \quad (7)$$

Consequently, $\chi_{q,p}$ is uniform over $\{-\Delta/2 + 1, \dots, \Delta/2\}$. Moreover, $\mathbb{E}[e] = 1/2$, $\text{Var}(e) = (\Delta^2 - 1)/12$, and $\mathbb{E}[e^2] = (\Delta^2 + 2)/12$. Every possible error value satisfies $|e| \leq \Delta/2$.

Proof. Nearest neighbor quantization onto $\Delta\mathbb{Z}$ is $Q_{\Delta\mathbb{Z}}(z) = \Delta \lfloor z/\Delta \rfloor$. For $u \in \{0, \dots, \Delta - 1\}$, this gives $e_{q,p}(u) = \Delta \lfloor u/\Delta \rfloor - u$.

Write $\Delta = 2h$. Under the convention $\lfloor x \rfloor = \lfloor x + 1/2 \rfloor$, we have $\lfloor u/\Delta \rfloor = 0$ for $0 \leq u < h$ and $\lfloor u/\Delta \rfloor = 1$ for $h \leq u < 2h$. Hence $e_{q,p}(u) = -u$ for $0 \leq u < h$, while $e_{q,p}(u) = \Delta - u$ for $h \leq u < \Delta$. The first range gives $0, -1, \dots, -h + 1$. The second range gives $h, h - 1, \dots, 1$. Every value in $\{-h + 1, \dots, h\}$ occurs exactly once. Thus $\chi_{q,p}$ is uniform over $\{-\Delta/2 + 1, \dots, \Delta/2\}$.

This support has center $1/2$ and length Δ . It follows that $\mathbb{E}[e] = 1/2$ and $\text{Var}(e) = (\Delta^2 - 1)/12$. The identity $\mathbb{E}[e^2] = \text{Var}(e) + (\mathbb{E}[e])^2 = (\Delta^2 + 2)/12$ follows immediately. The bound $|e| \leq \Delta/2$ follows from the endpoints of the support. $\square$

We lift $\chi_{q,p}$ coordinate wise to vectors in $\mathbb{Z}^m$. Thus $\mathbf{e} \leftarrow \chi_{q,p}^m$ denotes an error vector with independent coordinates sampled from $\chi_{q,p}$.

Lemma 1. Let $\mathbf{e} \leftarrow \chi_{q,p}^m$. Then $H_\infty(\mathbf{e}) = m \log_2 \Delta$, $\|\mathbf{e}\|_\infty \leq \Delta/2$, and $\|\mathbf{e}\|_2 \leq (\Delta/2)\sqrt{m}$. For any two possible error vectors $\mathbf{e}, \mathbf{e}'$, one has $\|\mathbf{e} - \mathbf{e}'\|_\infty \leq \Delta$ and $\|\mathbf{e} - \mathbf{e}'\|_2 \leq \Delta\sqrt{m}$. Moreover, $\mathbb{E}[\|\mathbf{e}\|_2^2] = m(\Delta^2 + 2)/12$, while the centered variance sum is $m(\Delta^2 - 1)/12$.

Proof. By Proposition 1, each coordinate of $\mathbf{e}$ is uniform over $\{-\Delta/2 + 1, \dots, \Delta/2\}$. There are Δ possible values per coordinate and m coordinates. This gives $H_\infty(\mathbf{e}) = m \log_2 \Delta$. The norm bounds follow from $|e_i| \leq \Delta/2$. The pairwise distance bounds follow by applying the same coordinate wise bound to $e_i - e'_i$. The moment and variance identities follow from Proposition 1 and summation over all coordinates. $\square$

3.2. LWE Correspondence

The next theorem specializes the input-independence principle of dithered LWQ [16] to $\Delta\mathbb{Z}^m$. It identifies the full vector error as an independent product of the exact scalar

law $\chi_{q,p}$. The proof uses uniform residues modulo Δ , with $\Delta \mid q$ ensuring compatibility with the sample space $\mathbb{Z}_q^m$.

Theorem 1. Let $q = 2^{t_1}$, $p = 2^{t_2}$, and $0 < t_2 < t_1$. Let $\Delta \triangleq q/p$, and let $\Lambda \triangleq \Delta\mathbb{Z}^m$. Let $\mathbf{x} \in \mathbb{Z}_q^m$ be arbitrary and represented by its canonical integer representative. Sample $\mathbf{d} \leftarrow \mathcal{U}(\mathcal{D}_\Delta^m)$, and quantize $\mathbf{x} + \mathbf{d}$ before reducing modulo q . For $E_\Lambda(\mathbf{z}) \triangleq Q_\Lambda(\mathbf{z}) - \mathbf{z}$, the distribution of $E_\Lambda(\mathbf{x} + \mathbf{d})$, over the randomness of $\mathbf{d}$, is independent of $\mathbf{x}$. The coordinates of $E_\Lambda(\mathbf{x} + \mathbf{d})$ are independent, and each coordinate follows $\chi_{q,p}$.

Proof. Since $\Lambda = \Delta\mathbb{Z}^m$, nearest neighbor quantization is invariant under shifts by $\Delta\mathbb{Z}^m$. Hence $E_\Lambda(\mathbf{z} + \Delta\mathbf{a}) = E_\Lambda(\mathbf{z})$ for every $\mathbf{a} \in \mathbb{Z}^m$. Thus $E_\Lambda(\mathbf{z})$ depends only on $\mathbf{z} \bmod \Delta$. Since $\mathbf{d} \leftarrow \mathcal{U}(\mathcal{D}_\Delta^m)$, the residue $\mathbf{x} + \mathbf{d} \bmod \Delta$ is uniform over $\mathcal{D}_\Delta^m$, independently of $\mathbf{x}$. Therefore, for every $\mathbf{t} \in \mathbb{Z}^m$,

$$\Pr[E_\Lambda(\mathbf{x} + \mathbf{d}) = \mathbf{t}] = \Pr_{\mathbf{r} \leftarrow \mathcal{U}(\mathcal{D}_\Delta^m)}[E_\Lambda(\mathbf{r}) = \mathbf{t}]. \quad (8)$$

The latter probability is independent of $\mathbf{x}$. Thus the distribution of $E_\Lambda(\mathbf{x} + \mathbf{d})$ is independent of $\mathbf{x}$.

The lattice $\Lambda = \Delta\mathbb{Z}^m$ is a product lattice, and the dither is sampled independently in all coordinates. Therefore the coordinates of $E_\Lambda(\mathbf{x} + \mathbf{d})$ are independent. By Proposition 1, each coordinate follows $\chi_{q,p}$. $\square$

Theorem 1 applies to every input $\mathbf{x} \in \mathbb{Z}_q^m$, including inputs depending on the public matrix and the secret. Applying it to $\mathbf{x} = \mathbf{A}\mathbf{s}$ in an LWQ sample gives an additive noise normal form over $\mathbb{Z}_q^m$ after dither compensation.

Lemma 2. The real LWQ normal form distribution is exactly the real LWE distribution with secret distribution χ_s and error distribution $\chi_{q,p}$. The random LWQ normal form distribution is exactly the uniform distribution over $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$.

Proof. For a real LWQ sample, let $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$, $\mathbf{s} \leftarrow \chi_s^n$, and $\mathbf{d} \leftarrow \mathcal{U}(\mathcal{D}_\Delta^m)$. Set $\mathbf{x} = \mathbf{A}\mathbf{s} \in \mathbb{Z}_q^m$. Let $\mathbf{c} = Q_{\Lambda, \mathbf{d}}(\mathbf{x})$, and let $\mathbf{b} = \mathbf{c} - \mathbf{d}$. Then

$$\mathbf{b} = \mathbf{A}\mathbf{s} + (Q_\Lambda(\mathbf{x} + \mathbf{d}) - (\mathbf{x} + \mathbf{d})) \in \mathbb{Z}_q^m. \quad (9)$$

By Theorem 1, the error term $\mathbf{e} = Q_\Lambda(\mathbf{x} + \mathbf{d}) - (\mathbf{x} + \mathbf{d})$ is independent of $\mathbf{x} = \mathbf{A}\mathbf{s}$. The theorem also gives that the coordinates of $\mathbf{e}$ are independent and each follows $\chi_{q,p}$. Thus $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, where $\mathbf{e} \leftarrow \chi_{q,p}^m$ is independent of $(\mathbf{A}, \mathbf{s})$. This is exactly the real LWE distribution.

The random LWQ normal form distribution is, by definition, the uniform distribution over $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$. This is exactly the random branch of decisional LWE. $\square$

Theorem 2. For every PPT distinguisher $\mathcal{A}$ for LWQ normal form samples versus uniform normal form samples, there exists a PPT distinguisher $\mathcal{B}$ for decisional LWE with secret distribution χ_s and error distribution $\chi_{q,p}$ such that

$$\text{Adv}_{q,n,m,p,\chi_s}^{\text{LWQ}}(\mathcal{A}) = \text{Adv}_{q,n,m,\chi_s,\chi_{q,p}}^{\text{LWE}}(\mathcal{B}). \quad (10)$$

One may take $\mathcal{B} = \mathcal{A}$.

Proof. The reduction is the identity map on normal form samples. By Lemma 2, real LWQ normal form samples are exactly real LWE samples with secret distribution χ_s and error distribution $\chi_{q,p}$. By the same lemma, uniform LWQ normal form samples are exactly uniform LWE samples. Thus the identity map sends real samples to real samples and uniform samples to uniform samples with no statistical loss. The distinguishing advantage is preserved exactly. $\square$

3.3. Split and Normal Forms of LWQ

For the split representation of LWQ [16], we make the map to the normal form explicit in the aligned integer setting. The correspondence follows from the unique decomposition of a normal form coordinate into a quantized coordinate and a dither coordinate. The quantized coordinate belongs to $\Lambda \cap \mathbb{Z}_q^m$, with $\mathbb{Z}_q^m$ represented by canonical integer representatives. Define the public map

$$\Phi : (\Lambda \cap \mathbb{Z}_q^m) \times \mathcal{D}_\Delta^m \rightarrow \mathbb{Z}_q^m, \quad \Phi(\mathbf{c}, \mathbf{d}) \triangleq (\mathbf{c} - \mathbf{d}) \bmod q. \quad (11)$$

For each canonical coordinate $b_i \in \{0, \dots, q-1\}$, set

$$d_i = (-b_i) \bmod \Delta, \quad c_i = (b_i + d_i) \bmod q. \quad (12)$$

Both reductions return canonical representatives, so $d_i \in \{0, \dots, \Delta-1\}$ and $c_i \in \{0, \Delta, \dots, q-\Delta\}$. The full inverse, with all reductions applied coordinate-wise, is

$$\begin{aligned} \Phi^{-1} : \mathbb{Z}_q^m &\rightarrow (\Lambda \cap \mathbb{Z}_q^m) \times \mathcal{D}_\Delta^m, \\ \mathbf{b} &\mapsto ((\mathbf{b} + (-\mathbf{b}) \bmod \Delta), (-\mathbf{b}) \bmod \Delta). \end{aligned} \quad (13)$$

The matrix $\mathbf{A}$ is unchanged in either direction. Each coordinate requires a constant number of elementary arithmetic operations, giving $O(m)$ operations for an m -dimensional sample.

Lemma 3. *The map Φ is a bijection. Moreover, Φ maps real split form samples to real normal form samples, and Φ^{-1} maps real normal form samples to real split form samples. It also maps uniform split form samples to uniform normal form samples, and Φ^{-1} maps uniform normal form samples to uniform split form samples.*

Proof. It suffices to prove the scalar bijection. Fix $b \in \mathbb{Z}_q$ and use its canonical representative in $\{0, \dots, q-1\}$. Since $\Delta \mid q$, the congruence $d \equiv -b \pmod{\Delta}$ has a unique solution $d \in \{0, \dots, \Delta-1\}$. Then $b + d$ is divisible by Δ in $\mathbb{Z}_q$, and there is a unique $a \in \mathbb{Z}_p$ such that $\Delta a = b + d$ in $\mathbb{Z}_q$. Thus $c = \Delta a$ is the unique element of $\Delta\mathbb{Z} \cap \mathbb{Z}_q$ satisfying $b = c - d$. The uniqueness of d follows by reducing $b = c - d$ modulo Δ , and the uniqueness of c follows once d is fixed. Applying the scalar argument coordinate wise proves that Φ is a bijection from $(\Lambda \cap \mathbb{Z}_q^m) \times \mathcal{D}_\Delta^m$ to $\mathbb{Z}_q^m$.

Now consider a real Split-LWQ sample. It has the form $(\mathbf{A}, \mathbf{c}, \mathbf{d})$, where $\mathbf{c} = \mathbf{Q}_{\Lambda, \Delta}(\mathbf{A}\mathbf{s})$. Its normal form is $(\mathbf{A}, \Phi(\mathbf{c}, \mathbf{d}))$, which is exactly $(\mathbf{A}, \mathbf{b})$ with $\mathbf{b} = \mathbf{c} - \mathbf{d}$. Thus Φ maps real split form samples to real normal form samples. Conversely, bijectivity implies that Φ^{-1} recovers the same split form pair $(\mathbf{c}, \mathbf{d})$ from the normal form. Thus Φ and Φ^{-1} preserve real samples exactly.

