When Algebraic Structure Meets Too Many Hints

Dongshu Cai[1,2], Yijian Liu[1,2], and Xianhui Lu[1,2](✉)

[1] Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China;
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{caidongshu, luxianhui}@iie.ac.cn

Abstract. The hint framework of Dachman-Soled, Ducas, Gong, and Rossi (CRYPTO 2020), together with the later too-many-hints analyses from May and Nowakowski (ASIACRYPT 2023) and Cao, Jiang, and Cheng (CRYPTO 2025), gives a classification of secret-dependent hints and LLL-breaking hint thresholds for LWE schemes. These values are now frequently adopted as the default benchmark for side-channel security evaluation. We show that applying them as a security metric is problematic.

DDGR categorizes hints based only on their local algebraic form, while the threshold analyses assume uniformly sampled hint vectors. Neither captures how the specific algebraic form of a hint governs the geometry of the resulting sparsified lattice. A hint's class label and the corresponding uniform threshold therefore cannot, on their own, capture its real cryptanalytic power.

We illustrate this mismatch with two forms of structured leakage that naturally appear in ring and module lattice implementations, validated via synthetic experiments on Kyber-512. NTT-domain residues qualify formally as mod-q hints, yet they recover the full Kyber-512 secret with roughly half as many hints as the uniform mod-q bound predicts. Finite-precision FFT samples fall into the weaker approximate-hint class, and still only need 25%–55% of the hint budget required for uniform perfect hints.

The gap stems from algebraic structure: these hint families carry additional ring structure that enables tailored lattice recovery strategies, a distinction invisible under the uniform-sampling model.

We do not propose new attack optimizations. Our findings indicate that DDGR hint classes and too-many-hints thresholds are syntactic descriptions, not standalone security metrics for real leakage traces. Evaluating a side-channel leak requires considering the algebraic structure of the hint vectors, not just their category and quantity.

Keywords: MLWE with Hints · Partial Key Exposure · NTT · FFT

1 Introduction

The Learning with Errors (LWE) problem, introduced by Regev at STOC 2005 [15], has played a central role in the development of post-quantum cryptography.

Building upon this foundation, the hardness of algebraically structured LWE variants and NTRU has been progressively established in [17,10,8,13,14], inspiring the design of a large body of practical cryptographic schemes based on RLWE, MLWE, and NTRU. More recently, NIST has standardized CRYSTALS-KYBER [2], CRYSTALS-DILITHIUM [6], and FALCON [16] as primary post-quantum cryptosystems. To reduce ciphertext size and improve computational performance, practical constructions inevitably rely on these algebraically structured hard problems.

From the perspective of hardness reduction, these structured computational problems correspond only to short-vector problems restricted to a special family of structured lattices. Unlike Regev's classical reduction for LWE, these problems do not capture the Short-Vector Problem (SVP) over arbitrary lattices. Traditionally, concrete security evaluations for schemes built upon structured lattices treated them as unstructured-lattice instances of equal dimension. Consequently, quantifying the security degradation introduced by algebraic structure has remained a prominent open question within lattice-based cryptanalysis. Very recently, two independent works by Ogilvie, and by Hou and Jiang, both at CRYPTO 2026 [7], demonstrated that, without external leakage, algebraic structure indeed lowers the concrete security of structured-lattice schemes compared with prior security estimates assuming generic unstructured lattices. In this work, we show from a side-channel perspective that the uniform-hint benchmark is unsound as a security metric: leaked information carrying algebraic structure can inflict security degradation far beyond what uniform-hint thresholds predict, and the gap is not a constant factor but a structural mismatch between the syntax of the hint taxonomy and the geometry of the lattice it sparsifies.

Dachman-Soled, Ducas, Gong, and Rossi at CRYPTO 2020 [5] introduced the first systematic framework for quantifying information leakage in LWE. For a secret vector $\boldsymbol{s}$ and given public vector $\boldsymbol{v}$, their work defined leakage of the exact inner product $\langle \boldsymbol{s}, \boldsymbol{v} \rangle$ as a perfect hint for $\boldsymbol{s}$, and leakage of $\langle \boldsymbol{s}, \boldsymbol{v} \rangle \bmod q$ as a mod-q hint for $\boldsymbol{s}$. Partial leakage of NTT-domain coordinates, for instance, is a mod-q hint under this taxonomy—an example DDGR themselves discuss.

May and Nowakowski at ASIACRYPT 2023 [11] investigated how many such hints are required to render lattice-based cryptosystems insecure when large numbers of these hints are available. Cao, Jiang, and Cheng at CRYPTO 2025 [4] advanced these results via a more unified lattice-construction technique.

Notably, their analyses assume fully random hints, i.e., the vectors $\boldsymbol{v}$ are sampled uniformly at random. This assumption, however, disregards possible intrinsic structure that hints may carry. We consider specific settings where an adversary can exploit the inherent algebraic structure originating from side-channel leakage to construct key-recovery algorithms more efficient than those expected under the generic random-hint setting. Accordingly, naively applying their conclusions to assess the security impact of concrete side-channel leakage would be ill-advised.

1.1 Our Contributions

Prior work establishes threshold hint counts at which lattice-based schemes become insecure: for Kyber-512, May and Nowakowski [11] show that full secret recovery via LLL requires 449 mod-q hints or 234 perfect hints, while Cao, Jiang, and Cheng [4] reduce the perfect-hint threshold to 230 with substantial runtime improvements. These numbers are increasingly used as benchmarks for side-channel security evaluation.

Our first contribution is a methodological one. We show that the DDGR taxonomy and the too-many-hints thresholds are syntactic descriptors of hint vectors, not security metrics for concrete leakage. The taxonomy labels each hint by its local algebraic form, and the threshold analyses assume that the hint vectors are uniformly sampled; neither captures how a given hint sparsifies the lattice containing the secret. Consequently, applying the uniform thresholds to a structured leakage instance can yield systematically optimistic security estimates.

We substantiate this claim with two structured leakage families that occur naturally when short ring elements are involved. Our recovery algorithm uses only the leaked hints and the corresponding hint vectors, without incorporating public-key samples, so the resulting hint counts isolate the cryptanalytic power of the hint structure itself.

NTT-domain leakage. We first consider mod-q hints. Prior works model the effect of each mod-q hint as reducing the dimension of the underlying LWE problem by one, an effect comparable to directly leaking one coordinate of the secret $\boldsymbol{s}$. Admittedly, this approach corresponds to a generic reduction that works for *arbitrary* mod-q hints. Our core observation is that the estimated number of hints required for key recovery differs drastically between uniformly-random hints and hints drawn from structured distributions. Furthermore, hints with intrinsic structure can enable entirely new exploitation strategies, even if they formally belong to the mod-q hint class.

Concretely, suppose several coordinates of $\boldsymbol{s}$ in the NTT domain are leaked. Within existing generic frameworks these are treated as ordinary mod-q hints; DDGR themselves discuss NTT-domain coordinates as an example of this hint class [5]. Nevertheless, our experiments show that for Kyber-512, 236 such hints already enable probabilistic full-secret recovery solely via LLL, using the same hint-only lattice construction as [11] but without incorporating public-key samples, and stable recovery is achieved with 240 hints. Moreover, for Kyber-family schemes where the NTT operates over an incomplete ring, these NTT-domain hints admit a natural subfield-aware algorithm for accelerated secret recovery. With this optimization, probabilistic key recovery is possible with merely 224 hints, and stable recovery is again attained at 240 hints. These numbers illustrate that structured mod-q hints can yield hint thresholds comparable to those of generic random perfect hints.

FFT-domain leakage. We next turn to approximate hints. A simple argument shows that in the ideal infinite-precision setting, a single FFT-domain coordinate $s(\zeta_i)$ already contains sufficient information to fully recover the secret

polynomial, and such leakage would count as two perfect hints (real and imaginary parts) under existing frameworks. In practice, however, FFT implementations only provide finite-precision measurements, and the quantized evaluations are properly classified as approximate hints under the DDGR taxonomy. We evaluate realistic 64-bit-precision FFT leakage under this weaker classification. Our experiments demonstrate that for Kyber-512, only 64 such approximate hints suffice for probabilistic full-secret recovery, and stable recovery is obtained with 128 hints. This threshold is substantially lower than the number required for generic random perfect hints—the strictly stronger category.

