Assessing the Impact of a Variant of MATZOV's Attack

Kevin Carrier[1], Charles Meyer-Hilfiger[2], Yixin Shen[3], and Jean-Pierre Tillich[2]

[1] ETIS UMR 8051,CY Cergy-Paris Université, ENSEA, CNRS, kevin.carrier@cyu.fr
[2] Project COSMIQ, Inria de Paris,
charles.meyer-hilfiger@inria.fr,jean-pierre.tillich@inria.fr
[3] Univ Rennes, Inria, CNRS, IRISA, Rennes, France yixin.shen@inria.fr [⋆]

Abstract. The dual attacks on the Learning With Errors (LWE) problem are currently a subject of controversy. In particular, the results of [MAT22], which claim to significantly lower the security level of Kyber [SAB+20], a lattice-based cryptosystem currently being standardized by NIST, are not widely accepted. The analysis behind their attack depends on a series of assumptions that, in certain scenarios, have been shown to contradict established theorems or well-tested heuristics [DP23b].

In this paper, we introduce a new dual lattice attack on LWE, drawing from ideas in coding theory. Our approach revisits the dual attack proposed by [MAT22], replacing modulus switching with an efficient decoding algorithm. This decoding is achieved by generalizing polar codes over $\mathbb{Z}_q$, and we confirm their strong distortion properties through benchmarks. This modification enables a reduction from small-LWE to plain-LWE, with a notable decrease in the secret dimension. Additionally, we replace the enumeration step in the attack by assuming the secret is zero for the portion being enumerated, iterating this assumption over various choices for the enumeration part.

We make an analysis of our attack without using the flawed independence assumptions used in [MAT22] and we fully back up our analysis with experimental evidences.

Lastly, we assess the complexity of our attack on Kyber; showing that the security levels for Kyber-512/768/1024 are 3.5/11.9/12.3 bits below the NIST requirements (143/207/272 bits) in the same nearest-neighbor cost model as in [SAB+20,MAT22]. All in all the cost of our attack matches and even slightly beat in some cases the complexities originally claimed by the attack of [MAT22].

Keywords: Post-Quantum Cryptography · Learning With Errors · Dual Attack · Fast Fourier Transform · Polar Code

1 Introduction

1.1 Background

The LWE Problem. In this paper, we address the Learning With Errors (LWE) problem originally introduced by Regev in [Reg05]. It revolves around the task of finding $\mathbf{s} \in \mathbb{Z}_q^n$ given $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ with $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ where $\mathbf{e}$ is of small Euclidean norm. This problem can be seen as the decoding problem for the code $\mathcal{C}(\mathbf{A})$ generated by the columns of $\mathbf{A}$ (*i.e.* $\mathcal{C}(\mathbf{A}) \overset{\triangle}{=} \{\mathbf{A}\mathbf{x},\ \mathbf{x} \in \mathbb{Z}_q^n\}$) and the Euclidean distance, where we are asked to find the codeword (*i.e.* the element of $\mathcal{C}$) which is close in Euclidean distance to $\mathbf{b}$. We are particularly interested in the case where $\mathbf{s}$ is short too which is called the small LWE problem. This problem has emerged as a fundamental challenge in cryptography. Notably, it underpins the construction of various cryptographic primitives and is conjectured to withstand attacks from quantum computers [LPR10]. Our motivation for exploring this problem stems particularly from the need to gauge the security level of Kyber, a lattice-based cryptosystem that is being standardized by the NIST[4].

[⋆] The work of KC and JPT was funded by the French Agence Nationale de la Recherche through ANR JCJC DECODE (ANR-22-CE39-0004-01) for KC and ANR-22-PETQ-0008 PQ-TLS for JPT. The work of CMH was funded by the French Agence de l'innovation de défense and by Inria. The work of YS was funded by EPSRC grant EP/W02778X/2 and the France 2030 program managed by the French National Research Agency under grant agreement ANR-22-PETQ-0008 PQ-TLS.