The uniform split form distribution samples $\mathbf{c} \leftarrow \mathcal{U}(\Lambda \cap \mathbb{Z}_q^m)$ and $\mathbf{d} \leftarrow \mathcal{U}(\mathcal{D}_\Delta^m)$ independently. Since Φ is a bijection from $(\Lambda \cap \mathbb{Z}_q^m) \times \mathcal{D}_\Delta^m$ to $\mathbb{Z}_q^m$, its image of the uniform split form distribution is the uniform normal form distribution over $\mathbb{Z}_q^m$. The same bijection implies that Φ^{-1} maps uniform normal form samples to uniform split form samples. $\square$

Theorem 3. *For every PPT distinguisher $\mathcal{A}$ for Split-LWQ versus the uniform split form distribution, there exists a PPT distinguisher $\mathcal{B}$ for decisional LWE with secret distribution χ_s and error distribution $\chi_{q,p}$ such that*

$$\text{Adv}_{q,n,m,p,\chi_s}^{\text{Split-LWQ}}(\mathcal{A}) = \text{Adv}_{q,n,m,\chi_s,\chi_{q,p}}^{\text{LWE}}(\mathcal{B}). \quad (14)$$

More explicitly, $\mathcal{B}$ maps an input normal form sample $(\mathbf{A}, \mathbf{b})$ to $(\mathbf{A}, \mathbf{c}, \mathbf{d})$, where $(\mathbf{c}, \mathbf{d}) = \Phi^{-1}(\mathbf{b})$, and then runs $\mathcal{A}$.

Proof. Assume first that $(\mathbf{A}, \mathbf{b})$ is drawn from the real decisional LWE distribution with secret distribution χ_s and error distribution $\chi_{q,p}$. By Lemma 2, this distribution is exactly the real LWQ normal form distribution. By Lemma 3, applying Φ^{-1} maps this real normal form sample to its real split form. Therefore the input given to $\mathcal{A}$ is distributed exactly as a real Split-LWQ sample.

Assume next that $(\mathbf{A}, \mathbf{b})$ is drawn from the uniform distribution in decisional LWE. Then $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$ and $\mathbf{b} \leftarrow \mathcal{U}(\mathbb{Z}_q^m)$ are independent. By Lemma 3, the inverse decomposition $(\mathbf{c}, \mathbf{d}) = \Phi^{-1}(\mathbf{b})$ has exactly the uniform split form distribution. Therefore the uniform LWE sample is mapped exactly to a uniform split form sample.

The reduction sends real samples to real split form samples and uniform samples to uniform split form samples with no statistical loss. The distinguishing advantage is therefore preserved exactly. $\square$

Corollary 1. *In the divisible power-of-two setting, the decision problems for LWQ and Split-LWQ are equivalent. More precisely, every PPT distinguisher against one form yields a PPT distinguisher against the other form with the same distinguishing advantage.*

Proof. The direction from split form to normal form uses the public map Φ . The direction from normal form to split form uses the inverse map Φ^{-1} . By Lemma 3, these transformations preserve both real samples and uniform samples. Since Φ is a bijection, the two transformations are inverse to each other on the support of the corresponding experiments. Hence both directions preserve distinguishing advantages exactly. $\square$

3.4. Hardness of LWE with Quantization Errors

The reductions above identify the exact decision problem induced by LWQ- $\mathbb{Z}$. Let $\Delta = q/p$, and define $I_\Delta \triangleq \{-\Delta/2, \dots, \Delta/2 - 1\} \subset \mathbb{Z}$. Define the bounded-sample uniform-error LWE distribution

$$\text{uLWE}_{q,n,m,\chi_s,\Delta} \triangleq (\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m, \quad (15)$$

where $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$, $\mathbf{s} \leftarrow \chi_s^n$, $\mathbf{e} \leftarrow \mathcal{U}(I_\Delta)^m$. The associated decision problem is to distinguish this distribution from $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$.

Lemma 4. *The decision problem with error distribution $\chi_{q,p}$ is equivalent, with identical distinguishing advantage, to the decision problem $\text{uLWE}_{q,n,m,\chi_s,\Delta}$.*

Proof. By Proposition 1, $\chi_{q,p} = \mathcal{U}(\{-\Delta/2 + 1, \dots, \Delta/2\})$. If $e \leftarrow \chi_{q,p}$, then $e - 1 \leftarrow \mathcal{U}(I_\Delta)$. For vectors, the public translation $T(\mathbf{A}, \mathbf{b}) \triangleq (\mathbf{A}, \mathbf{b} - \mathbf{1})$ is a bijection on $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, where $\mathbf{1} \in \mathbb{Z}_q^m$ is the all-one vector. It maps uniform samples to uniform samples, and it maps LWE samples with error $\chi_{q,p}^m$ to samples with error $\mathcal{U}(I_\Delta)^m$. Composing any distinguisher with T or T^{-1} preserves its distinguishing advantage exactly. $\square$

Combining Lemma 4 with Theorem 2 gives the hardness target in its canonical form:

$$\text{LWQ}_{q,p,\chi_s}^m \equiv \text{uLWE}_{q,n,m,\chi_s,\Delta} \quad (16)$$

as decision problems, with no statistical loss in either direction. For the split representation used by the construction, Theorem 3 and Corollary 1 give the same target after the public bijection Φ . Thus the quantization channel contributes an additive LWE error: it is independent of $(\mathbf{A}, \mathbf{s})$, coordinate-wise independent, and uniform on a full interval of width Δ .

This uniform interval error family is studied in the bounded-sample LWE literature. For m -bounded LWE with $\mathbf{e} \leftarrow \mathcal{U}([-r, r])^m$, a lossy-code reduction shows that solving this average-case family under the stated quantitative conditions yields a quantum algorithm for worst-case lattice problems [27]. Equivalently, for $r = \rho q$, this worst-to-average-case reduction can be summarized as

$$\begin{aligned} \text{GapSVP}_{\tilde{O}(n^{1+\sigma m/\rho})} &\leq_q \text{LWE}(n, m, q, \mathcal{U}([- \rho q, \rho q])), \\ \text{SIVP}_{\tilde{O}(n^{1+\sigma m/\rho})} &\leq_q \text{LWE}(n, m, q, \mathcal{U}([- \rho q, \rho q])), \end{aligned} \quad (17)$$

meaning that an algorithm for the average-case uniform-error LWE problem on the right yields quantum algorithms for either worst-case lattice problem on the left, with m fixed in advance as part of the instance. The quantization error above is exactly of this interval-uniform type, with radius $\Delta/2$. The bounded-sample setting is also the natural one here, since m is fixed by the LWQ instance rather than chosen adaptively by the adversary. This places the induced LWQ problem in a theoretically studied uniform-error LWE family.

The lossy-code viewpoint explains why public dithering is the right interface between quantization and LWE. Lossiness for interval errors is proved by showing that, for a lossy code, a uniformly sampled error from an interval admits multiple compatible preimages, whereas a random code remains non-lossy under the same bounded-noise model. In this LWQ instantiation, dithering makes the quantization residue independent of the hidden linear value and uniformly distributed over the quantization cell. The resulting decision problem is therefore an LWE instance with quantization errors, not a rounding problem with input-dependent noise.

The distributional parameters used by this target are explicit: for $e \leftarrow \chi_{q,p}$, $H_\infty(e) = \log_2 \Delta$, $|e| \leq \Delta/2$, $\mathbb{E}[e] = 1/2$, and $\text{Var}(e) = (\Delta^2 - 1)/12$. For $\mathbf{e} \leftarrow \chi_{q,p}^m$, Theorem 1 gives independent coordinates and $H_\infty(\mathbf{e}) = m \log_2 \Delta$. The hardness assumption reached by the preceding reductions is therefore the precisely specified bounded-sample LWE problem $\text{uLWE}_{q,n,m,\chi_s,\Delta}$, whose error is the uniform interval distribution induced by quantization.

4. CPA Secure Public Key Encryption

This section defines the public key encryption scheme underlying Frost and proves its IND-CPA security. The construction represents each transmitted component in split LWQ form: a compact coordinate together with the seed-derived public dither that determines its normal-form reconstruction over $\mathbb{Z}_q$. The proof then moves through the split-normal correspondence of Section 3. In the divisible power-of-two setting, this correspondence is a public bijection: real split transcripts map to LWE normal-form samples with the explicit quantization error distribution $\chi_{q,p}$, while uniform split transcripts map to uniform normal-form samples.

4.1. Specification of Frost.PKE

The scheme is parameterized by $(n, m, \ell, q, \eta_s, \eta_r, p_{\text{pk}}, p_u, p_v, b_{\text{msg}})$. Here $q = 2^{t_1}$, and $p_x = 2^{t_x}$ for $x \in \{\text{pk}, u, v\}$, so $p_x \mid q$ for each transmitted component. Write $\Delta_x \triangleq q/p_x$. The long term secret and the ephemeral secret are sampled entrywise as $\mathbf{S} \leftarrow \mathcal{B}_{\eta_s}^{n \times \ell}$ and $\mathbf{R} \leftarrow \mathcal{B}_{\eta_r}^{m \times \ell}$.

For $x \in \{\text{pk}, u, v\}$, the quantization lattice is $\Lambda_x \triangleq \Delta_x \mathbb{Z}^{N_x}$, where $N_{\text{pk}} = m\ell$, $N_u = n\ell$, and $N_v = \ell^2$. The dither domain is $\mathcal{D}_x \triangleq \{0, \dots, \Delta_x - 1\}^{N_x}$. A matrix input is vectorized, quantized over the corresponding product lattice, and reshaped back to matrix form. For $x \in \{\text{pk}, u, v\}$, multiplication by Δ_x is a coordinate wise bijection from $\mathbb{Z}_{p_x}^{N_x}$ onto $\Lambda_x \cap \mathbb{Z}_q^{N_x}$. Whenever $\mathbf{C} \in \Lambda_x \cap \mathbb{Z}_q^{N_x}$, the notation $\mathbf{C}/\Delta_x$ denotes the unique preimage under this bijection. Thus the algorithms store compact coordinates in $\mathbb{Z}_{p_x}$, while the corresponding normal form value is obtained by multiplying by Δ_x and subtracting the public dither.

The procedure $\text{GenPublic}(\rho) \rightarrow (\mathbf{A}, \mathbf{D}_{\text{pk}})$ deterministically derives a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a public key dither $\mathbf{D}_{\text{pk}} \in \mathcal{D}_{\text{pk}}$. The procedure $\text{GenDither}(\mu) \rightarrow (\mathbf{D}_u, \mathbf{D}_v)$ deterministically derives ciphertext dithers $\mathbf{D}_u \in \mathcal{D}_u$ and $\mathbf{D}_v \in \mathcal{D}_v$. The procedure $\text{GenSecret}_{\eta_r}(\sigma) \rightarrow \mathbf{R}$ deterministically expands the seed σ into an ephemeral secret in $\mathbb{Z}^{m \times \ell}$ using domain-separated centered-binomial sampling. Its prescribed output distribution in the ideal expansion experiment is $\mathcal{B}_{\eta_r}^{m \times \ell}$. The encoding map $\text{Encode} : \mathcal{M}_{\text{PKE}} \rightarrow \mathbb{Z}_q^{\ell \times \ell}$ is the E_8 coded block encoder specified in Section 4.2. It parses the plaintext into ℓ column payloads and maps each payload to one representative of the finite lattice code $\mathcal{C}_{E_8} \subseteq \mathbb{Z}_q^8$. The decoding map $\text{Decode} : \mathbb{Z}_q^{\ell \times \ell} \rightarrow \mathcal{M}_{\text{PKE}}$ applies the corresponding E_8 nearest neighbor decoder to each column, followed by the public inverse labeling map.

We write $\text{Frost.PKE.Enc}(\text{pk}, \mathbf{M}, r_{\text{Enc}})$ for encryption with explicit coins. The coin string is parsed as $r_{\text{Enc}} \mapsto (\sigma, \mu)$. The seed σ determines the random tape for sampling $\mathbf{R} \leftarrow \mathcal{B}_{\eta_r}^{m \times \ell}$. The seed μ determines the ciphertext side dithers. In randomized PKE encryption, r_{Enc} is sampled uniformly. Inside the FO transform, r_{Enc} is derived from the FO hash.