Runtime. We also provide a quantitative comparison between structured and unstructured hints in terms of runtime performance. Our results confirm that structured hints allow secret recovery using fewer hints and less computation time. This effect is especially pronounced when subfield reduction is applicable. For instance, for NTT-derived leakage, invoking BKZ with block size 30 over the subfield lattice achieves stable secret recovery within 90 seconds using only 204 hints. The high efficiency originates from the reduced lattice dimension handled by subfield-aware BKZ; in some cases BKZ on the subfield even outperforms plain LLL over the full-dimensional lattice. These runtime gains are not available to the uniform-hint attacks of [11,4], which operate on full-dimensional lattices.

1.2 Technical Overview

This section explains why the uniform-hint analysis and the structured-leakage reality diverge. We first recall why prior work treats each mod-q hint as one dimension reduction, then show that this treatment is not tight for structured hints.

First consider LWE samples $(\boldsymbol{a}, b = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e \bmod q)$, where $\boldsymbol{s}$ is an n-dimensional vector. Suppose the first k entries of $\boldsymbol{s}$ are leaked via side channels or other sources. One can then partition $\boldsymbol{a} = (\boldsymbol{a}_1, \boldsymbol{a}_2)$ accordingly and reduce the original n-dimensional LWE sample to an $(n-k)$-dimensional one via a straightforward reduction: $(\boldsymbol{a}_2, b - \langle \boldsymbol{a}_1, \boldsymbol{s}_1 \rangle \bmod q)$. This reduction implies that when leakage takes the form of individual coordinates of $\boldsymbol{s}$, any key-recovery algorithm is equivalent to solving an $(n-k)$-dimensional LWE problem. In other words, there exists no hint-exploitation strategy more efficient than directly solving the corresponding lower-dimensional LWE instance. We note that the approach in [11,4] of using k mod-q hints also reduces the n-dimensional LWE problem to dimension $n-k$.[3] This means that, under their treatment, each mod-q hint inflicts the same damage to the secret as leaking one coordinate of $\boldsymbol{s}$ directly.

Fundamentally, however, a mod-q hint should not be simply equated to a coordinate leakage. From an information-theoretic perspective, the former carries more information about the secret than the latter. This is because, by techniques from Brakerski, Langlois, Peikert, Regev, and Stehlé at STOC 2013 [3], each

[3] In Remark 4 of [4], the authors explicitly suggest to first perform dimension reduction with mod-q hints before considering other types of hints.

coordinate of the secret $\boldsymbol{s}$ can be chosen to be small without compromising the hardness of the problem, which is why all practical schemes adopt such small secret vectors. That is, each entry of $\boldsymbol{s}$ lies in $\{-\beta, -\beta+1, \ldots, 0, \ldots, \beta-1, \beta\}$ for some $\beta \ll q/2$. It follows that leaking one coordinate of $\boldsymbol{s}$ reveals information drawn from the set $S = \{-\beta, \ldots, \beta\}$, whereas a mod-$q$ hint $\langle \boldsymbol{s}, \boldsymbol{v} \rangle \bmod q$ carries information drawn from $Q = \{0, \ldots, q-1\}$. Since $|Q|$ is far larger than $|S|$, one mod-q hint conveys strictly more information than leaking one coordinate of $\boldsymbol{s}$. The fact that [11,4] only exploit mod-q hints to the extent of coordinate leakage suggests that part of the information contained in the hints may be left unused.

Naturally, practical attack algorithms are driven by computational efficiency, and the destructive power of different hints cannot be measured solely by information content. A core principle of public-key cryptography is that the public key information-theoretically contains all information about the secret key, yet the inability to extract this information efficiently guarantees security. After all, for completely unconstrained mod-q hints, the worst case is precisely that they take the form of coordinate leakage, for which, as argued above, no strategy more efficient than direct dimension reduction exists. Looking back at [11,4], their claim is that mod-q hints generically enable dimension reduction, and that in special cases (such as coordinate leakage) no better exploitation exists. The natural question then arises: *does no strategy more efficient than direct dimension reduction exist for all mod-q hints?* Our work answers this question in the negative. We provide a counterexample where structured mod-q hints inflict far greater damage to the secret than plain dimension reduction alone.

Modern structured-lattice cryptosystems universally employ NTT acceleration for polynomial multiplication. We therefore focus our analysis of mod-q hints on side-channel information related to NTT computations. We first briefly recall the mechanics of NTT. For the ring $R = \mathbb{Z}_q[x]/f(x) = \mathbb{Z}[x]/\langle q, f(x)\rangle$, the ideal admits the factorization $\langle q, f(x)\rangle = \prod_i \langle q, g_i(x)\rangle =: \prod_i \mathcal{G}_i$, and thus the ring decomposes as $R = \bigoplus_i \mathbb{Z}[x]/\mathcal{G}_i$. Let $s(x) \in R$ be the secret polynomial, which may constitute the entire secret key or one component of it. By the Chinese Remainder Theorem, $s(x)$ decomposes into $\mathsf{NTT}_i(s) \in \mathbb{Z}[x]/\mathcal{G}_i$. Usually, the representative of each quotient ring element is chosen to be a polynomial of degree one less than $g_i(x)$, i.e., $\deg(\mathsf{NTT}_i(s)) = \deg(g_i(x)) - 1$. It is straightforward to see that leaking one $\mathsf{NTT}_i(s)$ is equivalent to leaking $\deg(g_i(x))$ mod-q hints.

This setting is not arbitrary; analogous assumptions of partially leaked CRT coordinates were used by Bolboceanu, Brakerski, Perlman, and Sharma at ASIACRYPT 2019 [1] to study the relationship between secret entropy and RLWE security. That work, however, did not quantify the concrete attack cost or the number of hints required for efficient key recovery. It only served to argue that high-entropy secrets do not guarantee security, and to establish the associated threshold phenomenon in the noise magnitude—below which an ideal-supported secret distribution remains tractable, and above which hardness is restored.

In our attack experiments, we target Kyber-512, which involves two secret polynomials. We fix the two secrets $s_1(x)$ and $s_2(x)$, and measure how many

$\mathsf{NTT}_i(s_j)$ are required as hints to fully recover $s_j(x)$. Note that in Kyber-512 we have $\deg(g_i(x)) = 2$, so multiplying the number of required $\mathsf{NTT}_i(s_j)$ by two yields the total number of structured mod-q hints needed for full secret recovery of Kyber-512.

Our attack also employs several efficiency optimizations tailored for key recovery. For instance, the incomplete NTT expansion in Kyber-512 enables subfield acceleration, an idea originally explored by Pan, Xu, Wadleigh, and Cheng at EUROCRYPT 2021 [12]. In addition, the ring structure supports the short-vector rotation technique mentioned in [7] to boost success rate when recovering secrets from hints. We emphasize that our experiments directly synthesize the relevant hints. The goal of this work is to analyze whether hints with algebraic structure enable stronger attacks than generic frameworks, not to extract hints from hardware implementations, nor to mount practical side-channel attacks against deployed schemes. Nevertheless, since NTT is a standard technique for computational acceleration, it is reasonable to assume that information related to NTT computations may leak.

Another common acceleration technique for polynomial multiplication is FFT. Again, for the ring $R = \mathbb{Z}_q[x]/f(x)$, one evaluates $s(x)$ at the n roots ζ_i of $f(x)$ to obtain $s(\zeta_i)$, converting polynomial multiplication into point-wise multiplication. Since hardness proofs for algebraically structured LWE require $f(x)$ to be irreducible over $\mathbb{Q}$, the roots ζ_i are irrational. A simple argument shows that, in the ideal infinite-precision setting, leaking a single $s(\zeta_i)$ already contains all information needed to fully recover $s(x)$, and such leakage would count as two perfect hints (real and imaginary parts) under the DDGR taxonomy. In practice, however, FFT implementations use finite precision, typically 64 bits, so the quantized evaluations are properly classified as approximate hints. It is therefore inappropriate to measure the damage of FFT leakage using the loss metric of generic perfect hints. For ease of comparison, even though Kyber-512 does not use FFT for acceleration, we still evaluate how many 64-bit-precision FFT leakage samples suffice for full secret recovery. Our results confirm that FFT leakage inflicts substantially more damage than generic perfect hints—even though it occupies a strictly weaker category in the taxonomy.

Finally, we clarify that for both NTT and FFT leakage, our recovery algorithm does not make use of LWE samples to assist secret recovery. That is, the inputs to our recovery algorithm are exclusively the NTT hints $\langle v_i, s\rangle \bmod q$, the FFT hints $\langle v_i, s\rangle$, and the corresponding vectors v_i, consistent with the standard definitions of mod-q hints and perfect hints. Moreover, the computationally expensive parts of our algorithm, such as lattice basis reduction, take only v_i as input, which are completely independent of the secret and the public key. Therefore, if the NTT/FFT leakage scenarios we consider were to occur in practice, an adversary who knows the leakage positions in advance could even precompute these lattice basis reductions, which would drastically improve the overall attack efficiency. We leave a systematic study of precomputation strategies to future work.