[4] https://csrc.nist.gov/projects/post-quantum-cryptography

Dual Attacks. The most efficient cryptanalysis techniques against LWE(-like) problems are "primal" and "dual" lattice attacks. The primal attack corresponds to lattice reduction being performed on the "primal" lattice $\Lambda_q(\mathbf{A})$ which is Construction A of the q-ary lattice obtained from $\mathcal{C}(\mathbf{A})$, namely

$$\begin{aligned}\Lambda_q(\mathbf{A}) &= \{\mathbf{y} \in \mathbb{Z}^m \ : \ \exists \mathbf{c} \in \mathcal{C}(\mathbf{A}) \text{ such that } \mathbf{y} = \mathbf{c} \mod q\} \\ &= \{\mathbf{y} \in \mathbb{Z}^m \ : \ \exists \mathbf{s} \in \mathbb{Z}_q^n \text{ such that } \mathbf{y} = \mathbf{A}\mathbf{s} \mod q\}.\end{aligned}$$

Dual attacks mean that lattice reduction is performed over the dual lattice $\Lambda_q(\mathbf{A})^\vee$, which in this case, up to a q-multiplicative factor, is nothing but Construction A applied to the dual code $\mathcal{C}(\mathbf{A})^\perp \stackrel{\triangle}{=} \{\mathbf{x} \in \mathbb{Z}_q^m \ : \ \mathbf{A}^\top \mathbf{x} = \mathbf{0}\}$:

$$\begin{aligned}\Lambda_q(\mathbf{A})^\vee &\stackrel{\triangle}{=} \{\mathbf{x} \in \mathbb{R}^m \ : \ \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}, \ \forall \mathbf{y} \in \Lambda_q(\mathbf{A})\} \\ &= \tfrac{1}{q}\{\mathbf{x} \in \mathbb{Z}^m \ : \ \mathbf{A}^\top \mathbf{x} = \mathbf{0} \mod q\}.\end{aligned}$$

The lattice $\{\mathbf{x} \in \mathbb{Z}^m \ : \ \mathbf{A}^\top\mathbf{x} = \mathbf{0} \mod q\}$ is known under the name of the orthogonal lattice of $\mathbf{A}$, *i.e.* $\Lambda_q^\perp(\mathbf{A}) \stackrel{\triangle}{=} \{\mathbf{x} \in \mathbb{Z}^m \ : \ \mathbf{A}^\top\mathbf{x} = \mathbf{0} \mod q\}$.

Dual attacks were introduced in [MR09]. In its simplest form, a dual attack is a distinguisher attack which is given either $(\mathbf{A}, \mathbf{A}\cdot\mathbf{s}+\mathbf{e})$ or $(\mathbf{A},\mathbf{u})$ where $(\mathbf{A},\mathbf{u})$ are uniform and $\mathbf{e}$ is short, and answers if we are in the first or the second case. It starts by computing many short $\mathbf{x}_j$'s in the dual lattice $\Lambda_q(\mathbf{A})^\vee$ and the associated $\langle \mathbf{x}_j, \mathbf{b}\rangle$'s. Those short vectors are obtained by lattice reduction of $\Lambda_q(\mathbf{A})^\vee$ or what is basically equivalent $\Lambda_q^\perp(\mathbf{A})$. In the second case, we expect that these scalar products are uniformly distributed in $\mathbb{Z}_q$. On the other hand in the first case since $\langle \mathbf{x}_j, \mathbf{b}\rangle = \langle \mathbf{x}_j, \mathbf{A}\mathbf{s}+\mathbf{e}\rangle = \langle \mathbf{A}^\top\mathbf{x}_j, \mathbf{s}\rangle + \langle \mathbf{x}_j, \mathbf{e}\rangle = \langle \mathbf{x}_j, \mathbf{e}\rangle \mod q$, we get scalar products of small vectors which are tilted towards small entries.

There have been a sequence of developments in dual attacks, for instance by combining these attacks with a guessing stage [Alb17] consisting in splitting the support of $\mathbf{s}$ in two parts $\begin{pmatrix}\mathbf{s}_{\mathsf{enu}}\\ \mathbf{s}_{\mathsf{lat}}\end{pmatrix}$ and guessing one part of this support. $\mathbf{A}$ is split accordingly $\mathbf{A} = [\mathbf{A}_{\mathsf{enu}} \ \ \mathbf{A}_{\mathsf{lat}}]$. This allows to only perform lattice reduction on Construction A of the code generated by $\begin{bmatrix}\mathbf{Id}_m\\ \mathbf{A}_{\mathsf{lat}}^\top\end{bmatrix}$. That means we only look for short vectors $\begin{pmatrix}\mathbf{x}\\ \mathbf{y}_{\mathsf{lat}}\end{pmatrix}$ such that $\mathbf{A}_{\mathsf{lat}}{}^\top\mathbf{x} = \mathbf{y}_{\mathsf{lat}} \mod q$. Define $\mathbf{y}_{\mathsf{enu}}$ by $\mathbf{y}_{\mathsf{enu}} = \mathbf{A}_{\mathsf{enu}}^\top\mathbf{x}$. The point is that looking for such vectors is faster because of the dimension reduction. We then guess $\mathbf{s}_{\mathsf{enu}}$ and check whether the $\langle \mathbf{x}, \mathbf{b}\rangle - \langle \mathbf{y}_{\mathsf{enu}}, \mathbf{s}_{\mathsf{enu}}\rangle$'s are tilted towards small values or not. This comes from the fact that