Algorithm 1: $\text{Frost.PKE.KeyGen}()$

Output: Public key pk , secret key sk

- 1 $\rho \leftarrow \mathcal{U}(\{0, 1\}^{256})$;
- 2 $(\mathbf{A}, \mathbf{D}_{\text{pk}}) \leftarrow \text{GenPublic}(\rho)$;
- 3 $\mathbf{S} \leftarrow \mathcal{B}_{\eta_s}^{n \times \ell}$;
- 4 $\mathbf{B} \leftarrow \mathbf{Q}_{\Lambda_{\text{pk}}, \mathbf{D}_{\text{pk}}}(\mathbf{A}\mathbf{S})/\Delta_{\text{pk}}$;
- 5 $\text{pk} \leftarrow (\rho, \mathbf{B})$;
- 6 $\text{sk} \leftarrow \mathbf{S}$;
- 7 **return** (pk, sk)

Algorithm 2: $\text{Frost.PKE.Enc}(\text{pk}, \mathbf{M}, r_{\text{Enc}})$

Input: Public key $\text{pk} = (\rho, \mathbf{B})$, message $\mathbf{M} \in \mathcal{M}_{\text{PKE}}$, coins r_{Enc}

Output: Ciphertext $\text{ct} = (\mathbf{U}, \mathbf{V}, \mu)$

- 1 $r_{\text{Enc}} \mapsto (\sigma, \mu)$;
- 2 $(\mathbf{A}, \mathbf{D}_{\text{pk}}) \leftarrow \text{GenPublic}(\rho)$;
- 3 $\mathbf{R} \leftarrow \text{GenSecret}_{\eta_r}(\sigma)$;
- 4 $(\mathbf{D}_u, \mathbf{D}_v) \leftarrow \text{GenDither}(\mu)$;
- 5 $\hat{\mathbf{B}} \leftarrow \Delta_{\text{pk}}\mathbf{B} - \mathbf{D}_{\text{pk}}$;
- 6 $\mathbf{U} \leftarrow \mathbf{Q}_{\Lambda_u, \mathbf{D}_u}(\mathbf{A}^\top \mathbf{R})/\Delta_u$;
- 7 $\mathbf{V} \leftarrow \mathbf{Q}_{\Lambda_v, \mathbf{D}_v}(\hat{\mathbf{B}}^\top \mathbf{R} + \text{Encode}(\mathbf{M}))/\Delta_v$;
- 8 **return** $\text{ct} = (\mathbf{U}, \mathbf{V}, \mu)$

Algorithm 3: $\text{Frost.PKE.Dec}(\text{sk}, \text{ct})$

Input: Secret key $\text{sk} = \mathbf{S}$, ciphertext $\text{ct} = (\mathbf{U}, \mathbf{V}, \mu)$

Output: Message $\mathbf{M}' \in \mathcal{M}_{\text{PKE}}$

- 1 $(\mathbf{D}_u, \mathbf{D}_v) \leftarrow \text{GenDither}(\mu)$;
- 2 $\hat{\mathbf{U}} \leftarrow \Delta_u \mathbf{U} - \mathbf{D}_u$;
- 3 $\hat{\mathbf{V}} \leftarrow \Delta_v \mathbf{V} - \mathbf{D}_v$;
- 4 $\mathbf{W} \leftarrow \hat{\mathbf{V}} - \mathbf{S}^\top \hat{\mathbf{U}}$;
- 5 $\mathbf{M}' \leftarrow \text{Decode}(\mathbf{W})$;
- 6 **return** $\mathbf{M}'$

The public key is $\text{pk} = (\rho, \mathbf{B})$, where ρ determines both $\mathbf{A}$ and $\mathbf{D}_{\text{pk}}$. The ciphertext is $\text{ct} = (\mathbf{U}, \mathbf{V}, \mu)$, where μ determines the ciphertext side dithers. The seed μ is transmitted explicitly so that decryption and FO re-encryption regenerate the same public dithers.

The proof separates the algebraic reduction from the symmetric expansion procedures. We analyze an ideal expansion experiment in which the domain separated expansion calls used by GenPublic , GenDither , and $\text{GenSecret}_{\eta_r}$ are replaced by ideal expansion oracles with the prescribed output distributions. In this ideal experiment, $\text{GenPublic}(\rho)$

returns $(\mathbf{A}, \mathbf{D}_{\text{pk}})$, where $\mathbf{A}$ is uniform over $\mathbb{Z}_q^{m \times n}$ and $\mathbf{D}_{\text{pk}}$ is uniform over $\mathcal{D}_{\text{pk}}$. The procedure $\text{GenDither}(\mu)$ returns $(\mathbf{D}_u, \mathbf{D}_v)$ uniformly over $\mathcal{D}_u \times \mathcal{D}_v$. The procedure $\text{GenSecret}_{\eta_r}(\sigma)$ returns $\mathbf{R}$ according to $\mathcal{B}_{\eta_r}^{m \times \ell}$, deterministically for each fixed seed σ . For an adversary $\mathcal{C}_{\text{Exp}}$, let $\text{Adv}_{\text{Exp}}(\mathcal{C}_{\text{Exp}})$ denote its distinguishing advantage between the concrete expansion procedures and the ideal expansion experiment. This term accounts only for the pseudorandomness of the domain separated expansion calls. It is independent of the LWQ to LWE reduction. When these calls are modeled as random oracles with the prescribed output distributions, this term is zero.

4.2. Lattice Coded Message Encoding and Decoding

Lattice codes combine message embedding and error correction in a single geometric object [22], [23], [24]. We use the Gosset lattice E_8 to instantiate the abstract maps Encode and Decode used by Frost.PKE . This layer determines the message representation and the decryption-failure event. The LWQ transcript, the split-normal correspondence, and the LWE reductions in Section 3 remain defined at the quantization layer and are independent of this message code. Following the nested lattice viewpoint [22], we view each ciphertext column as one 8-dimensional lattice code block with hypercube shaping. For a profile with $b_{\text{msg}} \in \{2, 3, 4\}$, let $\alpha_{E_8} \triangleq q/2^{b_{\text{msg}}}$, let the fine lattice be $\Lambda_f = \alpha_{E_8} E_8$, and let the coarse shaping lattice be $\Lambda_c = q\mathbb{Z}^8$. The corresponding finite lattice code is

$$\mathcal{C}_{E_8} = \mathcal{C}(\Lambda_f, \Lambda_c) = \Lambda_f \cap [0, q)^8, \quad (18)$$

where the hypercube $[0, q)^8$ is used as the shaping region. Since $q\mathbb{Z}^8 \subseteq \alpha_{E_8} E_8$ for the chosen power-of-two profiles, this code contains

$$|\mathcal{C}_{E_8}| = \frac{\det(q\mathbb{Z}^8)}{\det(\alpha_{E_8} E_8)} = (q/\alpha_{E_8})^8 = 2^{8b_{\text{msg}}} \quad (19)$$

representatives. Thus one ciphertext column carries $8b_{\text{msg}}$ bits, and the full $\ell = 8$ message matrix carries $64b_{\text{msg}}$ FO plaintext bits. This gives message spaces of size 2^{128} , 2^{192} , and 2^{256} for Frost-128, Frost-192, and Frost-256, respectively. The concrete label map uses the mixed-radix payload layouts $(1, 2, 2, 2, 2, 2, 2, 3)$, $(2, 3, 3, 3, 3, 3, 3, 4)$, and $(3, 4, 4, 4, 4, 4, 4, 5)$ for the three profiles. These layouts only specify the public bijection between bit strings and representatives in $\mathcal{C}_{E_8}$; the decoding radius is determined by $\mathcal{C}_{E_8}$ and α_{E_8} .

Encoding parses each $8b_{\text{msg}}$ -bit column payload, applies the public label map to obtain a representative $\mathbf{c} \in \mathcal{C}_{E_8}$, and places $\mathbf{c}$ as one column of $\text{Encode}(\mathbf{M})$. Decoding takes a received column in $\mathbb{Z}_q^8$, applies nearest-neighbor decoding in $\Lambda_f = \alpha_{E_8} E_8$ to any integer lift, reduces the decoded lattice point modulo the coarse lattice $q\mathbb{Z}^8$ back to $\mathcal{C}_{E_8}$, and then

applies the public inverse label map. For the nearest neighbor step, we use the standard two coset decomposition

$$E_8 = D_8 \cup \left(D_8 + \frac{1}{2} \mathbf{1} \right),$$

$$D_8 = \left\{ \mathbf{z} \in \mathbb{Z}^8 : \sum_{i=1}^8 z_i \equiv 0 \pmod{2} \right\}. \quad (20)$$

After scaling the lifted received vector by $1/\alpha_{E_8}$, the decoder computes the closest candidate in D_8 , computes the closest candidate in $D_8 + \frac{1}{2} \mathbf{1}$, and selects the candidate with smaller squared Euclidean distance. This two-coset procedure avoids exhaustive search over E_8 and is the path implemented by the constant-time decoder discussed in Appendix B.

Let $\mathbf{c} \in \mathcal{C}_{E_8}$ be an encoded block and let $\mathbf{e} \in \mathbb{Z}^8$ be the total decryption error added to it. The received block is $\mathbf{c} + \mathbf{e}$ modulo $q\mathbb{Z}^8$. Nearest-neighbor decoding in $\alpha_{E_8} E_8$ returns a lattice point congruent to $\mathbf{c}$ modulo $q\mathbb{Z}^8$ whenever $\mathbf{e} \in \alpha_{E_8} \text{Vor}(E_8)$, with ties excluded by the fixed decoding convention. Reducing this lattice point to $\mathcal{C}_{E_8}$ recovers $\mathbf{c}$, after which the inverse label map recovers the original bit block.

4.3. Correctness and Decryption Failure Analysis

We analyze correctness through the normal form components reconstructed by decryption. For $x \in \{\text{pk}, \text{u}, \text{v}\}$, write $\chi_x \triangleq \chi_{q, p_x}$. By Proposition 1 and Theorem 1, the normal form components satisfy

$$\begin{aligned} \hat{\mathbf{B}} &= \mathbf{A}\mathbf{S} + \mathbf{E}_{\text{pk}}, \\ \hat{\mathbf{U}} &= \mathbf{A}^\top \mathbf{R} + \mathbf{E}_{\text{u}}, \\ \hat{\mathbf{V}} &= \hat{\mathbf{B}}^\top \mathbf{R} + \text{Encode}(\mathbf{M}) + \mathbf{E}_{\text{v}}. \end{aligned} \quad (21)$$

The entries of $\mathbf{E}_{\text{pk}}$, $\mathbf{E}_{\text{u}}$, and $\mathbf{E}_{\text{v}}$ follow χ_{pk} , χ_{u} , and χ_{v} , respectively, and are independent of the corresponding quantizer inputs. For the selected parameter sets, each Δ_x is even, so each scalar quantization error is uniform over $\{-\Delta_x/2 + 1, \dots, \Delta_x/2\}$, with mean $1/2$, centered variance $(\Delta_x^2 - 1)/12$, and second moment $(\Delta_x^2 + 2)/12$. Substituting the normal forms into decryption cancels the main term:

$$\begin{aligned} \mathbf{W} &= \hat{\mathbf{V}} - \mathbf{S}^\top \hat{\mathbf{U}} \\ &= \text{Encode}(\mathbf{M}) + \mathbf{E}_{\text{pk}}^\top \mathbf{R} - \mathbf{S}^\top \mathbf{E}_{\text{u}} + \mathbf{E}_{\text{v}}. \end{aligned} \quad (22)$$

Thus correctness is governed by the total decryption error

$$\mathbf{E}_{\text{tot}} \triangleq \mathbf{E}_{\text{pk}}^\top \mathbf{R} - \mathbf{S}^\top \mathbf{E}_{\text{u}} + \mathbf{E}_{\text{v}}. \quad (23)$$

Let $\mathbf{e}_j \in \mathbb{Z}^8$ denote the j -th column of $\mathbf{E}_{\text{tot}}$. By the decoding condition above, the j -th message block decodes correctly whenever $\mathbf{e}_j \in \alpha_{E_8} \text{Vor}(E_8)$. It remains to upper bound the probability that this condition fails for at least one ciphertext column. Let $\mathcal{R}(E_8)$ denote the 240 roots of E_8 , normalized so that every $\mathbf{r} \in \mathcal{R}(E_8)$ satisfies $\|\mathbf{r}\|^2 = 2$. The root system consists of 112 roots obtained by permuting $(\pm 1, \pm 1, 0, \dots, 0)$ and 128 roots of the form $(\pm \frac{1}{2}, \dots, \pm \frac{1}{2})$ with an even number of minus signs. Under

this normalization, the scaled Voronoi cell condition can be checked by the root facets. In particular, it is sufficient that

$$\max_{\mathbf{r} \in \mathcal{R}(E_8)} |\langle \mathbf{e}_j, \mathbf{r} \rangle| < \alpha_{E_8}. \quad (24)$$

When this condition holds, $\mathbf{e}_j \in \alpha_{E_8} \text{Vor}(E_8)$, and the j -th E_8 coded block decodes correctly. For a fixed root $\mathbf{r}$, define $X_{\mathbf{r},j} \triangleq \langle \mathbf{e}_j, \mathbf{r} \rangle$. A union bound over the root facets and then over the ℓ ciphertext columns gives

$$\begin{aligned} \delta_{\text{block}} &\leq \sum_{\mathbf{r} \in \mathcal{R}(E_8)} (\Pr[X_{\mathbf{r},j} \geq \alpha_{E_8}] + \Pr[X_{\mathbf{r},j} \leq -\alpha_{E_8}]), \\ \delta_{\text{dec}} &\leq \ell \delta_{\text{block}}. \end{aligned} \quad (25)$$

Each projection tail is evaluated from the exact finite support distribution induced by

$$\mathbf{e}_j = (\mathbf{E}_{\text{pk}}^\top \mathbf{R})_j - (\mathbf{S}^\top \mathbf{E}_{\text{u}})_j + (\mathbf{E}_{\text{v}})_j, \quad (26)$$

where $\mathbf{E}_{\text{pk}}$, $\mathbf{E}_{\text{u}}$, and $\mathbf{E}_{\text{v}}$ follow the corresponding quantization error distributions, while $\mathbf{S}$ and $\mathbf{R}$ follow the centered binomial secret distributions. The calculation is carried out at the root-projection level rather than by multiplying marginal coordinate failure probabilities. Thus dependencies among the eight coordinates of an E_8 block are handled inside the distribution of $X_{\mathbf{r},j}$, while the final step uses only union bounds over facets and ciphertext columns. The computation keeps the nonzero mean $1/2$ of the quantization errors. For $X = X_{\mathbf{r},j}$ and $K_X(\theta) = \log \mathbb{E}[\exp(\theta X)]$, Chernoff's method gives

$$\Pr[X \geq t] \leq \inf_{\theta > 0} \exp(K_X(\theta) - \theta t), \quad (27)$$

$$\Pr[X \leq -t] \leq \inf_{\theta < 0} \exp(K_X(\theta) + \theta t), \quad (28)$$

with $t = \alpha_{E_8}$. The $\log_2 \delta$ values reported in Table 2 are the resulting whole ciphertext upper bounds on δ_{dec} .