1.3 Related Work

The first general framework for modeling side-channel leakage of LWE as hints or linear equations on the secret was proposed by DDGR [5]. This work has guided a number of subsequent theoretical side-channel analyses, including [11,4,18] and the present paper.

In the original DDGR framework, modular hints are not treated as dimension reduction of the LWE problem. Instead, they are modeled as sparsifying the lattice. This perspective actually captures the essence of our work well: when hints carry specific algebraic structure, the way they sparsify the lattice containing the secret makes the secret amenable to more efficient recovery.

The interpretation of mod-q hints as dimension reduction of the LWE problem originated in [11], and has profoundly shaped how subsequent works such as [4] treat mod-q hints.

We note that Yu, Cheng, Yang, Wang, Pan, and Weng at TCHES 2024 [18] present a practical side-channel study on real-world implementations. The information obtained via Hertzbleed-based leakage consists of linear hints on the secret in the NTT domain, which formally fall into the category of mod-q hints. While the specific hint model there differs from the one assumed in this work, it nonetheless provides indirect corroboration for our choice of NTT-domain coefficients as structured mod-q hints.

The results of [18] already implicitly exhibit the deviation we formalize. For Kyber-512, 256 hints derived from Hertzbleed leakage together with 150 additional hints amount to 406 mod-q hints in total, which suffice for full secret recovery using the attack script from [11]—already below the 449-hint threshold predicted by the uniform model. Our work makes this implicit deviation explicit and systematic, and further shows that when hints leak complete values of NTT-domain coordinates rather than linear relations over the NTT domain, the resulting cryptanalytic damage is even larger.

2 Preliminaries

In this work, a lattice refers to a full-rank discrete additive subgroup $L \subseteq \mathbb{R}^n$. An n-by-n full-rank matrix B is called a basis of L if L can be generated by integer linear combinations of its row vectors $\{\boldsymbol{b}_i\}_i$, i.e.,

$$L = \left\{ \sum_{i=1}^{n} z_i \boldsymbol{b}_i \;\middle|\; z_i \in \mathbb{Z} \right\}.$$

A single lattice L admits infinitely many distinct bases with varying quality. In general, a basis is deemed high-quality when its basis vectors are short and nearly mutually orthogonal. The procedure of transforming a poor-quality basis into a superior one is known as lattice basis reduction, with LLL and BKZ being the most widely adopted algorithms.

High-quality bases greatly facilitate solving the Closest Vector Problem (CVP). Given a lattice L and a target vector $\boldsymbol{v} \in \mathbb{R}^n$, CVP requires finding a lattice

point $\boldsymbol{u} \in L$ that minimizes the Euclidean distance $\|\boldsymbol{u}-\boldsymbol{v}\|$. A standard practical solver for CVP is Babai's nearest-plane algorithm, which takes the target vector $\boldsymbol{v}$ and a lattice basis B as inputs. The higher the quality of B, the closer the output lattice point will be to $\boldsymbol{v}$. The algorithm yields the true closest vector to $\boldsymbol{v}$ (i.e., an exact solution to $\mathrm{CVP}(L, \boldsymbol{v})$) only when B possesses sufficiently good geometric properties.

2.1 Algebraic Number Theory

Adjoin an algebraic number ζ to $\mathbb{Q}$ to form the number field $K = \mathbb{Q}(\zeta)$, where $f(\zeta) = 0$ for a monic irreducible polynomial $f(x) \in \mathbb{Q}[x]$. We have $K \cong \mathbb{Q}[x]/(f(x))$, and the degree n of f equals the degree of K. Elements in $K_{\mathbb{R}} = K \otimes_{\mathbb{Q}} \mathbb{R}$ are conventionally expressed as real-coefficient polynomials in ζ with degree less than n. For $a = \sum_{i=0}^{n-1} a_i \zeta^i \in K_{\mathbb{R}}$, the coefficient embedding $\tau \colon K \to \mathbb{Q}^n$ is defined by $\tau(a) = (a_i)_i$. The canonical embedding $\sigma \colon K \to \mathbb{C}^n$ satisfies $\sigma(a) = (\sigma_j(a))_j$ with $\sigma_j(a) = \sum_{i=0}^{n-1} a_i \zeta_j^i$, where ζ_j denotes the j-th complex root of $f(x)$. Complex roots of rational polynomials come in conjugate pairs, so the entries of the canonical embedding $\sigma(a)$ can be partitioned into two conjugate subsets.

An algebraic number a is called an *algebraic integer* if there exists a monic polynomial $g(x) \in \mathbb{Z}[x]$ such that $g(a) = 0$. The set of all algebraic integers in K forms a ring, known as the *ring of algebraic integers* of K and denoted $\mathcal{O}_K$. When $K = \mathbb{Q}(\zeta)$ is a cyclotomic field — that is, when $\zeta^m = 1$ for some positive integer m — we have $\mathcal{O}_K = \mathbb{Z}[\zeta] \cong \mathbb{Z}[x]/(f(x))$.

The *sum* of two ideals $\mathcal{I}$ and $\mathcal{J}$, written $\mathcal{I} + \mathcal{J}$, is the set of all elements of the form $a + b$ with $a \in \mathcal{I}$ and $b \in \mathcal{J}$. The *product* $\mathcal{I}\mathcal{J}$ is the set of all finite sums of elements of the form ab where $a \in \mathcal{I}$ and $b \in \mathcal{J}$.

Two ideals $\mathcal{I}$ and $\mathcal{J}$ are said to be *coprime* if $\mathcal{I} + \mathcal{J} = \mathcal{O}_K$. An ideal $\mathfrak{p} \subseteq \mathcal{O}_K$ is called a *prime ideal* if for all $a, b \in \mathcal{O}_K$, the condition $ab \in \mathfrak{p}$ implies $a \in \mathfrak{p}$ or $b \in \mathfrak{p}$ (or both). Every non-zero ideal $\mathcal{I}$ of $\mathcal{O}_K$ admits a unique prime ideal factorization:

$$\mathcal{I} = \prod_i \mathfrak{p}_i^{e_i},$$

where each $\mathfrak{p}_i$ is a prime ideal, and the factorization is unique up to the ordering of the factors.

2.2 Number Theoretic Transform (NTT)

The Number Theoretic Transform (NTT) is a powerful primitive for computing polynomial multiplication over the ring $\mathcal{R}_p = \mathbb{Z}[x]/\langle p, x^n + 1\rangle$ for a prime p. Modulo p, we have the factorization $x^n + 1 \equiv \prod_i f_i(x) \pmod{p}$, which yields the ideal identity

$$\langle p, x^n + 1\rangle = \prod_i \langle p, f_i(x)\rangle.$$

For any element $a \in \mathcal{R}_p$, its NTT representation is defined as $\mathsf{NTT}(a) = \big(\mathsf{NTT}_i(a)\big)_i = \big(a \bmod \langle p, f_i(x)\rangle\big)_i$. By the Chinese Remainder Theorem (CRT) for rings, a can be uniquely recovered from $\mathsf{NTT}(a)$ via the inverse NTT transform.

This reveals an intrinsic connection between NTT operations and prime ideal factorization. Consider the number field $K \cong \mathbb{Q}[x]/(x^n+1)$, whose ring of algebraic integers satisfies $\mathcal{O}_K \cong \mathbb{Z}[x]/(x^n+1)$. In this field, only the rational prime 2 ramifies, and the prime ideal factorization of the principal ideal generated by an arbitrary prime p takes the form $\langle p\rangle = \prod_i \mathfrak{p}_i$. When expressed in polynomial terms, this factorization exactly matches the core identity underlying NTT: $\langle p, x^n+1\rangle = \prod_i \langle p, f_i(x)\rangle$. Consequently, suppose we disclose a subset of NTT components $\{\mathsf{NTT}_i(a)\}_{i\in I}$ for some index subset I of all basis indices, where $a \in \mathcal{R}_p$. This disclosure is equivalent to revealing a residue λ such that $a \equiv \lambda \pmod{\prod_{i\in I} \mathfrak{p}_i}$.

2.3 Fast Fourier Transform (FFT)

The Fast Fourier Transform (FFT) is a powerful tool for accelerating polynomial multiplication over the ring $\mathcal{R} = \mathbb{Z}[x]/(x^n+1)$. Its principle relies on the n roots $\zeta_1, \ldots, \zeta_n$ of x^n+1 over $\mathbb{C}$. For $a(x) \in \mathcal{R}$, its FFT is defined as $\mathsf{FFT}(a) = (\mathsf{FFT}_i(a))_i = \big(a(\zeta_i)\big)_i$.