$$\begin{aligned}\langle \mathbf{x}, \mathbf{b}\rangle &= \langle \mathbf{x}, \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} + \mathbf{A}_{\mathsf{lat}}\mathbf{s}_{\mathsf{lat}} + \mathbf{e}\rangle = \langle \mathbf{A}_{\mathsf{enu}}^\top\mathbf{x}, \mathbf{s}_{\mathsf{enu}}\rangle + \langle \mathbf{A}_{\mathsf{lat}}^\top\mathbf{x}, \mathbf{s}_{\mathsf{lat}}\rangle + \langle \mathbf{x}, \mathbf{e}\rangle \\ &= \langle \mathbf{y}_{\mathsf{enu}}, \mathbf{s}_{\mathsf{enu}}\rangle + \langle \mathbf{y}_{\mathsf{lat}}, \mathbf{s}_{\mathsf{lat}}\rangle + \langle \mathbf{x}, \mathbf{e}\rangle \mod q \qquad (1.1)\end{aligned}$$

In [EJK20] this was generalized to broader secret distributions paired with additional improvements on the exhaustive search. [GJ21] applied a Fast Fourier Transform-style algorithm to the search over $\mathbf{s}_{\mathsf{enu}}$ and the search space is significantly reduced by roughly considering only the most significant bits of $\mathbf{s}_{\mathsf{enu}}$. [MAT22] replaced this step with "modulus switching" [BV11,AFFP14], yielding significant performance gains. The algorithm can be described as follows.

The MATZOV template. Let $n_{\mathsf{enu}}, n_{\mathsf{fft}}, n_{\mathsf{lat}}$ be some positive integers such that $n_{\mathsf{enu}}+n_{\mathsf{fft}}+n_{\mathsf{lat}} = n$. The matrix $\mathbf{A}$ and the secret $\mathbf{s}$ are divided accordingly:

$$\mathbf{A} \stackrel{\triangle}{=} \begin{bmatrix}\mathbf{A}_{\mathsf{enu}} & \mathbf{A}_{\mathsf{fft}} & \mathbf{A}_{\mathsf{lat}}\end{bmatrix} \in \mathbb{Z}_q^{m\times n_{\mathsf{enu}}} \times \mathbb{Z}_q^{m\times n_{\mathsf{fft}}} \times \mathbb{Z}_q^{m\times n_{\mathsf{lat}}}, \qquad (1.2)$$

$$\mathbf{s} \stackrel{\triangle}{=} \begin{pmatrix}\mathbf{s}_{\mathsf{enu}}\\ \mathbf{s}_{\mathsf{fft}}\\ \mathbf{s}_{\mathsf{lat}}\end{pmatrix} \in \mathbb{Z}_q^{n_{\mathsf{enu}}} \times \mathbb{Z}_q^{n_{\mathsf{fft}}} \times \mathbb{Z}_q^{n_{\mathsf{lat}}}. \qquad (1.3)$$

The objective is to guess the part $\mathbf{s}_{\mathsf{enu}}$ of the secret vector but to do that, we need to be able to distinguish between a right and a wrong guess. By performing lattice reduction on the lattice generated by $\begin{bmatrix} \mathbf{Id}_m & \mathbf{0} \\ \mathbf{A}_{\mathsf{lat}}^\top & q\mathbf{Id}_{n_{\mathsf{lat}}} \end{bmatrix}$, we obtain a set $\mathscr{S}$ of vectors $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$ such that $\mathbf{x}$ and $\mathbf{y}$ are of small Euclidean norm in $\mathbb{Z}_q^m$ and $\mathbb{Z}_q^{n_{\mathsf{lat}}}$ respectively and are such that $\mathbf{y} = \mathbf{A}_{\mathsf{lat}}^\top \mathbf{x}$. Similarly to Equation (1.1), we have

$$\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} \rangle - \left\langle \mathbf{A}_{\mathsf{fft}}^\top \mathbf{x}, \mathbf{s}_{\mathsf{fft}} \right\rangle = \langle \mathbf{x}, \mathbf{e} \rangle + \langle \mathbf{y}, \mathbf{s}_{\mathsf{lat}} \rangle . \tag{1.4}$$

Since $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{e}, \mathbf{s}_{\mathsf{lat}})$ are short, Equation (1.4) is biased towards zero, raising the issue of finding efficiently $\mathbf{s}_{\mathsf{enu}}$ and $\mathbf{s}_{\mathsf{fft}}$ such that the left-hand term in (1.4) is small.