4.4. IND-CPA Security

We prove IND-CPA security by a two hop game sequence after replacing the concrete expansion procedures by the ideal expansion experiment. This replacement is accounted for by Adv_{Exp} . The term captures only the pseudorandomness of the domain separated expansion calls and is independent of the LWQ to LWE reduction.

We use matrix secret LWE notation for parallel right hand sides. The advantage $\text{Adv}_{q,n,m,\ell,\chi_s,\chi_e}^{\text{LWE}}$ denotes decisional LWE over $\mathbb{Z}_q^{m \times n}$ with secret matrix in $\mathbb{Z}_q^{n \times \ell}$ sampled entrywise from χ_s , and error matrix in $\mathbb{Z}_q^{m \times \ell}$ sampled entrywise from χ_e . The notation also permits a row wise mixed error distribution.

The public key hop replaces $\mathbf{B} \in \mathbb{Z}_{p_{\text{pk}}}^{m \times \ell}$ by a uniform matrix. By the split and normal form correspondence of Section 3, this hop is bounded by decisional LWE with secret distribution $\mathcal{B}_{\eta_s}$ and error distribution $\chi_{q,p_{\text{pk}}}$. The ciphertext hop replaces $\mathbf{U}$ and $\mathbf{V}$ jointly, since both ciphertext components are generated from the same ephemeral secret $\mathbf{R}$. It is therefore reduced to a row wise mixed error decisional

LWE instance. The upper n rows use χ_{q,p_u} , and the last ℓ rows use χ_{q,p_v} .

For the ciphertext hop, the relevant stacked LWE matrix is

$$\mathbf{C} = \begin{pmatrix} \mathbf{A}^\top \\ \widehat{\mathbf{B}}^\top \end{pmatrix} \in \mathbb{Z}_q^{(n+\ell) \times m}. \quad (29)$$

The secret is $\mathbf{R} \in \mathbb{Z}_q^{m \times \ell}$, sampled entrywise from $\mathcal{B}_{\eta_r}$. The stacked error distribution is denoted by $(\chi_{q,p_u}^n, \chi_{q,p_v}^\ell)$. This means that the upper n rows use error distribution χ_{q,p_u} , while the last ℓ rows use error distribution χ_{q,p_v} .

Theorem 4 (IND-CPA security). *For every PPT IND-CPA adversary $\mathcal{A}$ against the seeded implementation of Frost.PKE, there exist PPT adversaries $\mathcal{C}_{\text{Exp}}$, $\mathcal{B}_1$, and $\mathcal{B}_2$ such that*

$$\begin{aligned} \text{Adv}_{\text{Frost.PKE}}^{\text{IND-CPA}}(\mathcal{A}) &\leq \text{Adv}_{\text{Exp}}(\mathcal{C}_{\text{Exp}}) + \text{Adv}_{q,n,m,\ell,\mathcal{B}_{\eta_s},\chi_{q,p_{\text{pk}}}}^{\text{LWE}}(\mathcal{B}_1) \\ &\quad + \text{Adv}_{q,m,n+\ell,\ell,\mathcal{B}_{\eta_r},(\chi_{q,p_u}^n,\chi_{q,p_v}^\ell)}^{\text{LWE}}(\mathcal{B}_2). \end{aligned} \quad (30)$$

Here Adv_{Exp} denotes the distinguishing advantage for replacing the concrete expansion procedures by the ideal expansion experiment.

Proof. Let S_{real} be the event that $\mathcal{A}$ outputs the correct challenge bit in the seeded IND-CPA experiment. Let S_i be the corresponding event in Game G_i below. The transition from the seeded experiment to Game G_0 replaces the concrete expansion procedures by the ideal expansion experiment. By the definition of Adv_{Exp} , there exists a PPT adversary $\mathcal{C}_{\text{Exp}}$ such that

$$|\Pr[S_{\text{real}}] - \Pr[S_0]| \leq \text{Adv}_{\text{Exp}}(\mathcal{C}_{\text{Exp}}). \quad (31)$$

It remains to bound $\Pr[S_0] - 1/2$ in the ideal expansion experiment.

Game G_0 . This game is the IND-CPA experiment after the expansion replacement. The public key is generated as in Algorithm 1. On challenge messages $\mathbf{M}_0, \mathbf{M}_1$, the challenger samples $b \leftarrow \mathcal{U}(\{0, 1\})$, samples encryption coins $r_{\text{Enc}} \mapsto (\sigma, \mu)$, and returns $\text{Frost.PKE.Enc}(\text{pk}, \mathbf{M}_b, r_{\text{Enc}})$.

Game G_1 . We replace $\mathbf{B}$ by an independent uniform matrix in $\mathbb{Z}_{p_{\text{pk}}}^{m \times \ell}$. In the real branch of Game G_0 , the corresponding normal form satisfies

$$\widehat{\mathbf{B}} = \Delta_{\text{pk}} \mathbf{B} - \mathbf{D}_{\text{pk}} = \mathbf{A} \mathbf{S} + \mathbf{E}_{\text{pk}}, \quad (32)$$

where the entries of $\mathbf{E}_{\text{pk}}$ follow $\chi_{q,p_{\text{pk}}}$. Thus the real public key normal form sample is a matrix secret LWE sample with secret distribution $\mathcal{B}_{\eta_s}$ and error distribution $\chi_{q,p_{\text{pk}}}$. In Game G_1 , $\mathbf{B}$ is uniform over $\mathbb{Z}_{p_{\text{pk}}}^{m \times \ell}$, so $\Delta_{\text{pk}} \mathbf{B}$ is uniform over $\Lambda_{\text{pk}} \cap \mathbb{Z}_q^{m \times \ell}$ after vectorization. Together with the public dither $\mathbf{D}_{\text{pk}}$, this is exactly the split representation of a uniform normal form sample. By the split and normal form correspondence of Section 3, there exists a PPT adversary $\mathcal{B}_1$ such that

$$|\Pr[S_0] - \Pr[S_1]| \leq \text{Adv}_{q,n,m,\ell,\mathcal{B}_{\eta_s},\chi_{q,p_{\text{pk}}}}^{\text{LWE}}(\mathcal{B}_1). \quad (33)$$

Game G_2 . In Game G_1 , the public key normal form is independent of $\mathbf{S}$. We now replace $\mathbf{U}$ and $\mathbf{V}$ jointly by independent uniform matrices in $\mathbb{Z}_{p_u}^{n \times \ell}$ and $\mathbb{Z}_{p_v}^{\ell \times \ell}$. In the real

branch, define $\widehat{\mathbf{U}} = \Delta_u \mathbf{U} - \mathbf{D}_u$ and $\widehat{\mathbf{V}} = \Delta_v \mathbf{V} - \mathbf{D}_v$. The stacked normal form ciphertext satisfies

$$\begin{pmatrix} \widehat{\mathbf{U}} \\ \widehat{\mathbf{V}} - \text{Encode}(\mathbf{M}_b) \end{pmatrix} = \begin{pmatrix} \mathbf{A}^\top \\ \widehat{\mathbf{B}}^\top \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_u \\ \mathbf{E}_v \end{pmatrix}. \quad (34)$$

The upper n error rows follow χ_{q,p_u} . The last ℓ error rows follow χ_{q,p_v} . The shift by $\text{Encode}(\mathbf{M}_b)$ in the last block is fixed after the challenge message is chosen and preserves both branches of the decisional experiment. Under the compact to split identification $z \mapsto \Delta_x z$, the split and normal form correspondence transfers the row wise mixed error LWE hop to the ciphertext distribution used in the game. Therefore there exists a PPT adversary $\mathcal{B}_2$ such that

$$|\Pr[S_1] - \Pr[S_2]| \leq \text{Adv}_{q,m,n+\ell,\ell,\mathcal{B}_{\eta_r},(\chi_{q,p_u}^n,\chi_{q,p_v}^\ell)}^{\text{LWE}}(\mathcal{B}_2). \quad (35)$$

In Game G_2 , both $\mathbf{U}$ and $\mathbf{V}$ are uniform and independent of $\mathbf{M}_b$. Hence the challenge ciphertext is independent of the challenge bit, and $\Pr[S_2] = 1/2$. Combining the expansion replacement with the two LWE hops gives the claimed bound. $\square$

5. CCA Secure KEM via Fujisaki Okamoto

We obtain Frost.KEM by applying the Fujisaki Okamoto transform with implicit rejection to the IND-CPA secure PKE from Section 4. The resulting KEM uses deterministic encryption in encapsulation and in the decapsulation re-encryption check. The security proof depends on the IND-CPA advantage of Frost.PKE, the whole ciphertext decryption failure probability δ_{dec} , and the size of the FO plaintext space.

5.1. KEM Construction

Let $\mathcal{G}$ and $\mathcal{H}$ be independent random oracles with domain separated inputs. We write $\mathcal{G}_{\text{FO}}$ for the domain separated instance of $\mathcal{G}$ used to derive the preliminary key and the PKE coins. We write $\mathcal{H}_{\text{pk}}$, $\mathcal{H}_{\text{ct}}$, and $\mathcal{H}_{\text{K}}$ for the domain separated instances of $\mathcal{H}$ used for public key hashing, ciphertext hashing, and final key derivation. We use the deterministic encryption procedure $\text{Frost.PKE.Enc}(\text{pk}, \mathbf{M}, r_{\text{Enc}})$ from Section 4. The coin string r_{Enc} is parsed as $r_{\text{Enc}} \mapsto (\sigma, \mu)$. The seed σ determines the ephemeral secret $\mathbf{R}$. The seed μ determines the ciphertext side dithers $(\mathbf{D}_u, \mathbf{D}_v)$. Let $\mathcal{M}_{\text{FO}} \subseteq \mathcal{M}_{\text{PKE}}$ be the FO plaintext space.

Algorithm 4: Frost.KEM.KeyGen()

Output: Public key pk , secret key sk

- 1 $z \leftarrow \mathcal{U}(\{0, 1\}^{256});$
 - 2 $(\text{pk}, \text{sk}_{\text{pke}}) \leftarrow \text{Frost.PKE.KeyGen}();$
 - 3 $h_{\text{pk}} \leftarrow \mathcal{H}_{\text{pk}}(\text{pk});$
 - 4 $\text{sk} \leftarrow (\text{sk}_{\text{pke}}, \text{pk}, h_{\text{pk}}, z);$
 - 5 **return** (pk, sk)
-

The explicit ciphertext seed μ is needed for deterministic decapsulation. It lets the decryptor regenerate the same

Algorithm 5: Frost.KEM.Encaps(pk)

Input: Public key pk**Output:** Ciphertext ct, shared key K

- 1 $h_{pk} \leftarrow \mathcal{H}_{pk}(pk);$
 - 2 $\mathbf{M} \leftarrow \mathcal{U}(\mathcal{M}_{FO});$
 - 3 $(\hat{K}, r_{Enc}) \leftarrow \mathcal{G}_{FO}(h_{pk}, \mathbf{M});$
 - 4 $ct \leftarrow \text{Frost.PKE.Enc}(pk, \mathbf{M}, r_{Enc});$
 - 5 $K \leftarrow \mathcal{H}_K(\hat{K}, \mathcal{H}_{ct}(ct));$
 - 6 **return** (ct, K)
-

Algorithm 6: Frost.KEM.Decaps(sk, ct)

Input: Secret key $sk = (sk_{pke}, pk, h_{pk}, z)$, ciphertext ct**Output:** Shared key K

- 1 $\mathbf{M}' \leftarrow \text{Frost.PKE.Dec}(sk_{pke}, ct);$
 - 2 $(\hat{K}', r'_{Enc}) \leftarrow \mathcal{G}_{FO}(h_{pk}, \mathbf{M}');$
 - 3 $ct' \leftarrow \text{Frost.PKE.Enc}(pk, \mathbf{M}', r'_{Enc});$
 - 4 **if** $ct = ct'$ **then**
 - 5 $K \leftarrow \mathcal{H}_K(\hat{K}', \mathcal{H}_{ct}(ct));$
 - 6 **else**
 - 7 $K \leftarrow \mathcal{H}_K(z, \mathcal{H}_{ct}(ct));$
 - 8 **return** K
-

ciphertext side dithers and reproduce the same split ciphertext in the FO re-encryption check. The conditional branch in Algorithm 6 specifies the implicit-rejection functionality; the implementation realizes this final selection in constant time, so both candidate keys are always computed. Once $(pk, \mathbf{M}, r_{Enc})$ is fixed, encryption is deterministic, which is the scheme specific condition required by the FO transform. For our parameter sets, we fix $\ell = 8$ and choose $b_{msg} \in \{2, 3, 4\}$. Thus $|\mathcal{M}_{FO}| = 2^{b_{msg}\ell^2}$ gives message spaces of size 2^{128} , 2^{192} , and 2^{256} . The final shared key $K \in \{0, 1\}^{256}$ is obtained from $\mathcal{H}_K$.