By the Chinese Remainder Theorem over $\mathbb{C}$, there holds the ring isomorphism

$$\mathbb{C}[x]/(x^n+1) \cong \bigoplus_i \mathbb{C}[x]/(x-\zeta_i),$$

and thus a can be uniquely recovered from $\mathsf{FFT}(a)$.

In fact, the FFT described here is equivalent to the canonical embedding of number fields. Note that $\mathsf{FFT}(a) = \sigma(a)$, where σ denotes the canonical embedding map. Throughout this paper, except for sections specifically devoted to FFT leakage, we adopt the canonical embedding notation.

3 NTT Leakage and Attacks

Let K be an n-dimensional number field with ring of algebraic integers $\mathcal{R}$, and let $\mathcal{Q}$ denote an ideal of $\mathcal{R}$. Let $e \in \mathcal{R}$ be a certain short element, where the term "short" means that the length of the canonical embedding of e is bounded by ℓ, i.e., $\|\sigma(e)\| \le \ell$. The congruence $e \equiv \lambda \pmod{\mathcal{Q}}$ implies that $\lambda - e$ lies in the ideal lattice $\mathcal{Q}$. If there exists a unique lattice point of $\mathcal{Q}$ within distance less than ℓ from λ, then the conditions $e \equiv \lambda \pmod{\mathcal{Q}}$ and $\|\sigma(e)\| \le \ell$ uniquely determine e. Under this condition, given a sufficiently high-quality basis for the lattice $\mathcal{Q}$, we are able to fully recover e solely from the value λ.

Restrict our attention to the cyclotomic number field $K \cong \mathbb{Z}[x]/(x^n+1)$, which is widely adopted in cryptographic schemes. For any element $a \in K_{\mathbb{R}}$, the equality $\sqrt{n}\cdot\|\tau(a)\| = \|\sigma(a)\|$ holds. Consequently, we may discard the canonical embedding entirely and work exclusively with the coefficient embedding τ.

Algorithm 1 Recover short element e from partial NTT leakage

Require: Index subset I, partial NTT residues $\{\lambda_i = \mathsf{NTT}_i(e)\}_{i\in I}$, bound ℓ
Ensure: Short element $e \in \mathcal{R}$ satisfying $e \equiv \lambda_i \pmod{\mathfrak{p}_i}, \|\sigma(e)\| \le \ell$
1: $\mathcal{Q} \leftarrow \prod_{i\in I} \mathfrak{p}_i$
2: Apply CRT to construct residue λ such that $\lambda \equiv \lambda_i \pmod{\mathfrak{p}_i}, \forall i \in I$
3: Run lattice basis reduction on ideal lattice $\mathcal{Q}$ to get a reduced basis B
4: Utilize lattice basis B to find a lattice point $v \in \mathcal{Q}$ near λ
5: $e \leftarrow \lambda - v$
6: **return** e

In what follows, we elaborate on the concrete implementation details and optimization strategies for each step in Algorithm 1. We begin with the construction of a lattice basis B for $\mathcal{Q}$; all preceding steps can be efficiently realized via standard off-the-shelf techniques.

Recall the ideal lattice $\mathcal{Q} = \prod_{i\in I} \mathfrak{p}_i = \prod_{i\in I} \langle p, f_i(\zeta)\rangle = \langle p, \prod_{i\in I} f_i(\zeta)\rangle$. Its associated lattice under the coefficient embedding admits the basis matrix

$$L_{\mathcal{Q}} = \begin{bmatrix} \mathsf{Rot}_\zeta\left(\prod_{i\in I} f_i(\zeta)\right)_{n\times n} \\ p \cdot \mathrm{I}_{n\times n} \end{bmatrix},$$

where for any polynomial $h(\zeta) \in \mathbb{Z}[\zeta]$ and all $j = 0, \ldots, k-1$, the j-th row of $\mathsf{Rot}_\zeta(h(\zeta))_{k\times k}$ is defined as $\tau\big(\zeta^j h(\zeta)\big)$. We then perform lattice basis reduction on $L_{\mathcal{Q}}$ to obtain the reduced basis B.

In practice, beyond conventional basis reduction heuristics, we exploit a more sophisticated toolkit systematically formalized as subfield attacks for ideal lattices. The work [12] proposes tailored optimizations exactly for the cyclotomic setting considered in this paper; we refer readers to the original manuscript for full technical details.

For our target number field $K \cong \mathbb{Z}[x]/(x^n+1)$, the prime ideal factors of any rational prime $p \in \mathbb{Z}$ follow a uniform structural pattern:

- If $p \equiv 1 \pmod 4$, each prime ideal factor takes the form $\langle p, \zeta^t + u\rangle$;
- If $p \equiv 3 \pmod 4$, each prime ideal factor can be written as $\langle p, \zeta^{2t} + u\zeta^t + 1\rangle$.

where t is a power of two and u denotes a certain integer constant.

When $t \ge 1$, the product polynomial $\prod_{i\in I} f_i(\zeta)$ inside $\mathcal{Q} = \big\langle p, \prod_{i\in I} f_i(\zeta)\big\rangle$ factors into a polynomial $f^*(\zeta^t)$ in the variable ζ^t. Let $\zeta' \triangleq \zeta^t$, which satisfies $(\zeta')^{n/t} + 1 = 0$. Under this substitution, the basis matrix $L_{\mathcal{Q}}$ can be restructured as

$$L_{\mathcal{Q}} = \begin{bmatrix} \mathsf{Rot}_{\zeta'}\big(f^*(\zeta'), \zeta^0\big)_{(n/t)\times(n/t)} \\ \vdots \\ \mathsf{Rot}_{\zeta'}\big(f^*(\zeta'), \zeta^{t-1}\big)_{(n/t)\times(n/t)} \\ p \cdot \mathrm{I}_{n\times n} \end{bmatrix}.$$

For $i = 0, \ldots, t-1$ and $j = 0, \ldots, k-1$, the j-th row of $\mathsf{Rot}_{\zeta'}\big(h(\zeta'), \zeta^i\big)_{k\times k}$ is defined as

$$\tau\big(\zeta^i \cdot \big[(\zeta')^j h(\zeta')\big]\big) = \tau\big(\zeta^{i+jt} h(\zeta^t)\big).$$

This new representation endows the basis matrix with a highly symmetric block structure. We first perform lattice reduction on the smaller-dimensional submatrix

$$L'_{\mathcal{Q}} = \begin{bmatrix} \mathsf{Rot}_{\zeta'}\big(f^*(\zeta')\big)_{(n/t)\times(n/t)} \\ p \cdot \mathrm{I}_{(n/t)\times(n/t)} \end{bmatrix}.$$

Here, $f^*(\zeta')$ is treated as an element of the subfield $K' = \mathbb{Q}(\zeta')$, which has degree n/t. Accordingly, each row of $\mathsf{Rot}_{\zeta'}\big(f^*(\zeta')\big)_{(n/t)\times(n/t)}$ has exactly n/t entries. This step reduces the lattice dimension from n to n/t. After computing the reduced basis for $L'_{\mathcal{Q}}$, we lift this low-dimensional basis back to the full matrix $L_{\mathcal{Q}}$, yielding a high-quality global basis B with substantially improved computational efficiency.

It has been proven in [12] that the shortest vector of $L_{\mathcal{Q}}$ also corresponds to the shortest vector of $L'_{\mathcal{Q}}$.[4] Nevertheless, two practical obstacles remain: first, we cannot guarantee successful extraction of the shortest vector from $L'_{\mathcal{Q}}$; second, our ultimate target is to compute a high-quality basis for the full lattice $L_{\mathcal{Q}}$. Accordingly, when lifting the reduced basis of $L'_{\mathcal{Q}}$ back to the full-dimensional lattice $L_{\mathcal{Q}}$, we multiply each element of the basis of $L'_{\mathcal{Q}}$ by ζ^i for all $i = 0, 1, \ldots, t'-1$ where $t' > t$. We then feed this expanded set of short vectors into a lattice reduction algorithm such as LLL to perform linear combination and orthogonalization, which outputs a higher-quality global lattice basis B.

We deploy standard routines such as Babai's nearest-plane algorithm to solve the closest vector problem (CVP) with the reduced basis B. Instead of only running CVP on λ against B, we additionally execute the same CVP procedure for $\zeta^i\lambda$ with $i = 1, 2, \ldots, n$. If the CVP solver succeeds for any $\zeta^i\lambda$, we immediately retrieve $\zeta^i e$, from which the secret element e can be fully recovered.