One can notice that (1.4) actually gives us many small-LWE sample[5] $(\mathbf{a}', b') \in \mathbb{Z}_q^{n_{\mathsf{fft}}} \times \mathbb{Z}_q$ with secret $\mathbf{s}' \in \mathbb{Z}_q^{n_{\mathsf{fft}}}$ and error $e' \in \mathbb{Z}_q$ where

$$\mathbf{a}' \triangleq \mathbf{A}_{\mathsf{fft}}^\top \mathbf{x}, \tag{1.5}$$

$$b' \triangleq \langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} \rangle , \tag{1.6}$$

$$\mathbf{s}' \triangleq \mathbf{s}_{\mathsf{fft}}, \tag{1.7}$$

$$e' \triangleq \langle \mathbf{x}, \mathbf{e} \rangle + \langle \mathbf{y}, \mathbf{s}_{\mathsf{lat}} \rangle . \tag{1.8}$$

Thus, we can distinguish a correct and an incorrect guess for $\mathbf{s}_{\mathsf{enu}}$ and recover $\mathbf{s}_{\mathsf{fft}}$ by solving this small-LWE instance. Since n_{enu} is small enough, $\mathbf{s}_{\mathsf{enu}}$ can be exhaustively searched. Once $\mathbf{s}_{\mathsf{enu}}$ is identified, $\mathbf{s}_{\mathsf{fft}}$ is recovered by solving the above small-LWE instance. Specifically, this involves exhaustively searching for a $\mathbf{z} \in \mathbb{Z}_q^{n_{\mathsf{fft}}}$ such that the vector $\left(\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} \rangle - \left\langle \mathbf{A}_{\mathsf{fft}}^\top \mathbf{x}, \mathbf{z} \right\rangle\right)_{(\mathbf{x},\mathbf{y}) \in \mathscr{S}}$ has a small Euclidean norm. One approach to speed up the search is to use a Fourier transform. The method introduces an evaluation function E, which assigns a real value to each guess $\widetilde{\mathbf{s}_{\mathsf{enu}}} \in \mathbb{Z}_q^{n_{\mathsf{enu}}}$:

$$E(\widetilde{\mathbf{s}_{\mathsf{enu}}}) \triangleq \max_{\mathbf{z} \in \mathbb{Z}_q^{n_{\mathsf{fft}}}} F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}(\mathbf{z}) \tag{1.9}$$

where

$$F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}(\mathbf{z}) \triangleq \sum_{(\mathbf{x},\mathbf{y}) \in \mathscr{S}} \cos\left(\frac{2\pi}{q} \left(\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\widetilde{\mathbf{s}_{\mathsf{enu}}} \rangle - \left\langle \mathbf{A}_{\mathsf{fft}}^\top \mathbf{x}, \mathbf{z} \right\rangle \right) \right) . \tag{1.10}$$

MATZOV's algorithm essentially consists in finding $\widetilde{\mathbf{s}_{\mathsf{enu}}} \in \mathbb{Z}_q^{n_{\mathsf{enu}}}$ such that $E(\widetilde{\mathbf{s}_{\mathsf{enu}}}) \geq T$, where $T \in \mathbb{N}$ is a threshold chosen around the expected value of E when evaluated on the correct guess, namely $T \approx E(\mathbf{s}_{\mathsf{enu}})$. The key idea is that we expect $E(\mathbf{s}_{\mathsf{enu}})$ to be significantly larger than $E(\widetilde{\mathbf{s}_{\mathsf{enu}}})$ when $\widetilde{\mathbf{s}_{\mathsf{enu}}}$ is incorrect, *i.e.* when $\widetilde{\mathbf{s}_{\mathsf{enu}}} \neq \mathbf{s}_{\mathsf{enu}}$. Indeed, for the correct guess, Equation (1.4) shows that $\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} \rangle - \left\langle \mathbf{A}_{\mathsf{fft}}^\top \mathbf{x}, \mathbf{s}_{\mathsf{fft}} \right\rangle \bmod q$ is biased toward zero. Consequently, each term in the sum in Equation (1.10) will be biased toward 1, resulting in a large total value for $\mathbb{E}\left(F_{\mathbf{s}_{\mathsf{enu}}}(\mathbf{s}_{\mathsf{fft}})\right)$. On the other hand, if $\widetilde{\mathbf{s}_{\mathsf{enu}}} \neq \mathbf{s}_{\mathsf{enu}}$, then for all $\mathbf{z} \in \mathbb{Z}_q^{n_{\mathsf{fft}}}$, $\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\widetilde{\mathbf{s}_{\mathsf{enu}}} \rangle - \left\langle \mathbf{A}_{\mathsf{fft}}^\top \mathbf{x}, \mathbf{z} \right\rangle \bmod q$ is uniformly distributed over $\mathbb{Z}_q$, meaning that $\mathbb{E}\left(F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}(\mathbf{z})\right) = 0$.