5.2. IND-CCA Security

The FO proof applies once the underlying PKE is IND-CPA secure and has negligible decryption failure probability. Let δ_{dec} denote the whole ciphertext decryption failure probability of Frost.PKE from Section 4.3. Let $q_{RO} \triangleq q_G + q_H$, where q_G counts queries to $\mathcal{G}_{FO}$, and q_H counts all queries to the domain separated instances of $\mathcal{H}$.

Theorem 5 (Classical ROM). *Assume that $\mathcal{G}$ and $\mathcal{H}$ are random oracles. For every classical adversary $\mathcal{A}$ against the IND-CCA security of Frost.KEM making at most q_G queries to $\mathcal{G}_{FO}$, at most q_H total queries to the domain separated instances of $\mathcal{H}$, and at most q_{dec} decapsulation queries, there exists an IND-CPA adversary $\mathcal{B}$ against Frost.PKE such that*

$$\text{Adv}_{\text{Frost.KEM}}^{\text{IND-CCA}}(\mathcal{A}) \leq 3\text{Adv}_{\text{Frost.PKE}}^{\text{IND-CPA}}(\mathcal{B}) + q_{RO}\delta_{dec} + \frac{3q_{RO}}{|\mathcal{M}_{FO}|}. \quad (36)$$

The running time of $\mathcal{B}$ is approximately that of $\mathcal{A}$ plus $O(q_{RO})$ PKE operations.

Proof. We apply the modular analysis of the implicit-rejection $\text{FO}^\perp$ transform in [39, Section 3.3]. The corresponding bound applies to a PKE with message space $\mathcal{M}$ and correctness error δ , after derandomization by the FO coins. The scheme specific condition is determinism of encryption once $(pk, \mathbf{M}, r_{Enc})$ is fixed. In Frost.PKE, the coin string r_{Enc} fixes the ephemeral secret through σ and the ciphertext side dithers through μ . The public seed ρ in pk fixes the public matrix and the public key dither. Hence the re-encryption step in decapsulation is well defined. The term $q_{RO}\delta_{dec}$ accounts for random oracle queries whose associated ciphertexts hit a decryption failure event. The term $3q_{RO}/|\mathcal{M}_{FO}|$ accounts for message space collisions in the implicit-rejection FO analysis. Substituting $\mathcal{M} = \mathcal{M}_{FO}$ and $\delta = \delta_{dec}$ gives the stated bound. $\square$

Theorem 6 (Quantum ROM). *Assume that $\mathcal{G}$ and $\mathcal{H}$ are quantum accessible random oracles. For every quantum adversary $\mathcal{A}$ against the IND-CCA security of Frost.KEM making at most q_G quantum queries to $\mathcal{G}_{FO}$, at most q_H quantum or classical queries to the domain separated instances of $\mathcal{H}$, and at most q_{dec} classical decapsulation queries, there exists an IND-CPA adversary $\mathcal{B}$ against Frost.PKE such that*

$$\begin{aligned} \text{Adv}_{\text{Frost.KEM}}^{\text{IND-CCA}}(\mathcal{A}) &\leq 2\sqrt{(q_{RO} + 1)\text{Adv}_{\text{Frost.PKE}}^{\text{IND-CPA}}(\mathcal{B})} + \frac{2(q_{RO} + 1)^2}{|\mathcal{M}_{FO}|} \\ &\quad + \frac{2q_H}{\sqrt{|\mathcal{M}_{FO}|}} + 4q_G\sqrt{\delta_{dec}}. \end{aligned} \quad (37)$$

The adversary $\mathcal{B}$ makes at most $2q_G$ queries to its own challenger, and its running time is approximately that of $\mathcal{A}$ plus $O(q_{RO})$ PKE operations.

Proof. We apply the QROM analysis of $\text{FO}^\perp$ in [40], [41]. In particular, [41, Theorem 3] gives a tighter IND-CPA-to-IND-CCA reduction for the implicit-rejection FO transform whose final key is derived from the FO message and ciphertext, with a square-root term in the IND-CPA advantage and the FO plaintext-space size, a quantum search term over the message space, and the quantum amplification term for decryption failures. The same scheme specific determinism condition as in the classical proof holds. Once $(pk, \mathbf{M}, r_{Enc})$ is fixed, encryption is deterministic, because r_{Enc} fixes the ephemeral secret and the ciphertext side dithers. The decapsulation algorithm decrypts, derives r'_{Enc} , re-encrypts, and applies implicit rejection. The only correctness loss is the event that decryption crosses the decoding boundary, which is accounted for by δ_{dec} . Applying the generic theorem with $\mathcal{M} = \mathcal{M}_{FO}$, $\delta = \delta_{dec}$, and $q_{RO} = q_G + q_H$ gives the stated inequality. $\square$

For the concrete parameter sets, the FO plaintext space has size

$$|\mathcal{M}_{FO}| = 2^{b_{msg}\ell^2}. \quad (38)$$

Thus $|\mathcal{M}_{FO}| = 2^{128}$, 2^{192} , and 2^{256} for Frost-128, Frost-192, and Frost-256, respectively. For any fixed classical random-

oracle query budget q_{RO} , the message space term in Theorem 5 is

$$\frac{3q_{\text{RO}}}{|\mathcal{M}_{\text{FO}}|}. \quad (39)$$

The decryption failure contribution is $q_{\text{RO}}\delta_{\text{dec}}$, where δ_{dec} is the whole ciphertext bound reported in Table 2. In the quantum bound, the message space terms scale as $2(q_{\text{RO}} + 1)/\sqrt{|\mathcal{M}_{\text{FO}}|}$ and $2q_{\text{H}}/\sqrt{|\mathcal{M}_{\text{FO}}|}$, while the decryption failure term scales as $4q_{\text{G}}\sqrt{\delta_{\text{dec}}}$. These terms are separate from the IND-CPA term, which is instantiated by the LWE reductions in Theorem 4. Together, Theorems 5 and 6 state the CCA security of Frost.KEM in terms of the IND-CPA security of Frost.PKE, the decryption failure probability δ_{dec} , and the size of $\mathcal{M}_{\text{FO}}$.

6. Concrete Parameters and Evaluation

This section instantiates Frost.KEM at NIST security levels 1, 3, and 5. All parameter sets use power-of-two moduli with $p_{\text{pk}} \mid q$, $p_u \mid q$, and $p_v \mid q$. This choice makes modular reduction, quantization, and dither compensation implementable by shifts and masks, while preserving the exact split and normal form correspondence of Section 3. The resulting concrete security estimates are stated for unstructured normal form LWE instances with explicit bounded quantization error distributions.

6.1. Parameter Sets and Concrete Security

We instantiate Frost.KEM with $n = m$, $\ell = 8$, and $q = 2^{16}$ across all security levels. The FO plaintext length is $b_{\text{msg}}\ell^2 \in \{128, 192, 256\}$ bits. The long term secret $\mathbf{S}$ and the ephemeral secret $\mathbf{R}$ are sampled entrywise from centered binomial distributions $\mathcal{B}_{\eta_s}$ and $\mathcal{B}_{\eta_r}$, respectively. The communication components use compression moduli $p_{\text{pk}} = 2^{t_{\text{pk}}}$, $p_u = 2^{t_u}$, and $p_v = 2^{t_v}$. Only compact split coordinates are transmitted.

Parameter-selection rationale. The parameter sets were selected by a constrained search over power-of-two compression widths, dimensions, and centered-binomial secret widths. The search is governed by three coupled objectives: the normal-form LWE instances must meet the target Core-SVP costs, the whole ciphertext decryption-failure bound must remain below the target security level after the E_8 decoding union bound, and the transmitted public key and ciphertext sizes should be minimized. These objectives pull in different directions. More aggressive compression reduces bandwidth but increases the quantization error, which affects both the Lattice Estimator [42]² inputs and the DFR bound. Increasing the dimension or secret width improves the security margin but increases size and runtime, while the p_v width directly controls the correctness margin of the E_8 -coded message layer. The profiles in Table 2 are the smallest balanced points we found under these constraints, with the final reported security determined by the weakest of the public-key/U instance and the stacked ciphertext instance.

2. <https://github.com/malb/lattice-estimator>

The public key is $\text{pk} = (\rho, \mathbf{B} \in \mathbb{Z}_{p_{\text{pk}}}^{m \times \ell})$, where the 256-bit seed ρ expands to $\mathbf{A}$ and $\mathbf{D}_{\text{pk}}$. The ciphertext is $\text{ct} = (\mathbf{U} \in \mathbb{Z}_{p_u}^{n \times \ell}, \mathbf{V} \in \mathbb{Z}_{p_v}^{\ell \times \ell}, \mu)$, where the 256-bit seed μ expands to $\mathbf{D}_u$ and $\mathbf{D}_v$. Thus

$$|\text{pk}| = 32 + \frac{m\ell t_{\text{pk}}}{8}, \quad |\text{ct}| = 32 + \frac{n\ell t_u}{8} + \frac{\ell^2 t_v}{8}. \quad (40)$$

The KEM secret key stores four objects, namely sk_{pke} , pk , h_{pk} , and z . Here $\text{sk}_{\text{pke}} = \mathbf{S}$ is packed using $\lceil \log_2(2\eta_s + 1) \rceil$ bits per coefficient, while h_{pk} and z each occupy 32 bytes. Consequently,

$$|\text{sk}| = |\text{pk}| + \frac{n\ell \lceil \log_2(2\eta_s + 1) \rceil}{8} + 64. \quad (41)$$

This gives the secret key sizes reported in Table 2.

Concrete hardness is evaluated through the normal form LWE instances induced by the IND-CPA proof. There are three instances. The public key instance is $\hat{\mathbf{B}} = \Delta_{\text{pk}}\mathbf{B} - \mathbf{D}_{\text{pk}} = \mathbf{A}\mathbf{S} + \mathbf{E}_{\text{pk}}$, where each entry of $\mathbf{E}_{\text{pk}}$ follows $\chi_{q, p_{\text{pk}}}$. The U instance is $\hat{\mathbf{U}} = \Delta_u\mathbf{U} - \mathbf{D}_u = \mathbf{A}^\top \mathbf{R} + \mathbf{E}_u$, where each entry of $\mathbf{E}_u$ follows χ_{q, p_u} . The joint ciphertext instance is the stacked instance used in the $(\mathbf{U}, \mathbf{V})$ replacement:

$$\begin{pmatrix} \hat{\mathbf{U}} \\ \hat{\mathbf{V}} - \text{Encode}(\mathbf{M}) \end{pmatrix} = \begin{pmatrix} \mathbf{A}^\top \\ \hat{\mathbf{B}}^\top \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_u \\ \mathbf{E}_v \end{pmatrix}. \quad (42)$$

The reported security is the minimum over the public key, U, and stacked $(\mathbf{U}, \mathbf{V})$ instances. Since our parameter sets have $n = m$ and $t_{\text{pk}} = t_u$, the public key and U estimates coincide. Table 2 reports the resulting classical and quantum Core-SVP costs, with detailed estimates in Appendix B.1. For the public-key and U-side instances, the estimator input uses the native bounded uniform distribution matching $\chi_{q, p}$, namely $\mathcal{U}(\{-\Delta/2 + 1, \dots, \Delta/2\})$. The stacked $(\mathbf{U}, \mathbf{V})$ ciphertext instance audits the joint replacement used in the IND-CPA proof. For this audit, the estimator receives a single-noise summary whose variance equals the row average of the two quantization-error widths:

$$\sigma_{\text{stack}}^2 = \frac{n \text{Var}(\chi_{q, p_u}) + \ell \text{Var}(\chi_{q, p_v})}{n + \ell}. \quad (43)$$

In all parameter sets, this stacked audit gives a larger attack cost than the public-key/U estimate, so the reported security is determined by the native-uniform public-key/U instance.