Since the basis B constructed via our preprocessing exhibits no directional bias, the probability that at least one of the n independent CVP trials succeeds is approximately n times the success probability of a single standalone trial. This rotation-augmented CVP technique was first formally introduced by Hou, Jiang, and Ogilvie in [7], and it drastically boosts the success rate when solving CVP over structured ideal lattices.

4 FFT Leakage and Attacks

We begin with an interesting observation. The theorem below shows that for a secret $s \in \mathcal{R}$, leaking merely a single FFT coordinate $\mathsf{FFT}_i(s)$ for some index i uniquely determines s information-theoretically.

Theorem 1. *Let $a, b \in \mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$, where n is a power of two. If $\mathsf{FFT}_i(a) = \mathsf{FFT}_i(b)$ for some i, then $a = b$. Hence, knowing $\mathsf{FFT}_i(s)$ for any single i uniquely determines the unknown s information-theoretically.*

[4] This holds up to appropriate zero-padding to match dimensions, or equivalently, by treating all entries as polynomials in ζ directly

Proof. It follows from the assumption that $\mathsf{FFT}_i(a-b)=0$, so $x^n+1 \mid a-b$. Since a and b are elements of $\mathcal{R}$ with degree less than n, we must have $a=b$.

While a single infinite-precision FFT coordinate uniquely determines the secret polynomial in theory, practical FFT computations only yield finite-precision values. We therefore design a lattice-based key-recovery approach that recovers the secret from multiple finite-precision FFT hints. We first present the general lattice construction for a single secret polynomial, and then detail the experimental setup for Kyber-512.

4.1 General Lattice Construction from FFT Hints

Let $s(x) \in \mathcal{R} = \mathbb{Z}[x]/(x^n+1)$ be the target secret polynomial, where n is a power of two. We select a set of non-conjugate roots $\rho_0, \ldots, \rho_{K-1}$ of x^n+1 over $\mathbb{C}$. We fix a fixed-point scaling factor $Q = 2^p$ for some precision p, and define integer matrices encoding the polynomial evaluation map:

$$A_{\mathrm{Re}}[i,j] = \left\lfloor Q \,\mathrm{Re}(\rho_j^i) \right\rceil, \qquad A_{\mathrm{Im}}[i,j] = \left\lfloor Q \,\mathrm{Im}(\rho_j^i) \right\rceil.$$

The leaked hints are the quantized evaluations of s at these roots:

$$u_j = \left\lfloor Q \,\mathrm{Re}(s(\rho_j)) \right\rceil, \qquad v_j = \left\lfloor Q \,\mathrm{Im}(s(\rho_j)) \right\rceil.$$

We construct an integer lattice basis in row form:

$$B = \begin{pmatrix} SI_n & A_{\mathrm{Re}} & A_{\mathrm{Im}} \\ 0 & u & v \end{pmatrix},$$

where $S = 2^\sigma$ is an additional scaling factor for the secret coordinates, used for lattice sieving. The parameter σ controls the relative weight of the secret block and is independent of the fixed-point precision p.

The lattice contains the short target vector

$$(s, -1)B = \big(Ss,\ sA_{\mathrm{Re}} - u,\ sA_{\mathrm{Im}} - v\big).$$

Since the coefficients of s are small and the latter two blocks consist only of fixed-point rounding residuals, this vector is unusually short compared to the lattice determinant. We run LLL lattice reduction on B, which produces a unimodular transformation T such that $R = TB$ is LLL-reduced. We then scan rows of T for entries equal to ± 1 at the secret-offset position. If $a = T_{r,n} \in \{\pm 1\}$, we recover a candidate secret $\hat{s}_i = -aT_{r,i}$. We filter candidates to retain only those with all coefficients within the support $[-\beta, \beta]$, and select the best candidate by the norm of the corresponding reduced lattice vector.

5 Experiment

We instantiate our hint-based key-recovery algorithm on Kyber-512 to enable comparisons with prior work.

5.1 Setup

We now specialize the construction to Kyber-512. The ring dimension is $n = 256$, the modulus is $q = 3329$, and the MLWE module rank is two. Both the secret polynomials $s^{(0)}, s^{(1)}$ and the error polynomials are sampled from the centered binomial distribution CBD_3, so every secret coefficient lies in $[-3, 3]$. The two secret polynomials are processed independently: each polynomial is recovered from its own lattice, and we consider the key fully recovered only if all 512 secret coefficients are correct. We synthesize the hints directly from freshly sampled secret keys; the reduction inputs are therefore the leakage hints alone, and no physical trace acquisition is involved.

NTT hints. For Kyber-512, $x^n + 1$ factors modulo q into 128 quadratic factors, and each leaked quadratic CRT residue $\mathsf{NTT}_i(s^{(j)})$ corresponds to two scalar mod-q hints, since $\deg(g_i) = 2$. We leak the same number m of slots for each secret polynomial, so the total number of structured mod-q scalar hints is

$$h = 2 \times m \times 2 = 4m.$$

For example, $h = 240$ corresponds to $m = 60$ leaked CRT slots per secret, $h = 244$ to $m = 61$, and $h = 204$ to $m = 51$. For each secret we build the ideal lattice $\mathcal{Q}$ associated with the leaked residues. To describe the reduction, we decompose the subfield-aware implementation into a four-stage pipeline:

1. construct the 128-dimensional subfield lattice from the leaked residues;
2. run lattice reduction on the 128-dimensional subfield lattice;
3. lift the reduced basis exactly back to the 256-dimensional lattice;
4. run a final LLL cleanup on the 256-dimensional lattice.

We evaluate three LLL campaigns with respect to this pipeline. The *original direct* campaign executes only Stage 4: it constructs the public 256-dimensional lattice and runs LLL directly, without any subfield descent. The *full subfield* campaign executes Stages 1–4, performing the subfield LLL, the exact lift, and the final 256-dimensional cleanup. The *truncated subfield* campaign executes Stages 1–3, performing the subfield LLL and the exact lift but omitting the final cleanup. We additionally evaluate two BKZ campaigns that replace Stage 2 by BKZ with block size 10 or 30, followed by the same lift and the rotation-augmented nearest-plane decoding described in Algorithm 1. The stage numbers above label the reduction pipeline and are not the numbered lines of Algorithm 1. Each of these five campaigns is a separate, fixed single configuration and is run on the same seed cohort, so their per-instance outcomes are directly comparable.

FFT hints. We set the fixed-point precision to $p = 64$, i.e. $Q = 2^{64}$. The scaling factor $S = 2^{\sigma}$ is tuned as an attack parameter; in particular, it is not the fixed-point precision of the measurements. We count the total number of scalar hints as follows:

$$h = 2 \times K \times 2 = 4K,$$

where the factors correspond to the two secret polynomials, K complex FFT slots per polynomial, and two real coordinates (real and imaginary) per complex slot. For example, $h = 84$ means $K = 21$ complex FFT slots per secret, yielding 42 scalar hints per polynomial and 84 scalar hints across both components. For each reported point we build two independent lattices of size $257 \times (256 + 42) = 257 \times 298$, run LLL on each separately, and consider the key fully recovered only if both secret polynomials are reconstructed. Both hint counts and reported runtimes account for both secret components, so no additional factor of two is applied.

Parameter selection and the fixed fallback. All tuning uses a disjoint screening cohort of 16 independent public inputs; every configuration is frozen before the confirmation cohort is evaluated. For FFT we fix $p = 64$ and scan even scaling exponents $\sigma \in \{2, 4, \ldots, 16\}$ at $h = 72, 80, 84$, all exponents $\sigma \in \{1, \ldots, 16\}$ at $h = 76$, and use the anchors $\sigma = 4$ at $h = 64, 68$ and $\sigma = 12$ at $h = 128$. At each h, we select the candidate with the largest whole-cell success count on the screening cohort, breaking ties by median compute time and then by a deterministic parameter order. For NTT we analogously select between a no-cleanup presentation and several redundant lift presentations. The confirmation cohort consists of 100 independent public inputs, and each reported cell uses 100 trials. The NTT campaigns reported below use one fixed configuration per point and no fallback. For FFT we additionally report a *fixed multi-configuration fallback* curve: at each available h, the event is the paired-seed union of all pre-confirmation-frozen constituent attacks at hint budgets $h' \leq h$, with the attack for h' using the first h' hints. This curve demonstrates the hint threshold achievable by adding fixed fallback configurations; its success events are nested as h grows, and it involves no per-instance parameter selection based on the confirmation truth. Because the union event has no corresponding single-attack run, it is assigned no time and is excluded from every timing comparison; the sequential and parallel costs of its constituents are retained solely as an audit attachment.