Modulus switching. The point of using the evaluation function E is that $F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}$ can be computed efficiently with a Fast Fourier Transform (FFT). However, the large input space $\mathbb{Z}_q^{n_{\mathsf{fft}}}$ makes the FFTcostly. This size can be reduced, though at the cost of slightly weakening the bias of $F_{\mathbf{s}_{\mathsf{enu}}}(\mathbf{s}_{\mathsf{fft}})$.

To do that, MATZOV proposes to reduce the size of the field $\mathbb{Z}_q$ by using a *modulus switching* technique: instead of considering $F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}$, they consider

$$\begin{aligned} F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}^{(\mathsf{ms})} : \mathbb{Z}_p^{n_{\mathsf{fft}}} &\longrightarrow \mathbb{R} \\ \mathbf{z} &\longmapsto \sum_{(\mathbf{x},\mathbf{y}) \in \mathscr{S}} \cos\left(\frac{2\pi}{p} \left(\frac{p}{q} \langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\widetilde{\mathbf{s}_{\mathsf{enu}}} \rangle - \left\langle \left\lfloor \frac{p}{q} \mathbf{A}_{\mathsf{fft}}^\top \mathbf{x} \right\rceil, \mathbf{z} \right\rangle \right) \right) \end{aligned}$$

[5] By using N samples $\{(\mathbf{a}'_\mathbf{i}, b'_i)\}_{i \in [\![1,N]\!]}$, we can rewrite the LWE instance with a matrix $\mathbf{A}' \triangleq \left[\mathbf{a}'_\mathbf{1} \cdots \mathbf{a}'_\mathbf{N}\right]$ and a vector $\mathbf{b}' \triangleq (b'_1 \cdots b'_N)^\top$.

where $p \leq q$ is a smaller modulus and $\lfloor \cdot \rceil$ stands for the integer rounding operation. Thus, for the wrong guess $\widetilde{\mathbf{s}_{\mathsf{enu}}} \neq \mathbf{s}_{\mathsf{enu}}$ and for all $\mathbf{z} \in \mathbb{Z}_p^{n_{\mathsf{fft}}}$, $F^{(\mathsf{ms})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}(\mathbf{z})$ is still expected to be 0 whereas for the good guess $\mathbf{s}_{\mathsf{enu}}$ we have, from Equation (1.4),

$$\begin{aligned} F^{(\mathsf{ms})}_{\mathbf{s}_{\mathsf{enu}}}(\mathbf{s}_{\mathsf{fft}} \bmod p) &= \sum_{(\mathbf{x},\mathbf{y}) \in \mathscr{S}} \cos\left(\frac{2\pi}{p}\left(\frac{p}{q}\langle \mathbf{x}, \mathbf{e}\rangle + \frac{p}{q}\langle \mathbf{y}, \mathbf{s}_{\mathsf{lat}}\rangle + \left\langle \frac{p}{q}\mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} - \left\lfloor \frac{p}{q}\mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} \right\rceil, \mathbf{s}_{\mathsf{fft}} \right\rangle\right)\right) \\ &= \sum_{(\mathbf{x},\mathbf{y}) \in \mathscr{S}} \cos\left(\frac{2\pi}{q}\left(\langle \mathbf{x}, \mathbf{e}\rangle + \langle \mathbf{y}, \mathbf{s}_{\mathsf{lat}}\rangle + \left\langle \mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} - \frac{q}{p}\left\lfloor \frac{p}{q}\mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} \right\rceil, \mathbf{s}_{\mathsf{fft}} \right\rangle\right)\right). \end{aligned}$$