The decryption failure bounds were computed by the script `Frost_E8_DFREst_Chernoff.m`. The script constructs exact finite support probability mass functions for the centered binomial secrets, the quantization errors, and the product terms appearing in

$$\mathbf{e}_j = (\mathbf{E}_{\text{pk}}^\top \mathbf{R})_j - (\mathbf{S}^\top \mathbf{E}_u)_j + (\mathbf{E}_v)_j. \quad (44)$$

For each E_8 block, it groups the 240 roots into symmetry classes, evaluates the cumulant generating function of each one dimensional projection exactly from the finite support distributions, and applies Chernoff's method to the two tails. This projection-level computation avoids an independence assumption among the coordinates inside an E_8 block, a point that is important in DFR estimates for lattice-coded

TABLE 2. PARAMETER SETS FOR Frost.KEM. HERE b_{msg} IS THE AVERAGE NUMBER OF MESSAGE BITS PER COORDINATE IN EACH 8-DIMENSIONAL MESSAGE BLOCK, AND $\mathcal{D}_x$ IS THE PUBLIC DITHER SPACE FOR $x \in \{\text{pk}, \text{u}, \text{v}\}$. SIZES ARE IN BYTES AND INCLUDE TRANSMITTED OR STORED SEEDS. THE COLUMN $\log_2 \delta$ REPORTS THE WHOLE CIPHERTEXT E_8 VORONOI FACET CHERNOFF BOUND. SEC. C AND SEC. Q REPORT THE SELECTED CLASSICAL AND QUANTUM CORE-SVP COSTS FOLLOWING [43].

Profile	n	m	ℓ	b_{msg}	q	(η_s, η_r)	t_{pk}	t_u	t_v	$\mathcal{D}_{\text{pk}}$	$\mathcal{D}_u$	$\mathcal{D}_v$	$ \text{pk} $	$ \text{ct} $	$ \text{sk} $	$\log_2 \delta$	Sec. C/Q
Frost-128	576	576	8	2	2^{16}	(3, 3)	10	10	8	$\mathbb{Z}_{26}^{m \times \ell}$	$\mathbb{Z}_{26}^{n \times \ell}$	$\mathbb{Z}_{28}^{\ell \times \ell}$	5792	5856	7584	-143.45	128.56/116.93
Frost-192	968	968	8	3	2^{16}	(1, 1)	11	11	7	$\mathbb{Z}_{25}^{m \times \ell}$	$\mathbb{Z}_{25}^{n \times \ell}$	$\mathbb{Z}_{29}^{\ell \times \ell}$	10680	10736	12680	-233.57	195.91/179.51
Frost-256	1320	1320	8	4	2^{16}	(2, 2)	13	13	8	$\mathbb{Z}_{23}^{m \times \ell}$	$\mathbb{Z}_{23}^{n \times \ell}$	$\mathbb{Z}_{28}^{\ell \times \ell}$	17192	17256	21216	-289.51	256.67/235.06

schemes [44]. It then applies a union bound over the root facets and over the $\ell = 8$ ciphertext columns. The reported $\log_2 \delta$ values are the resulting whole ciphertext upper bounds. The only numerical approximation in this computation is the floating point optimization used to minimize the Chernoff exponent.

6.2. Implementation and Performance Evaluation

We evaluate a portable reference implementation and an AVX2 optimized implementation of Frost.KEM. The public matrix expansion uses AES-128 by default, and the dense matrix kernels follow the multiplication strategies for FrodoKEM in [45]. All benchmarks were run on an Intel Core i7-10700K CPU under Ubuntu 20.04.6 LTS, compiled with gcc 9.4 using `-O3 -march=native`. Hyper threading and turbo boost were disabled, and the processor frequency was fixed at 3.80 GHz. Each reported cycle count is the median over 1000 executions. Table 3 compares Frost with representative LWE and LWR based KEMs under the same local measurement protocol. ML-KEM is the standardized practical reference point for lattice KEMs. FrodoKEM, the non-ring Round5 variants, and SCloud⁺ provide more direct comparisons of unstructured-matrix arithmetic and the associated LWE or LWR assumptions. Thus ML-KEM supplies practical context, while these plain-matrix baselines expose the structural tradeoffs relevant to Frost. For Round5, the Ref. column denotes its optimized portable non-AVX2 implementation.

The comparison highlights the bandwidth and runtime profile obtained from dithered quantization over unstructured matrices. Compared with FrodoKEM, Frost reduces both public key and ciphertext sizes at all three security levels. At level 1, the combined public key and ciphertext size is 11648 bytes for Frost-128, compared with 19368 bytes for FrodoKEM-640. The reduction is about 40%. Compared with SCloud⁺, Frost has a smaller combined bandwidth footprint at levels 1 and 3, and a comparable footprint at level 5, while using an explicit input independent quantization error distribution.

Frost benefits from its $q = 2^{16}$ design, which aligns naturally with native 16-bit arithmetic. Quantization, compensation, and modular reduction use constant-time shifts, masks, and additions. Relative to FrodoKEM, Frost also avoids discrete-Gaussian sampling. In the reference implementation, Frost encapsulation is 9.9 $\times$, 6.7 $\times$, and 9.4 $\times$ faster than FrodoKEM at levels 1, 3, and 5, respectively. Reference

decapsulation is 8.8 $\times$, 6.1 $\times$, and 8.7 $\times$ faster. Under AVX2, Frost is faster than the measured unstructured LWE baselines on all three KEM operations.

Round5 presents a different tradeoff [11]. Frost is faster in the reported level-1 AVX2 encapsulation and decapsulation measurements, while Round5 is faster in all three AVX2 operations at levels 3 and 5. At these higher levels, Round5 uses smaller main dimensions: 876 versus 968, and 1217 versus 1320. Round5 also avoids Frost’s public-dither processing, and its fixed-weight ternary secrets permit sparse-index arithmetic in the optimized portable path. The compared AVX2 configuration selects a constant-time arithmetic path that does not exploit the same sparse-index operations. The additional work in that path can offset SIMD gains, so its encapsulation and decapsulation timings need not improve over the optimized portable implementation. The observed comparison therefore reflects matrix dimension, dither processing, and the arithmetic path selected by each implementation.

7. Conclusion and Future Work

Starting from the general dithered LWQ–LWE equivalence established in [16], we specialized quantization to the integer product lattice with aligned power-of-two moduli. This specialization gives the exact finite-support distribution $\chi_{q,p}$ and an explicit reversible map between split transcripts and additive-noise normal-form samples. These ingredients connect Frost’s transmitted representation to the LWE instances in its PKE security proof and support the Fujisaki–Okamoto construction of a CCA-secure KEM. The E_8 -coded message embedding improves the correctness margin, with whole-ciphertext decryption-failure bounds derived from its Voronoi facets.

We parameterized and implemented Frost at NIST security levels 1, 3, and 5. The $q = 2^{16}$ design aligns with native 16-bit arithmetic and implements quantization, compensation, and modular reduction with shifts, masks, and additions. The evaluation reports bandwidth savings over FrodoKEM and lower AVX2 runtimes than the measured unstructured LWE baselines. The Round5 comparison shows how smaller dimensions and sparse arithmetic trade off against Frost’s public-dither processing.

Several directions remain open. First, LWQ can be instantiated with quantization lattices beyond $\mathbb{Z}^m$. Changing the quantization lattice changes the induced error distribution, so each such choice requires a matching security analysis,

TABLE 3. COMPARISON OF Frost with REPRESENTATIVE LWE and LWR BASED KEMs AT NIST SECURITY LEVELS 1, 3, AND 5. DFR DENOTES $\log_2 \delta_{\text{dec}}$ UNDER THE WHOLE CIPHERTEXT FAILURE CONVENTION USED IN THIS WORK. EFFICIENCY IS REPORTED IN KCYCLES.

Scheme	Type	q	Size (B)			DFR	Ref. kCycles			AVX2 kCycles		
			pk	ct	sk		KG	Enc	Dec	KG	Enc	Dec
Security Level 1												
Frost-128 ¹	LWQ	2 ¹⁶	5792	5856	7584	−143	571	764	863	483	690	807
FrodoKEM-640 ² [6]	LWE	2 ¹⁵	9616	9752	19888	−138	1493	7583	7629	1123	1271	1193
SCLoud ⁺ -128 ³ [8]	LWE	2 ¹²	7200	5456	–	−134	1177	1215	1167	999	1062	1014
R5N1-1CCA-0d ⁴ [11]	LWR	2 ¹²	5740	5788	5772	–	992	446	556	470	770	857
CCALizard-128 ⁵ [13]	LWE+LWR	2 ¹¹	524320	955	34304	−154	63042	50	76	41113	33	56
Security Level 3												
Frost-192 ¹	LWQ	2 ¹⁶	10680	10736	12680	−234	1634	1904	2076	1267	1704	1917
FrodoKEM-976 ² [6]	LWE	2 ¹⁶	15632	15792	31296	−199	2945	12823	12735	2290	2396	2255
SCLoud ⁺ -192 ³ [8]	LWE	2 ¹²	11136	10832	–	−201	2369	2463	2384	2008	2040	1925
R5N1-3CCA-0d ⁴ [11]	LWR	2 ¹⁵	9660	9716	9708	–	4505	1277	1693	902	1389	1677
Security Level 5												
Frost-256 ¹	LWQ	2 ¹⁶	17192	17256	21216	−290	2796	3157	3452	2145	2832	3106
FrodoKEM-1344 ² [6]	LWE	2 ¹⁶	21520	21632	43088	−252	5358	29818	29988	4603	4551	3909
SCLoud ⁺ -256 ³ [8]	LWE	2 ¹²	18744	16916	–	−266	3826	4236	4225	2953	3248	3150
R5N1-5CCA-0d ⁴ [11]	LWR	2 ¹⁵	14636	14708	14700	–	6123	1467	1849	1421	2649	2256

All implementations in this table were benchmarked locally on an Intel Core i7-10700K CPU at 3.80 GHz with AVX2 enabled under Ubuntu 20.04.6 LTS. Hyper threading and turbo boost were disabled. All local builds used gcc 9.4 with `-O3 -march=native`. Cycle counts are medians over 1000 executions.

¹ Frost, paper parameter implementation measured using the default AES-128 public matrix backend.

² FrodoKEM, branch `main`, commit `6f9adea`.

³ SCLoud⁺, branch `main`, commit `123fe28`; the public implementation reports public-key and ciphertext sizes in the compared format.

⁴ Round5, branch `master`, commit `35e97f0`; the whole-ciphertext DFR value under this work’s convention is unavailable in the compared implementation.

⁵ CCALizard, branch `master`, commit `43e30f7`. The implementation exposes 128-bit parameter variants.

concrete estimator model, and constant-time implementation strategy. Second, dithered quantization may be useful in ring or module settings, where the main question is whether the input-independent error view can be preserved while exploiting algebraic structure for efficiency. Third, LWQ may be useful beyond encryption and key encapsulation. Its input-independent quantization interface could support other lattice-based primitives, such as signatures, commitments, and zero-knowledge proofs, especially in ring or module variants. Identifying which primitives can exploit this interface while retaining simple security reductions is an open direction.

References

- [1] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” in *37th Annual ACM Symposium on Theory of Computing*, H. N. Gabow and R. Fagin, Eds. Baltimore, MA, USA: ACM Press, May 22–24, 2005, pp. 84–93.
- [2] C. Peikert, “Public-key cryptosystems from the worst-case shortest vector problem: extended abstract,” in *41st Annual ACM Symposium on Theory of Computing*, M. Mitzenmacher, Ed. Bethesda, MD, USA: ACM Press, May 31 – Jun. 2, 2009, pp. 333–342.
- [3] V. Lyubashevsky, C. Peikert, and O. Regev, “On ideal lattices and learning with errors over rings,” in *Advances in Cryptology – EUROCRYPT 2010*, ser. Lecture Notes in Computer Science, H. Gilbert, Ed., vol. 6110. French Riviera: Springer Berlin Heidelberg, Germany, May 30 – Jun. 3, 2010, pp. 1–23.
- [4] J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, “CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM,” in *2018 IEEE European Symposium on Security and Privacy*. London, United Kingdom: IEEE Computer Society Press, Apr. 24–26, 2018, pp. 353–367.
- [5] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, “CRYSTALS-Dilithium: A lattice-based digital signature scheme,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 1, pp. 238–268, 2018. [Online]. Available: <https://tches.iacr.org/index.php/TCHES/article/view/839>
- [6] J. W. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila, “Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE,” in *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. Vienna, Austria: ACM Press, Oct. 24–28, 2016, pp. 1006–1018.
- [7] Z. Zheng, A. Wang, H. Fan, C. Zhao, C. Liu, and X. Zhang, “SCLoud: Public key encryption and key encapsulation mechanism based on learning with errors,” *Cryptology ePrint Archive*, Report 2020/095, 2020. [Online]. Available: <https://eprint.iacr.org/2020/095>
- [8] A. Wang, Z. Zheng, C. Zhao, Z. Qiu, G. Zeng, Y. Yuan, C. Mu, and X. Wang, “SCLoud+: An efficient LWE-based KEM without ring/module structure,” in *International Conference on Research in Security Standardisation*. Springer, 2024, pp. 147–174.
- [9] A. Banerjee, C. Peikert, and A. Rosen, “Pseudorandom functions and lattices,” in *Advances in Cryptology – EUROCRYPT 2012*, ser. Lecture Notes in Computer Science, D. Pointcheval and T. Johansson, Eds., vol. 7237. Cambridge, UK: Springer Berlin Heidelberg, Germany, Apr. 15–19, 2012, pp. 719–737.
- [10] J.-P. D’Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren, “Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM,” in *AFRICACRYPT 18: 10th International Conference on Cryptology in Africa*, ser. Lecture Notes in Computer Science, A. Joux, A. Nitaj, and T. Rachidi, Eds., vol. 10831. Marrakesh, Morocco: Springer, Cham, Switzerland, May 7–9, 2018, pp. 282–305.
- [11] H. Baan, S. Bhattacharya, S. R. Fluhrer, Ó. García-Morchón, T. Laarhoven, R. Rietman, M.-J. O. Saarinen, L. Tolhuizen, and Z. Zhang, “Round5: Compact and fast post-quantum public-key encryption,” in *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, J. Ding and R. Steinwandt, Eds. Chongqing, China: Springer, Cham, Switzerland, May 8–10, 2019, pp. 83–102.