Statistics and timing. Unless stated otherwise, every cell is measured on 100 independent trials, and success requires all 512 secret coefficients to be correct. We denote by H_{50}, H_{90}, and H_{100} the smallest tested hint budgets at which the observed success rate reaches 50%, 90%, and 100%; H_{100} is a sample-wide statement, not a probability-one guarantee. All error bars are pointwise 95% Wilson intervals, and a 100/100 observed outcome still only corresponds to a Wilson lower bound of approximately 96.3%. Every timed point is one fixed attack configuration run once, with its own per-trial measured time; no fallback or cross-h times are summed, and no constituent cost is substituted for a union event. Runtimes cover basis construction, one offline preprocessing step where applicable, reduction, and public decoding; they exclude synthetic input generation and separately recorded algebraic and truth audits. The two FFT curves share the paired seeds 7000–7099; the three NTT LLL campaigns, the NTT BKZ campaigns, and the Li campaign share the seeds 5000–5099; and the Cao–Jiang–

Table 1. Minimum tested hint counts attaining each observed full-recovery threshold for Kyber-512. Every point uses one fixed attack configuration except the explicitly marked FFT multi-configuration success row, whose event is the paired-seed union of all frozen FFT attacks with at most h hints and which is assigned no time. Original LLL directly reduces the public 256-dimensional lattice; full subfield LLL includes 128-dimensional reduction, exact lift, and 256-dimensional cleanup, while the truncated variant omits the cleanup. Hint units are native to each model and are not equi-informative per scalar. For single-configuration rows, the final column is the median compute time at the H_{100} configuration, not a deadline for all trials.

Method	First success	H_{50}	H_{90}	H_{100}	Median time (s)
NTT / original direct 256D LLL	236	240	240	240	85.23
NTT / full subfield LLL, Stages 1–4	236	240	240	240	69.62
NTT / truncated subfield LLL, Stages 1–3	224	240	240	240	67.52
FFT / fixed multi-configuration fallback, $p = 64$ (success only)	64	68	76	84	–
FFT / one selected scale per h, $p = 64$	64	68	84	128	251.89
Cao–Jiang–Cheng / first lattice + LLL	224	256	256	256	871.04
Cao–Jiang–Cheng / first lattice + flatter + LLL	224	256	256	256	262.95
May–Nowakowski / random mod-q + LLL	448	448	448	464	222.84

Cheng and May–Nowakowski baselines use the separate cohort 1000–1099. The campaigns were run under different concurrency limits and machine loads, so wall-clock comparisons are observed alignments rather than controlled speedup measurements, and no interpolation or imputation is used.

No use of the MLWE public key. Our NTT and FFT lattices do not jointly incorporate the MLWE public-key equation $(A, b = As+e)$. The MLWE instance is only used to sample the secrets according to the correct CBD distribution. Lattice reduction uses only the leakage hints $\langle v_i, s\rangle \bmod q$ or $\langle v_i, s\rangle$ and the corresponding vectors v_i; the attack is thus direct secret recovery from partial structured hints, rather than a joint lattice attack combining hints and MLWE samples. In particular, the reduction depends only on the leakage positions and not on the secret or the public key, which in principle permits precomputation of the reduced bases.

Full-Recovery Thresholds Table 1 reports the smallest tested hint budgets at which each method first succeeds and at which it reaches the 50%, 90%, and 100% observed recovery thresholds; Figure 1 plots the full confirmation curves. Native hint units differ across methods, so comparisons are made under the exact accounting of the hint taxonomy: NTT scalars are counted as mod-q hints, quantized FFT scalars as approximate hints, and the baseline rows in their respective native units.

The three NTT LLL campaigns behave identically at their common measured points. The original direct attack recovers the first instances at $h = 236$ (8/100) and reaches 100/100 at $h = 240$ and $h = 244$, with a median compute time of 85.23 seconds at $h = 240$. The full subfield campaign has exactly the same rates $8\%, 100\%, 100\%$ at $h = 236, 240, 244$, but a lower median of

69.62 seconds at $h = 240$: the 128-dimensional reduction plus exact lift is 18.3% faster than direct reduction at the same success rate. The truncated campaign, which omits the final cleanup, is faster still, with a median of 67.52 seconds at $h = 240$, i.e. 20.8% below the original attack; its first success occurs as early as $h = 224$ (11/100), and its measured rates at $h = 220, 224, 228, 232, 236, 240, 244$ are $0\%, 11\%, 29\%, 14\%, 8\%, 100\%, 100\%$. The non-monotone middle points of the truncated and full campaigns are retained as measured, rather than smoothed; the subfield descent therefore buys a modest early-success probability and a roughly 20% runtime reduction, while the stable threshold remains $h = 240$ for all three presentations.

The FFT transition is even sharper. With one independently screened scale per h, the measured rates rise from 12% at $h = 64$, to 68% at $h = 68$, 65% at $h = 72$, 77% at $h = 76$, 72% at $h = 80$, and 91% at $h = 84$, reaching 100% at $h = 128$; a single fixed configuration therefore reaches 100/100 at $h = 128$ with a median of 251.89 seconds and a sample-wide deadline of 266.31 seconds. The fixed multi-configuration fallback, which tries every frozen constituent with at most h hints, lifts the rates on the same seeds: it starts at 12% for $h = 64$, reaches 71% at $h = 68$, 87% at $h = 72$, 97% at $h = 76$, and 99% at $h = 80$, and attains 100/100 at $h = 84$ and $h = 128$. This gain is a union effect of fixed fallback configurations rather than a single-attack result, so it is reported as a success-only curve with no time. The selected scale exponents are $\sigma = 4$ at $h = 64, 68, 76, 80$, $\sigma = 6$ at $h = 72$, $\sigma = 10$ at $h = 84$, and $\sigma = 12$ at $h = 128$, illustrating that the optimal σ is an empirical lattice parameter rather than a fixed-point width.

For the uniform-hint baselines, our first-projection-lattice implementation of [4] recovers the first instances at $h = 224$ (7/100 with plain LLL and 16/100 with flatter) and reaches 100/100 at $h = 256$, both for plain LLL (median 871.04 seconds) and for the flatter-preprocessed variant followed by LLL (median 262.95 seconds). These numbers implement only the first projection lattice and the public decoding of [4], not the full multi-stage attack of that paper. A direct re-implementation of the random mod-q attack of [11] succeeds on no instance at $h = 432$, on 94/100 at $h = 448$, and on 100/100 at $h = 464$, with a median time of 222.84 seconds; the May–Nowakowski curve uses random mod-q hints rather than perfect hints.

Subfield BKZ and the Progressive G6K Baseline We next strengthen the NTT attack by replacing LLL with BKZ on the 128-dimensional subfield lattice, using independent fixed block sizes, MPFR 128-bit arithmetic, LLL preprocessing, and at most four BKZ tours before lifting back to 256 dimensions; we then compare against the progressive G6K implementation of Li et al. [9], which builds two 256-dimensional lattices from 256 ratio equations per instance and runs the G6K pump-and-jump schedule with jump 9, `extra_dim4free` = 27, one round per block size, starting at $\beta = 25$. Table 2 and Figure 2 report the transitions.

Table 2. Measured NTT subfield BKZ and Li progressive G6K transitions. The NTT block sizes and the Li schedule caps are not equivalent reduction parameters, and their native hint units differ (scalar mod-q ranks vs. ratio equations). Times are medians over all 100 attempts, including failures.