Even if it means reducing the threshold a little, we can still expect $F^{(\mathsf{ms})}_{\mathbf{s}_{\mathsf{enu}}}(\mathbf{s}_{\mathsf{fft}} \bmod p) \geq T$ since the additional term $\left\langle \mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} - \frac{q}{p}\left\lfloor \frac{p}{q}\mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} \right\rceil, \mathbf{s}_{\mathsf{fft}} \right\rangle$ is also biased toward zero. Indeed, it is the scalar product of two short vectors; in particular:

$$d_{\mathsf{ms}} \triangleq \mathbb{E}\left(\left\| \mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} - \frac{q}{p}\left\lfloor \frac{p}{q}\mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} \right\rceil \right\|\right) \approx \frac{q}{p}\sqrt{\frac{n_{\mathsf{fft}}}{12}} \tag{1.11}$$

since we can make the approximation that each of the coordinates of $\frac{p}{q}\mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} - \left\lfloor \frac{p}{q}\mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} \right\rceil$ are drawn uniformly at random in $]-0.5, 0.5]$.

Finally, [MAT22] claims that the security level of NIST candidates like Kyber could be significantly lowered. However, this result is not widely accepted, as the analysis relies on assumptions which turn out to be false according to [DP23b].

Dual Attacks in Code-Based Cryptography and Analyzing Dual Attacks. Dual attacks in lattice based cryptography can be viewed as the lattice based analogue of statistical decoding in code-based cryptography which dates back to Al Jabri [Jab01]. In this case, short codewords $\mathbf{h}$ in the dual code are computed, where short is with respect to the Hamming distance. The goal is to solve the decoding problem $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ where $\mathbf{b}$ and $\mathbf{A}$ are given whereas $\mathbf{s}$ and $\mathbf{e}$ are unknown and $\mathbf{e}$ is of small Hamming weight (rather than of small Euclidean norm for the LWE problem). The issue is to recover $\mathbf{s}$. Here too, the inner product $\langle \mathbf{h}, \mathbf{b}\rangle$ is biased towards zero. This is used to solve the decoding problem very much in the same way as it is used to solve the LWE problem. Similarly to what happened in lattice based cryptography, dual attacks became much more effective through a splitting strategy which allowed to look for short codewords in a smaller code [CDMT22]. In an analogous way to what was done in lattice based cryptography, these attacks were analyzed by making assumptions and in particular independence assumptions [CDMT22, Ass. 3.7] which are close to the independence assumptions made for analyzing MATZOV's attack [MAT22, Ass. 4.4, Ass. 5.8]. In the lattice based case, these assumptions were shown to contradict some theorems in certain regimes or well-tested heuristics in some other regimes [DP23b].

Note that it was already noticed in [CDMT22, §3.4] that the i.i.d. Bernoulli model used for analyzing dual attacks in code-based cryptography is not always accurate. However, it was conjectured there that the difference between this ideal model and experiments has no impact on the asymptotic analysis of the decoding based on this model. This was proved to be wrong in [MT23]. However, this paper gave at the same time an approach for analyzing rigorously dual attacks in coding theory by bringing in a duality equation [MT23, Prop. 1.3] shedding some light on the fundamental quantities manipulated by the decoder. This allowed to obtain a proof of the correctness of a slightly modified version of the dual attack proposed in [CDMT22]. Dual attacks for solving the decoding problem have been further improved in [CDMT24] and the approach of [MT23] has been carried over to this improved dual attack. [CDMT24] can be viewed as a somewhat improved version of the MATZOV attack in the context of codes, where the modulus switching part is replaced by an optimal lossy source encoder. It is worthwhile to note that [CDMT24, Section 8] shows that the fundamental duality equation [MT23, Prop. 1.3] used to analyze dual attacks for codes also carries over to the lattice setting and could serve as a tool to analyze dual attacks for lattices. Moreover, at the same time, a series of papers [DP23a,PS24] provide some new ideas to properly analyze dual attacks in lattices. In particular, concurrently to [CDMT24, Section 8], [DP23a] provided, in a more in-depth work and using comparable but not identical reasoning, similar heuristics to predict the behavior of these attacks.

1.2 Our Contribution

Our purpose here is to come up with a variation of the MATZOV algorithm, that improves it and which also helps analyzing it. This is obtained by a lattice based analogue of [CDMT24], replacing the modulus switching by lossy source encoding/using the relevant quantizer based on polar coding. We use the duality approach [MT23,CDMT24,DP23a] to analyze our new dual attack to avoid relying on independence assumptions. This duality approach allows us to derive a new simple heuristic that is backed up by experimental evidence and which is essentially a generalization of the heuristics made in [CDMT24,DP23a]. All in all, it turns out that the complexities claimed in [MAT22] can be achieved and even slightly surpassed with our algorithm, which dents the parameters of Kyber by a few bits, as predicted in [MAT22].