- [12] S. Park, C.-G. Jung, A. Park, J. Choi, and H. Kang, "TiGER: Tiny bandwidth key encapsulation mechanism for easy miGRation based on RLWE(R)," Cryptology ePrint Archive, Report 2022/1651, 2022. [Online]. Available: <https://eprint.iacr.org/2022/1651>
- [13] J. H. Cheon, D. Kim, J. Lee, and Y. Song, "Lizard: Cut off the tail! A practical post-quantum public-key encryption from LWE and LWR," in *SCN 18: 11th International Conference on Security in Communication Networks*, ser. Lecture Notes in Computer Science, D. Catalano and R. De Prisco, Eds., vol. 11035. Amalfi, Italy: Springer, Cham, Switzerland, Sep. 5–7, 2018, pp. 160–177.
- [14] J. Alwen, S. Krenn, K. Pietrzak, and D. Wichs, "Learning with rounding, revisited - new reduction, properties and applications," in *Advances in Cryptology – CRYPTO 2013, Part I*, ser. Lecture Notes in Computer Science, R. Canetti and J. A. Garay, Eds., vol. 8042. Santa Barbara, CA, USA: Springer Berlin Heidelberg, Germany, Aug. 18–22, 2013, pp. 57–74.
- [15] P. Newton and S. Richelson, "A lower bound for proving hardness of learning with rounding with polynomial modulus," in *Advances in Cryptology – CRYPTO 2023, Part V*, ser. Lecture Notes in Computer Science, H. Handschuh and A. Lysyanskaya, Eds., vol. 14085. Santa Barbara, CA, USA: Springer, Cham, Switzerland, Aug. 20–24, 2023, pp. 805–835.
- [16] S. Lyu, L. Liu, and C. Ling, "Learning with quantization: Construction, hardness, and applications," Cryptology ePrint Archive, Report 2024/714, 2024. [Online]. Available: <https://eprint.iacr.org/2024/714>
- [17] R. Zamir, *Lattice coding for signals and networks: A structured coding approach to quantization, modulation, and multiuser information theory*. Cambridge University Press, 2014.
- [18] C. Peikert, "Lattice cryptography for the internet," in *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014*, M. Mosca, Ed. Waterloo, Ontario, Canada: Springer, Cham, Switzerland, Oct. 1–3, 2014, pp. 197–219.
- [19] J. H. Cheon, A. Kim, M. Kim, and Y. S. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology – ASIACRYPT 2017, Part I*, ser. Lecture Notes in Computer Science, T. Takagi and T. Peyrin, Eds., vol. 10624. Hong Kong, China: Springer, Cham, Switzerland, Dec. 3–7, 2017, pp. 409–437.
- [20] S. Liu and A. Sakzad, "Crystals-kyber with lattice quantizer," in *2024 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2024, pp. 2886–2891.
- [21] R. G. McKilliam, W. D. Smith, and I. V. L. Clarkson, "Linear-time nearest point algorithms for Coxeter lattices," *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1015–1022, 2010.
- [22] S. Lyu, L. Liu, C. Ling, J. Lai, and H. Chen, "Lattice codes for lattice-based PKE," *Designs, Codes and Cryptography*, vol. 92, no. 4, pp. 917–939, 2024.
- [23] S. Liu and A. Sakzad, "Lattice codes for CRYSTALS-kyber," *Designs, Codes and Cryptography*, vol. 93, no. 8, pp. 3181–3205, 2025.
- [24] S. Liu and A. Sakzad, "Compact lattice-coded (multi-recipient) kyber without CLT independence assumption," in *Advances in Cryptology – ASIACRYPT 2025, Part III*, ser. Lecture Notes in Computer Science, G. Hanaoka and B.-Y. Yang, Eds., vol. 16247. Melbourne, VIC, Australia: Springer, Singapore, Singapore, Dec. 8–12, 2025, pp. 363–395.
- [25] D. Micciancio and C. Peikert, "Hardness of SIS and LWE with small parameters," in *Advances in Cryptology – CRYPTO 2013, Part I*, ser. Lecture Notes in Computer Science, R. Canetti and J. A. Garay, Eds., vol. 8042. Santa Barbara, CA, USA: Springer Berlin Heidelberg, Germany, Aug. 18–22, 2013, pp. 21–39.
- [26] C. Sun, M. Tibouchi, and M. Abe, "Revisiting the hardness of binary error LWE," in *ACISP 20: 25th Australasian Conference on Information Security and Privacy*, ser. Lecture Notes in Computer Science, J. K. Liu and H. Cui, Eds., vol. 12248. Perth, WA, Australia: Springer, Cham, Switzerland, Nov. 30 – Dec. 2, 2020, pp. 425–444.
- [27] N. Döttling and J. Müller-Quade, "Lossy codes and a new variant of the learning-with-errors problem," in *Advances in Cryptology – EUROCRYPT 2013*, ser. Lecture Notes in Computer Science, T. Johansson and P. Q. Nguyen, Eds., vol. 7881. Athens, Greece: Springer Berlin Heidelberg, Germany, May 26–30, 2013, pp. 18–34.
- [28] Z. Brakerski and N. Döttling, "Hardness of LWE on general entropic distributions," in *Advances in Cryptology – EUROCRYPT 2020, Part II*, ser. Lecture Notes in Computer Science, A. Canteaut and Y. Ishai, Eds., vol. 12106. Zagreb, Croatia: Springer, Cham, Switzerland, May 10–14, 2020, pp. 551–575.
- [29] Z. Brakerski and N. Döttling, "Lossiness and entropic hardness for ring-LWE," in *TCC 2020: 18th Theory of Cryptography Conference, Part I*, ser. Lecture Notes in Computer Science, R. Pass and K. Pietrzak, Eds., vol. 12550. Durham, NC, USA: Springer, Cham, Switzerland, Nov. 16–19, 2020, pp. 1–27.
- [30] H. Lin, M. Wang, J. Zhuang, and Y. Wang, "Hardness of module-LWE and ring-LWE on general entropic distributions," Cryptology ePrint Archive, Report 2020/1238, 2020. [Online]. Available: <https://eprint.iacr.org/2020/1238>
- [31] K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen, "Entropic hardness of module-LWE from module-NTRU," in *Progress in Cryptology - INDOCRYPT 2022: 23rd International Conference in Cryptology in India*, ser. Lecture Notes in Computer Science, T. Isobe and S. Sarkar, Eds., vol. 13774. Kolkata, India: Springer, Cham, Switzerland, Dec. 11–14, 2022, pp. 78–99.
- [32] K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen, "On the hardness of module learning with errors with short distributions," *Journal of Cryptology*, vol. 36, no. 1, p. 1, Jan. 2023.
- [33] K. Boudgoust, C. Jeudy, E. Tairi, and W. Wen, "Hardness of M-LWE with general distributions and applications to leaky variants," in *IACR International Conference on Public-Key Cryptography*. Springer, 2026, pp. 3–37.
- [34] L. Ducas, L. Engelberts, and P. de Perthuis, "Predicting module-lattice reduction," in *Advances in Cryptology – ASIACRYPT 2025, Part III*, ser. Lecture Notes in Computer Science, G. Hanaoka and B.-Y. Yang, Eds., vol. 16247. Melbourne, VIC, Australia: Springer, Singapore, Singapore, Dec. 8–12, 2025, pp. 133–166.
- [35] P. de Perthuis and F. Trenkić, "Refined modelling of the primal attack, and variants against module-learning with errors," Cryptology ePrint Archive, Report 2025/2195, 2025. [Online]. Available: <https://eprint.iacr.org/2025/2195>
- [36] E. Wenger, E. Saxena, M. Malhou, E. Thieu, and K. E. Lauter, "Benchmarking attacks on learning with errors," in *2025 IEEE Symposium on Security and Privacy*, M. Blanton, W. Enck, and C. Nita-Rotaru, Eds. San Francisco, CA, USA: IEEE Computer Society Press, May 12–15, 2025, pp. 279–297.
- [37] A. Pouly and Y. Shen, "Provable dual attacks on learning with errors," in *Advances in Cryptology – EUROCRYPT 2024, Part VII*, ser. Lecture Notes in Computer Science, M. Joye and G. Leander, Eds., vol. 14657. Zurich, Switzerland: Springer, Cham, Switzerland, May 26–30, 2024, pp. 256–285.
- [38] L. Ducas and L. N. Pulles, "Does the dual-sieve attack on learning with errors even work?" in *Advances in Cryptology – CRYPTO 2023, Part III*, ser. Lecture Notes in Computer Science, H. Handschuh and A. Lysyanskaya, Eds., vol. 14083. Santa Barbara, CA, USA: Springer, Cham, Switzerland, Aug. 20–24, 2023, pp. 37–69.
- [39] D. Hofheinz, K. Hövelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," in *TCC 2017: 15th Theory of Cryptography Conference, Part I*, ser. Lecture Notes in Computer Science, Y. Kalai and L. Reyzin, Eds., vol. 10677. Baltimore, MD, USA: Springer, Cham, Switzerland, Nov. 12–15, 2017, pp. 341–371.
- [40] H. Jiang, Z. Zhang, L. Chen, H. Wang, and Z. Ma, "IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited," in *Advances in Cryptology – CRYPTO 2018, Part III*, ser. Lecture Notes in Computer Science, H. Shacham and A. Boldyreva, Eds., vol. 10993. Santa Barbara, CA, USA: Springer, Cham, Switzerland, Aug. 19–23, 2018, pp. 96–125.

- [41] H. Jiang, Z. Zhang, and Z. Ma, “Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model,” in *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, J. Ding and R. Steinwandt, Eds. Chongqing, China: Springer, Cham, Switzerland, May 8–10, 2019, pp. 227–248.
- [42] M. R. Albrecht, R. Player, and S. Scott, “On the concrete hardness of learning with errors,” *Journal of Mathematical Cryptology*, vol. 9, no. 3, pp. 169–203, 2015.
- [43] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, “Post-quantum key exchange - A new hope,” in *USENIX Security 2016: 25th USENIX Security Symposium*, T. Holz and S. Savage, Eds. Austin, TX, USA: USENIX Association, Aug. 10–12, 2016, pp. 327–343. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim>
- [44] J.-P. D’Anvers, F. Vercauteren, and I. Verbauwhede, “The impact of error dependencies on ring/mod-LWE/LWR based schemes,” in *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, J. Ding and R. Steinwandt, Eds. Chongqing, China: Springer, Cham, Switzerland, May 8–10, 2019, pp. 103–115.
- [45] J. W. Bos, M. Ofner, J. Renes, T. Schneider, and C. van Vredendaal, “The matrix reloaded: Multiplication strategies in FrodoKEM,” in *CANS 21: 20th International Conference on Cryptology and Network Security*, ser. Lecture Notes in Computer Science, M. Conti, M. Stevens, and S. Krenn, Eds., vol. 13099. Vienna, Austria: Springer, Cham, Switzerland, Dec. 13–15, 2021, pp. 72–91.