Method	β / cap	h	Successes / 100	Median time (s)	Notes
NTT / subfield BKZ-10	10	208	5	69.20	first success
NTT / subfield BKZ-10	10	216	50	73.45	H_{50}
NTT / subfield BKZ-10	10	224	99	72.00	H_{90}
NTT / subfield BKZ-10	10	228	100	70.54	H_{100}
NTT / subfield BKZ-30	30	188	2	77.83	first success
NTT / subfield BKZ-30	30	192	25	77.37	
NTT / subfield BKZ-30	30	200	75	77.32	H_{50}
NTT / subfield BKZ-30	30	204	100	78.88	$H_{90} = H_{100}$
Li / progressive G6K	35	256	0	3310.25	
Li / progressive G6K	36	256	7	3652.42	first success
Li / progressive G6K	37	256	40	3935.99	
Li / progressive G6K	38	256	74	4053.09	H_{50}
Li / progressive G6K	39	256	91	4097.34	H_{90}
Li / progressive G6K	40	256	100	4097.34	H_{100}

Subfield BKZ with $\beta = 10$ reduces the stable threshold to $h = 228$ (100/100, median 70.54 seconds), and $\beta = 30$ reduces it further to $h = 204$, where 90% is reached already at the same point and 100/100 succeeds with a median time of 78.88 seconds (maximum 80.66 seconds over the 100 successful trials). Relative to the full subfield LLL threshold $h = 240$, this is a 15% reduction within the same hint model. The improvement is structural rather than algorithmic: reduction is performed on a lattice of dimension 128 instead of 256, and in this regime subfield BKZ-30 is not only more hint-efficient but also faster than plain LLL on the full lattice. By contrast, Li et al. [9] observe no success up to schedule cap 35, first succeed at cap 36, and reach 100/100 only at cap 40, with a median time of 4097.34 seconds and a sample-wide deadline of 4884.63 seconds; their block-size caps are schedule parameters and must not be read as equivalent fixed BKZ block sizes. Moreover, from $\beta \geq 40$ onward the G6K default extra free dimensions are retained, and each higher-cap point reuses the first publicly accepted output together with accumulated time rather than constituting an independent run.

Time-Aligned Comparison Hint counts alone do not distinguish between a cheap and an expensive single reduction, so we also report the minimum tested hint count that achieves a target recovery proportion within a time budget T. For a method m with budget h, let

$$\widehat{P}_{m,h}(T) = \frac{1}{100} \sum_{i=1}^{100} \mathbf{1}\{\text{single-configuration full recovery of } i \text{ and } t_i \leq T\},$$

Table 3. Minimum tested hint counts for 100/100 full recoveries within the stated time budget. Every success and time belongs to the same single fixed configuration; no fallback or cross-h times are summed, and the FFT multi-configuration success row is excluded. A dash denotes that no tested configuration met the target; native hint units differ across rows.

Method	90 s	180 s	240 s	300 s	600 s	1800 s
NTT / original direct 256D LLL	240	240	240	240	240	240
NTT / full subfield LLL, Stages 1–4	240	240	240	240	240	240
NTT / truncated subfield LLL, Stages 1–3	240	240	240	240	240	240
NTT / subfield BKZ-10	228	228	228	228	228	228
NTT / subfield BKZ-30	204	204	204	204	204	204
FFT / one selected scale per h, $p = 64$	–	–	–	128	128	128
Cao–Jiang–Cheng / first lattice + LLL	–	–	–	–	–	256
Cao–Jiang–Cheng / first lattice + flatter + LLL	–	–	320	320	256	256
May–Nowakowski / random mod-q + LLL	–	–	–	464	464	464

and define $H_m(T; p) = \min\{h : \widehat{P}_{m,h}(T) \geq p\}$, leaving the entry blank when no tested configuration meets the target. Table 3 and Figure 3 report the resulting frontiers.

All three NTT LLL campaigns achieve 100/100 at $h = 240$ within a 90-second budget, with medians 85.23, 69.62, and 67.52 seconds; the subfield presentations are therefore 18.3% and 20.8% faster at the same hint count. Subfield BKZ-30 reaches 100/100 with 204 hints and BKZ-10 with 228 hints under the same 90-second budget. The FFT single-scale campaign first reaches 100/100 at $h = 128$ from the 300-second budget onward (all successful instances complete by 266.31 seconds); its multi-configuration success curve, which reaches 100/100 at $h = 84$, is deliberately excluded because the union event has no single-configuration time. The May–Nowakowski random mod-q attack requires $h = 464$ and a 300-second budget, and the first-lattice implementation of Cao–Jiang–Cheng needs $h = 320$ at 240 seconds and $h = 256$ from the 600-second budget onward with flatter preprocessing, while plain LLL reaches $h = 256$ only within 1800 seconds. Because the campaigns were run on a shared machine with different concurrency limits, these numbers are measured wall-time alignments and should not be quoted as hardware speedup factors.

Experimental Summary Under the generic threshold analysis of May and Nowakowski [11], breaking Kyber-512 with LLL requires 449 uniformly random mod-q hints or 234 uniformly random perfect hints. The refined construction of Cao, Jiang, and Cheng [4] keeps the mod-q threshold unchanged and lowers the perfect-hint threshold to 230. Our measured thresholds for structured leakage are dramatically smaller.

For NTT leakage, every leaked scalar is formally a mod-q hint under the DDGR taxonomy [5]. All three LLL campaigns—original direct, full subfield, and truncated subfield—nevertheless recover the full key from $h = 240$ hints,

roughly 47% below the 449-hint uniform mod-q estimate, i.e. about half; the truncated presentation additionally reaches its first success at $h = 224$ while running about 21% faster than the direct attack. Subfield BKZ lowers the stable threshold to 228 hints with $\beta = 10$ (about 49%) and to 204 hints with $\beta = 30$ (about 55%). The structured mod-q hints thus reach thresholds that the generic framework predicts only for uniform perfect hints, which are a strictly stronger class.

For FFT leakage, every leaked scalar is a finite-precision, hence approximate, hint—formally a weaker class than a perfect hint. Probabilistic recovery starts at 64 such scalars, the fixed multi-configuration fallback reaches 100/100 at $h = 84$ as a success-only union, and a single fixed configuration reaches 100/100 at $h = 128$. Measured against the 234 perfect-hint threshold of [11] (the 230 threshold of [4] gives nearly identical figures), the reductions are approximately 73% at the first-success point 64, 64% at the fallback point 84, and 45% at the single-configuration point 128: overall, roughly one half to three quarters of the generic perfect-hint budget, i.e. an improvement of approximately 50%–75%. Approximate hints that exploit algebraic structure are therefore more powerful than uniformly sampled perfect hints, despite occupying a weaker class in the taxonomy.

The central cryptanalytic conclusion is that the DDGR hint classification alone is insufficient for concrete security estimation. The taxonomy labels each hint by its local algebraic form—perfect, mod-q, or approximate—and the too-many-hints thresholds of [11,4] assume that the corresponding vectors v_i are uniformly sampled. This accounting discards the geometry by which the hints sparsify the lattice containing the secret. NTT coordinates do not sparsify a random lattice: they prescribe a residue modulo a product of prime ideals, whose ideal lattice has a rotation-symmetric block structure and a subfield decomposition that the uniform model cannot access. FFT coordinates do not provide independent approximate scalars either: real and imaginary parts of the same evaluation are coordinates of one algebraic number, and the information-theoretic uniqueness of a single infinite-precision coordinate is the extremal manifestation of that redundancy. Classifying structured leakage by hint type and then inserting the resulting count into a uniform-model estimator consequently ignores the algebraic structure and yields overly optimistic security estimates. Finally, both attacks achieve these numbers without using the MLWE public-key equation at all, and their expensive reduction phases depend only on the leakage positions; we leave a systematic study of precomputed bases for known leakage positions to future work.

References

1. Bolboceanu, M., Brakerski, Z., Perlman, R., Sharma, D.: Order-lwe and the hardness of ring-lwe with entropic secrets. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 91–120. Springer International Publishing, Cham (2019)

2. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehle, D.: CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 353–367 (2018). https://doi.org/10.1109/EuroSP.2018.00032
3. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing. p. 575–584. STOC '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2488608.2488680, https://doi.org/10.1145/2488608.2488680
4. Cao, J., Jiang, H., Cheng, Q.: Refined attack on lwe with hints: Constructing lattice via gaussian elimination. In: Tauman Kalai, Y., Kamara, S.F. (eds.) Advances in Cryptology – CRYPTO 2025. pp. 385–416. Springer Nature Switzerland, Cham (2025)
5. Dachman-Soled, D., Ducas, L., Gong, H., Rossi, M.: Lwe with side information: Attacks and concrete security estimation. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 329–358. Springer International Publishing, Cham (2020)
6. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems **2018**(1), 238–268 (Feb 2018). https://doi.org/10.13154/tches.v2018.i1.238-268, https://tches.iacr.org/index.php/TCHES/article/view/839
7. Hou, J., Jiang, H., Ogilvie, T.: Careful with the ring! concrete hardness gaps between lwe and mlwe. In: Heninger, N., Rosulek, M. (eds.) Advances in Cryptology – CRYPTO 2026. pp. 462–497. Springer Nature Switzerland, Cham (2026)
8. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography **75** (06 2014). https://doi.org/10.1007/s10623-014-9938-4
9. Li, Z., Xu, J., Song, J., Xu, H., Jia, Y., Zou, Y., Hu, L.: Improved attacks against lattice-based kems using hints from hertzbleed. IACR Transactions on Cryptographic Hardware and Embedded Systems **2025**(4), 463–485 (2025). https://doi.org/10.46586/tches.v2025.i4.463-485, https://tches.iacr.org/index.php/TCHES/article/view/12418
10. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) Advances in Cryptology – EUROCRYPT 2010. pp. 1–23. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
11. May, A., Nowakowski, J.: Too many hints – when lll breaks lwe. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 106–137. Springer Nature Singapore, Singapore (2023)
12. Pan, Y., Xu, J., Wadleigh, N., Cheng, Q.: On the ideal shortest vector problem over random rational primes. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021. pp. 559–583. Springer International Publishing, Cham (2021)
13. Peikert, C., Regev, O., Stephens-Davidowitz, N.: Pseudorandomness of ring-lwe for any ring and modulus. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing. p. 461–473. STOC 2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3055399.3055489, https://doi.org/10.1145/3055399.3055489
14. Pellet-Mary, A., Stehlé, D.: On the hardness of the ntru problem. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2021. pp. 3–35. Springer International Publishing, Cham (2021)

15. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing. p. 84–93. STOC '05, Association for Computing Machinery, New York, NY, USA (2005). https://doi.org/10.1145/1060590.1060603, https://doi.org/10.1145/1060590.1060603
16. Soni, D., Basu, K., Nabeel, M., Aaraj, N., Manzano, M., Karri, R.: FALCON. In: Hardware Architectures for Post-Quantum Digital Signature Schemes, pp. 31–41. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-57682-0_3
17. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Matsui, M. (ed.) Advances in Cryptology – ASIACRYPT 2009. pp. 617–635. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
18. Yu, T., Cheng, C., Yang, Z., Wang, Y., Pan, Y., Weng, J.: Hints from hertz: Dynamic frequency scaling side-channel analysis of number theoretic transform in lattice-based kems. IACR Transactions on Cryptographic Hardware and Embedded Systems **2024**(3), 200–223 (Jul 2024). https://doi.org/10.46586/tches.v2024.i3.200-223, https://tches.iacr.org/index.php/TCHES/article/view/11675

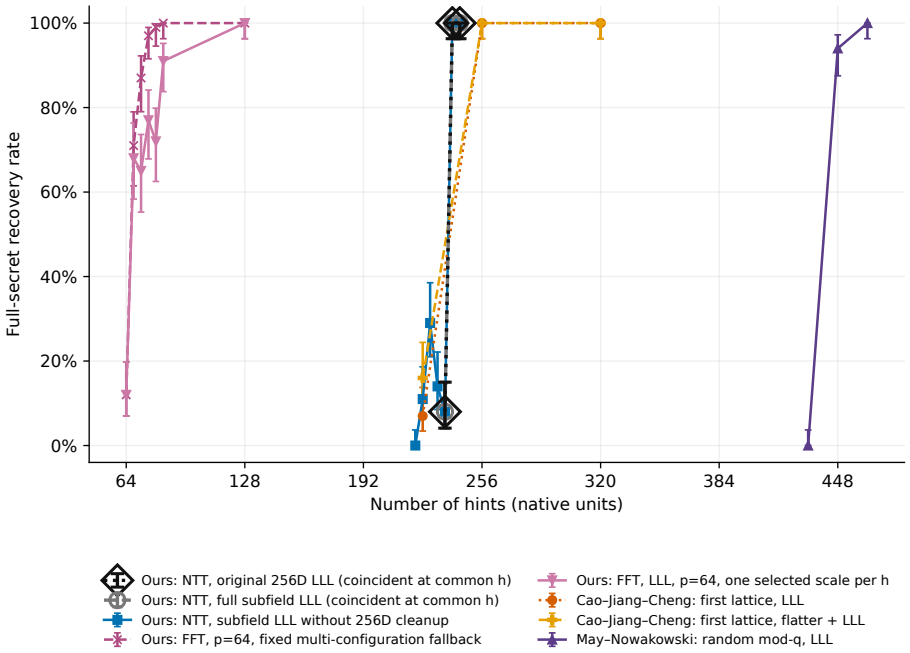

Fig. 1. Full-secret recovery with LLL-based reduction on synthetic MLWE instances ($n = 256$, $q = 3329$, module rank two, CBD_3 secrets and errors). Each point uses 100 independent confirmation trials and requires all 512 secret coefficients to be correct. The FFT scale and the NTT lift presentation at each h were selected with a disjoint 16-trial screening cohort. Original NTT LLL directly reduces the public 256-dimensional lattice; full subfield LLL performs 128-dimensional LLL, exact lift, and 256-dimensional LLL cleanup, and the truncated variant omits the cleanup. All three have rates $8\%, 100\%, 100\%$ at their common measured points $h = 236, 240, 244$; nested black open diamonds, gray open circles, and blue filled squares show this exact overlap. FFT is shown in two forms: one independently screened scale per h, and a fixed multi-configuration fallback whose event is the paired-seed union of all frozen FFT attacks with at most h hints; the latter reaches 100/100 at $h = 84$ but has no single-configuration time and is excluded from timing comparisons. The Cao–Jiang–Cheng baseline is our first-projection-lattice implementation (its flatter variant includes a final LLL step), and the May–Nowakowski curve uses random mod-q hints. Error bars are pointwise 95% Wilson intervals; native hint units differ across curves, and lines only connect measured points.

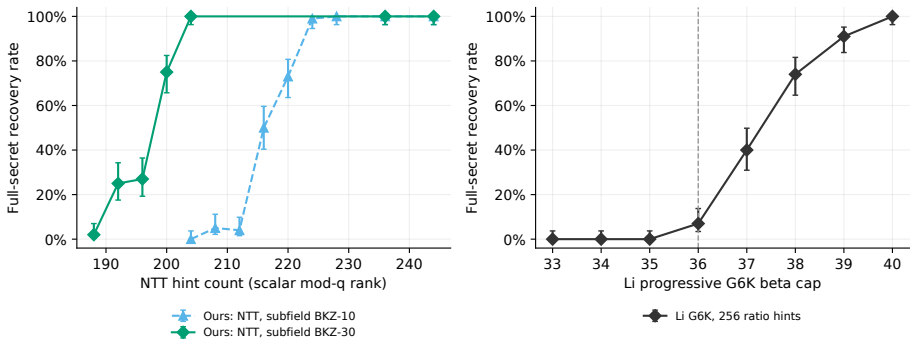

Fig. 2. Full-secret recovery transitions for our NTT subfield BKZ and the Li progressive G6K implementation, on the same 100 public inputs. The NTT curves use independent fixed block sizes on the 128-dimensional subfield lattice (hint axis, scalar mod-q rank); the Li horizontal coordinate is the progressive G6K schedule cap, not an equivalent fixed BKZ block size, and all Li trials use 256 ratio equations. Error bars are pointwise 95% Wilson intervals.

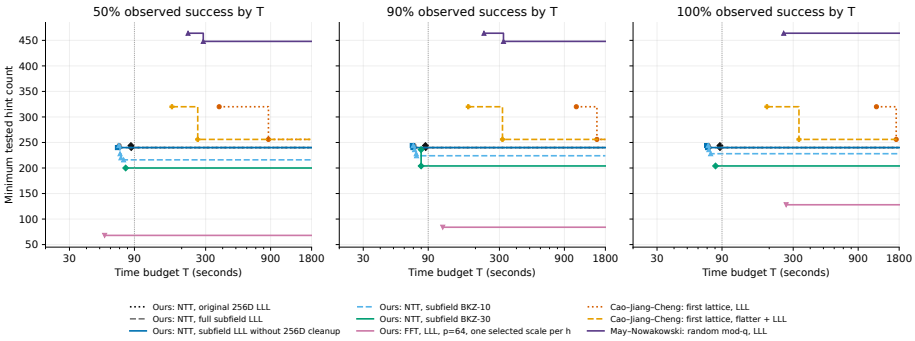

Fig. 3. Minimum tested hint count achieving 50%, 90%, and 100% observed full-secret recovery within time budget T. The feasibility statistic is $\widehat{P}_{m,h}(T)$, including every failure in the denominator. For every method, success and t_i come from the same single fixed configuration run once; no fallback or cross-h times are summed, and the FFT multi-configuration success curve is excluded. Time includes basis construction, one offline preprocessing cost where applicable, reduction, and public decoding, excluding separately recorded audits and synthetic input generation. The curves aggregate observed wall times from separate batches with different concurrent loads, not a controlled speedup measurement; blank regions mean that no tested configuration was feasible.