A Lossy Source Coding/Quantizer Approach. Similarly to [CDMT24] we observe that the FFT approach factorizing the common computations for computing all the $F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}(\mathbf{z})$ for $\mathbf{z} \in \mathbb{Z}_q^{n_{\mathsf{fft}}}$ is probably suboptimal, since the $\mathbf{z}$ such that $F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}(\mathbf{z})$ is maximum is likely to be attained for $\mathbf{z} = \mathbf{s}_{\mathsf{fft}}$ which is of rather small norm. The problem is that the fast FFT algorithm doesn't leverage the fact that we only need to compute it for small $\mathbf{z}$'s which have the same Euclidean norm as $\mathbf{s}_{\mathsf{fft}}$. In a sense, the modulus switching approach of [MAT22] is a way to alleviate this phenomenon since $\mathbf{s}_{\mathsf{fft}} \mod p$ is more uniformly distributed in $\mathbb{Z}_p^{n_{\mathsf{fft}}}$ than $\mathbf{s}$ is in $\mathbb{Z}_q^{n_{\mathsf{fft}}}$. A further refinement of this method involves approximating the relevant $\mathbf{z} \in \mathbb{Z}_q^{n_{\mathsf{fft}}}$'s by close enough codewords. This also allows to reduce the size $q^{n_{\mathsf{fft}}}$ of the space over which the fast Fourier transform is applied.

Basically the lossy source/quantizing approach can be explained as follows. We choose a linear code $\mathcal{C}_{\mathsf{lsc}}$ generated by the matrix $\mathbf{G} \in \mathbb{Z}_q^{n_{\mathsf{fft}} \times k_{\mathsf{fft}}}$, *i.e.* $\mathcal{C}_{\mathsf{lsc}} \triangleq \{\mathbf{G}\mathbf{u}_{\mathsf{lsc}} : \mathbf{u}_{\mathsf{lsc}} \in \mathbb{Z}_q^{k_{\mathsf{fft}}}\}$ so that we can find efficiently, for any $\mathbf{y}_{\mathsf{fft}} \in \mathbb{Z}_q^{n_{\mathsf{fft}}}$, a $\mathbf{u}_{\mathsf{lsc}} \in \mathbb{Z}_q^{k_{\mathsf{fft}}}$ such that $\mathbf{G}\mathbf{u}_{\mathsf{lsc}}$ is close to $\mathbf{y}_{\mathsf{fft}}$. $\mathbf{G}\mathbf{u}_{\mathsf{lsc}}$ can be viewed as a "quantization" of $\mathbf{y}_{\mathsf{fft}}$ and $\mathcal{C}_{\mathsf{lsc}}$ as a lossy source code or code used for quantization. We apply this quantization to all pairs of short dual vectors $(\mathbf{x}, \mathbf{y})$ in $\mathscr{S}$ and compute for all such $\mathbf{x}$'s a corresponding $\mathbf{u}_{\mathsf{lsc}} \in \mathbb{Z}_q^{k_{\mathsf{fft}}}$ such that

$$\left\| \mathbf{A}_{\mathsf{fft}}^\top \mathbf{x} - \mathbf{G}\mathbf{u}_{\mathsf{lsc}} \right\| \approx d_{\mathsf{lsc}} \tag{1.12}$$

where d_{lsc} is the decoding distance of the lossy source code. The point is that the left hand term in (1.4) can be rewritten as

$$\begin{aligned} \langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} \rangle - \left\langle \mathbf{A}_{\mathsf{fft}}^\top \mathbf{x}, \mathbf{s}_{\mathsf{fft}} \right\rangle &= \langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} \rangle - \langle \mathbf{G}\mathbf{u}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}} \rangle - \langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}} \rangle && (1.13) \\ &= \langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} \rangle - \left\langle \mathbf{u}_{\mathsf{lsc}}, \mathbf{G}^\top \mathbf{s}_{\mathsf{fft}} \right\rangle - \langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}} \rangle && (1.14) \end{aligned}$$

where $\mathbf{e}_{\mathsf{lsc}} \triangleq \mathbf{A}_{\mathsf{fft}}^\top \mathbf{x} - \mathbf{G}\mathbf{u}_{\mathsf{lsc}}$. If we use (1.4) we see that

$$\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} \rangle - \left\langle \mathbf{u}_{\mathsf{lsc}}, \mathbf{G}^\top \mathbf{s}_{\mathsf{fft}} \right\rangle = \langle \mathbf{x}, \mathbf{e} \rangle + \langle \mathbf{y}, \mathbf{s}_{\mathsf{lat}} \rangle + \langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}} \rangle . \tag{1.15}$$