Appendix A. Rectangular-Form Labeling for the E_8 -Coded Message Space

This appendix specifies the deterministic label and inverse-label maps for the E_8 -coded message layer. The maps give a public bijection between one column payload and the hypercube-shaped code $\mathcal{C}_{E_8} = \alpha_{E_8} E_8 \cap [0, q]^8$. We use the rectangular-form labeling method for hypercube-shaped lattice codes [22]. Let

$$\mathbf{U}_{E_8} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \in \text{GL}_8(\mathbb{Z}) \quad (45)$$

and define

$$\mathbf{B}_{E_8} = \mathbf{U}_{E_8} \text{diag} \left(2, 1, 1, 1, 1, 1, 1, \frac{1}{2} \right). \quad (46)$$

This is a rectangular-form basis of E_8 , i.e., $E_8 = \mathcal{L}(\mathbf{B}_{E_8})$. For $b_{\text{msg}} \in \{2, 3, 4\}$, set $\alpha_{E_8} \triangleq q/2^{b_{\text{msg}}}$, $\Lambda_f \triangleq \alpha_{E_8} E_8$, and $\Lambda_c \triangleq q\mathbb{Z}^8$. Let $\mathbf{B}_f \triangleq \alpha_{E_8} \mathbf{B}_{E_8}$. Then $\Lambda_f = \mathcal{L}(\mathbf{B}_f)$ and

$$\mathbf{B}_f = \mathbf{U}_{E_8} \text{diag} (q/P_1, \dots, q/P_8), \quad (47)$$

where $P_1 \triangleq 2^{b_{\text{msg}}-1}$, $P_i \triangleq 2^{b_{\text{msg}}}$ for $2 \leq i \leq 7$, and $P_8 \triangleq 2^{b_{\text{msg}}+1}$. The corresponding coarse lattice relation is

$$\mathcal{L}(\mathbf{B}_f \text{diag}(P_1, \dots, P_8)) = \mathcal{L}(q\mathbf{U}_{E_8}) = q\mathbb{Z}^8 = \Lambda_c. \quad (48)$$

Thus the mixed-radix payload layout is

$$\beta(b_{\text{msg}}) = (b_{\text{msg}} - 1, b_{\text{msg}}, b_{\text{msg}}, b_{\text{msg}}, b_{\text{msg}}, b_{\text{msg}}, b_{\text{msg}}, b_{\text{msg}} + 1). \quad (49)$$

For the three profiles this gives

b_{msg}	$\beta(b_{\text{msg}})$
2	(1, 2, 2, 2, 2, 2, 2, 3)
3	(2, 3, 3, 3, 3, 3, 3, 4)
4	(3, 4, 4, 4, 4, 4, 4, 5)

(50)

For one column payload, the bit mapper parses the input according to this layout and obtains

$$\mathbf{z} = (z_1, \dots, z_8) \in \mathcal{I}_{E_8} \triangleq \prod_{i=1}^8 \{0, \dots, P_i - 1\}. \quad (51)$$

The label map is defined by $\text{Label}_{E_8}(\mathbf{z}) \triangleq [\mathbf{B}_f \mathbf{z}]_q \in \mathcal{C}_{E_8}$, where $[\cdot]_q$ denotes coordinate-wise reduction to $[0, q)$. The encoder applies Label_{E_8} to each column payload and places the resulting vectors as the columns of $\text{Encode}(\mathbf{M})$.

For $\mathbf{c} \in \mathcal{C}_{E_8}$, the inverse-label map is $\text{Delabel}_{E_8}(\mathbf{c}) \triangleq \mathbf{B}_f^{-1} \mathbf{c} \bmod (P_1, \dots, P_8)$, where the reduction is coordinate-wise. Since $\mathbf{c} \in \Lambda_f$, the vector $\mathbf{B}_f^{-1} \mathbf{c}$ is integral before the mixed-radix reduction. The bit demapper then writes each recovered z_i using exactly $\beta_i(b_{\text{msg}})$ bits.

Lemma 5. *The map $\text{Label}_{E_8} : \mathcal{I}_{E_8} \rightarrow \mathcal{C}_{E_8}$ is a bijection, and its inverse is Delabel_{E_8} .*

Proof. The basis $\mathbf{B}_f$ has rectangular form $\mathbf{B}_f = \mathbf{U}_{E_8} \text{diag}(q/P_1, \dots, q/P_8)$, and the coarse lattice satisfies $\Lambda_c = \mathcal{L}(\mathbf{B}_f \text{diag}(P_1, \dots, P_8)) = q\mathbb{Z}^8$. The rectangular-form labeling theorem for hypercube-shaped lattice codes [22] therefore gives that $\mathbf{z} \mapsto [\mathbf{B}_f \mathbf{z}]_q$ is a bijection from $\prod_i \{0, \dots, P_i - 1\}$ to $\mathcal{C}(\Lambda_f, \Lambda_c)$. The inverse is obtained by multiplying by $\mathbf{B}_f^{-1}$ and reducing the i -th coordinate modulo P_i . $\square$

Appendix B. Implementation Specification and Performance Analysis

B.1. Concrete Security Estimates

Table 4 reports the concrete lattice-estimator attack costs used in Table 2. The normal-form LWE instances follow the split-normal correspondence in Section 3; since $n = m$ and $t_{\text{pk}} = t_u$, the public-key and U-side instances have identical hardness profiles.

The estimates use lattice-estimator commit 27a581b. The artifact notebook `FrostUniSecEst_CQ.ipynb` runs with `WIDTH_MODEL=variance`, enables the public-key, U-side, and stacked row-average surfaces, and prints the estimator results underlying Table 4.

B.2. Symmetric Primitives, Domain Separation, and Sampling

The public matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is expanded with AES-128 by default, with a portable SHAKE128 backend. Dither

TABLE 4. CONCRETE SECURITY ESTIMATES ($\log_2 \text{rop}$) FOR Frost.KEM UNDER THE CORE-SVP MODELS [43]. THE REPORTED SECURITY IS THE MINIMUM OVER THE PUBLIC-KEY/ $\mathbf{U}$ INSTANCE AND THE STACKED CIPHERTEXT INSTANCE INDUCED BY THE JOINT $(\mathbf{U}, \mathbf{V})$ REPLACEMENT IN THE IND-CPA PROOF. THE STACKED CIPHERTEXT ROWS USE THE ROW-AVERAGED-VARIANCE SINGLE-NOISE ESTIMATE DESCRIBED IN SECTION 6.1.

Scheme	Model	Instance	BKW	uSVP	BDD	Dual	Dual-Hybrid	Reported Sec.
Frost-128	Classical	Public Key / $\mathbf{U}$	194.51	131.69	132.55	137.24	128.56	128.56
		Stacked $(\mathbf{U}, \mathbf{V})$	200.26	133.44	134.31	139.28	130.09	
	Quantum	Public Key / $\mathbf{U}$	194.51	119.52	120.39	124.55	116.93	116.93
		Stacked $(\mathbf{U}, \mathbf{V})$	200.26	121.11	121.98	126.41	118.48	
Frost-192	Classical	Public Key / $\mathbf{U}$	257.99	207.61	208.61	214.91	195.91	195.91
		Stacked $(\mathbf{U}, \mathbf{V})$	271.76	221.92	222.92	229.80	208.78	
	Quantum	Public Key / $\mathbf{U}$	257.99	188.42	189.42	195.04	179.51	179.51
		Stacked $(\mathbf{U}, \mathbf{V})$	271.76	201.40	202.40	208.56	191.60	
Frost-256	Classical	Public Key / $\mathbf{U}$	369.18	269.22	270.22	277.11	256.67	256.67
		Stacked $(\mathbf{U}, \mathbf{V})$	378.32	299.30	300.16	308.35	283.68	
	Quantum	Public Key / $\mathbf{U}$	369.18	244.33	245.33	251.49	235.06	235.06
		Stacked $(\mathbf{U}, \mathbf{V})$	378.32	271.62	272.50	279.84	260.23	

generation, secret sampling, FO coins, and transcript hashing use profile-local XOFs: SHAKE128 for Frost-128 and SHAKE256 for Frost-192/Frost-256. Fixed-width domain tags are absorbed into each seed stream. Because q , Δ_{pk} , Δ_u , and Δ_v are powers of two, matrix and dither parsing use bit-slicing and masking rather than rejection sampling.

B.3. Implementation Validation

Validation combines KAT generation, randomized KEM tests, and message-codec round trips $\text{Decode}(\text{Encode}(\mathbf{M})) = \mathbf{M}$ for all parameter sets. The tests cover reference and AVX2 implementations, AES-128 and SHAKE128 matrix backends, and both scalar and E_8 -coded message embeddings. The benchmark harness queries the compiled binary for the active message-codec path, tying each row to the executed implementation. Table 5 summarizes the reproduction entries for the source, scripts, raw logs, and checklist outputs.

B.4. Component-Level Profiling Breakdown

Table 6 reports median component-level cycle profiles from targeted instrumentation; component columns are non-additive relative to the full-flow measurements.

Matrix expansion and dense products account for over 85% of runtime, while dither generation, packing, formatting, and hashing remain nearly invariant. AVX2 gains at Frost-128 are mostly tied to symmetric-pipeline throughput; higher levels benefit increasingly from vectorized matrix arithmetic.

B.5. Constant-Time Considerations

The implementation uses source-level constant-time code for operations depending on secrets, decoded messages, or

decapsulation outcomes. Loop bounds in secret sampling, dense products, quantization, packing, message decoding, and FO re-encryption depend only on public parameters. Power-of-two moduli implement reduction, quantization, and dither compensation with additions, shifts, and masks, avoiding division, rejection sampling, and data-dependent reduction loops.

The E_8 -coded decoder runs a fixed nearest-neighbor procedure over D_8 and $D_8 + \frac{1}{2}\mathbf{1}$: it evaluates both coset candidates, computes integer squared distances, and selects with constant-time conditional moves. It uses no floating point, secret-indexed table lookups, or early termination. Decapsulation always performs decryption, coin derivation, re-encryption, ciphertext comparison, and final key derivation; the comparison bit is used only for the final constant-time key selection, so malformed ciphertexts follow the same path.

B.6. Algebraic Quantization Optimization

For $x \in \{\text{pk}, u, v\}$, let $p_x = 2^{t_x}$, $q = 2^{q_{\text{bits}}}$, and $\Delta_x = q/p_x = 2^{h_x}$. The compact coordinate $Q_{\Delta_x \mathbb{Z}, d}(\cdot)/\Delta_x$ and reconstruction $\Delta_x b - d$ are implemented as

$$Q_{\Delta_x \mathbb{Z}, d}(a)/\Delta_x \triangleq ((a + d + 2^{h_x-1}) \gg h_x) \bmod 2^{t_x}, \quad (52)$$

$$\Delta_x b - d \triangleq ((b \ll h_x) - d) \bmod 2^{q_{\text{bits}}}, \quad (53)$$

where $a \in [0, q]$, $d \in [0, \Delta_x]$, and $b \in [0, p_x]$. The code path uses fixed shifts, additions, subtractions, and masks, with no secret-dependent branches.

B.7. Matrix Arithmetic Optimization and Space Derivations

Dense products over $\mathbb{Z}_q$ use the streaming matrix-expansion strategy from optimized FrodoKEM multiplica-

TABLE 5. ARTIFACT REPRODUCTION MAP.

Artifact task	Reproduction entry	Output checked	Paper target
KAT and randomized tests	<code>make check; make test128/192/256</code>	KEM pass/fail logs	validation text
KAT files	<code>make KATS; run PQCTestKAT_kem</code>	<code>PQCKemKAT_*.rsp</code>	KAT material
KEM benchmarks	<code>./scripts/bench_levels_ref_avx2.sh ./bench.csv</code>	median kCycle CSV	Table 3
DFR script	<code>Frost_E8_DFREst_Chernoff('all')</code>	$\log_2 \delta_{\text{dec}}$ rows	Table 2
Estimator notebook	<code>run Sage notebook FrostUniSecEst_CQ.ipynb</code>	printed estimator results	Table 4

TABLE 6. FULL-FLOW AND COMPONENT-LEVEL EFFICIENCY PROFILES OF Frost.KEM WITH E_8 -CODED MESSAGE EMBEDDING ENABLED. ALL VALUES ARE REPORTED IN KCYCLES.

Scheme	Impl.	A-gen	Full Flow			KeyGen Components			Encaps Components			Decaps Components		
			KG	Enc	Dec	A-gen	AS	Aux.	A-gen	$A^\top R$	Aux.	A-gen	$A^\top R$	Aux.
Frost-128	Ref	AES	571	764	863	249	134	157	354	267	202	374	272	109
		SHAKE	4574	4750	4869	4412	170	148	4510	275	192	4526	279	112
	AVX2	AES	483	690	807	202	83	148	204	96	196	208	108	109
		SHAKE	1335	1535	1644	1062	69	126	1130	85	169	1005	85	97
Frost-192	Ref	AES	1634	1904	2076	588	590	278	887	647	376	897	646	211
		SHAKE	13049	13490	13563	11592	499	271	11625	604	378	13330	709	206
	AVX2	AES	1267	1704	1917	540	333	295	567	349	410	564	351	221
		SHAKE	3565	3957	4138	3268	352	298	3393	343	372	3539	342	217
Frost-256	Ref	AES	2796	3157	3452	1372	776	406	1291	1111	517	1487	1042	311
		SHAKE	23691	24237	24022	22104	1202	419	21926	1054	554	22258	1158	293
	AVX2	AES	2145	2832	3106	1009	580	498	1383	672	615	1319	620	318
		SHAKE	6569	7149	7355	6553	682	475	5626	646	612	6124	620	322

tion [45]. The reference code applies this layout to $\mathbf{AS}$, $\mathbf{A}^\top \mathbf{R}$, $\widehat{\mathbf{B}}^\top \mathbf{R}$, and $\mathbf{S}^\top \widehat{\mathbf{U}}$; the AVX2 code vectorizes the same inner products with SIMD additions and masks, avoiding Montgomery and Barrett reductions.

Serialization follows the bit widths implied by the parameter table. Secrets from $\mathcal{B}_{\eta_s}$ use $\lceil \log_2(2\eta_s + 1) \rceil$ bits per coefficient, giving packed secret matrices of 1728, 1936, and 3960 bytes. Together with 256-bit explicit seeds and hashes, these allocations produce the byte counts in Table 2.