So we expect that the left-hand term $\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} \rangle - \left\langle \mathbf{u}_{\mathsf{lsc}}, \mathbf{G}^\top \mathbf{s}_{\mathsf{fft}} \right\rangle$ is still small. In other words, we once again performed a reduction to an LWE problem. Indeed, Equation (1.15) can be interpreted as an LWE sample $(\mathbf{a}', b') \in \mathbb{Z}_q^{k_{\mathsf{fft}}} \times \mathbb{Z}_q$, with secret $\mathbf{s}' \in \mathbb{Z}_q^{k_{\mathsf{fft}}}$ and error term $e' \in \mathbb{Z}_q$, where

$$\mathbf{a}' \triangleq \mathbf{u}_{\mathsf{lsc}}, \tag{1.16}$$

$$b' \triangleq \langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}} \rangle , \tag{1.17}$$

$$\mathbf{s}' \triangleq \mathbf{G}^\top \mathbf{s}_{\mathsf{fft}}, \tag{1.18}$$

$$e' \triangleq \langle \mathbf{x}, \mathbf{e} \rangle + \langle \mathbf{y}, \mathbf{s}_{\mathsf{lat}} \rangle + \langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}} \rangle . \tag{1.19}$$

The advantage of this reduction is that it drastically reduces the dimension of the problem. However, it is important to note that the secret $\mathbf{s}'$ is no longer small, but uniformly distributed over $\mathbb{Z}_q^{k_{\mathsf{fft}}}$; thus, this new problem becomes a plain-LWE problem. We can now solve it using a technique similar to the one previously described. This motivates to replace $F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}$ by

$$\begin{aligned} F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}^{(\mathsf{lsc})} : \mathbb{Z}_q^{k_{\mathsf{fft}}} &\longrightarrow \mathbb{R} \\ \mathbf{z} &\longmapsto \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{S}} \cos\left(\tfrac{2\pi}{q}\left(\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\widetilde{\mathbf{s}_{\mathsf{enu}}}\rangle - \langle \mathbf{u}_{\mathsf{lsc}}, \mathbf{z}\rangle\right)\right). \end{aligned} \tag{1.20}$$

Again, for the wrong guess $\widetilde{\mathbf{s}_{\mathsf{enu}}} \neq \mathbf{s}_{\mathsf{enu}}$ and for all $\mathbf{z} \in \mathbb{Z}_p^{k_{\mathsf{fft}}}$, $F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}^{(\mathsf{lsc})}(\mathbf{z})$ is still expected to be 0 whereas for the good guess $\mathbf{s}_{\mathsf{enu}}$ we have, from Equation (1.15), that

$$F_{\mathbf{s}_{\mathsf{enu}}}^{(\mathsf{lsc})}\left(\mathbf{G}^{\top}\mathbf{s}_{\mathsf{fft}}\right) = \sum_{(\mathbf{x},\mathbf{y})\in\mathscr{S}} \cos\left(\tfrac{2\pi}{q}\left(\langle \mathbf{x}, \mathbf{e}\rangle + \langle \mathbf{y}, \mathbf{s}_{\mathsf{lat}}\rangle + \langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}}\rangle\right)\right) \tag{1.21}$$

which is still expected to be large because the additional term $\langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}}\rangle$ is biased towards zero.

The pro and cons of this approach is that on the positive side:
– It allows us to choose large values for n_{fft}, since we are now not limited by the $q^{n_{\mathsf{fft}}}$ term in the complexity coming from evaluating $F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}$ but by a smaller $q^{k_{\mathsf{fft}}}$ term. This in turn allows us to decrease the n_{lat} term and therefore the cost of lattice reduction.
– It allows us to solve radically the problem that $\mathbf{s}_{\mathsf{fft}}$ is not uniformly distributed in $\mathbb{Z}_q^{n_{\mathsf{fft}}}$. If k_{fft} is low enough, it can namely be verified that the likely argmax $\mathbf{G}^{\top}\mathbf{s}_{\mathsf{fft}}$ of $F_{\mathbf{s}_{\mathsf{enu}}}^{(\mathsf{lsc})}(\mathbf{z})$ is uniformly distributed in $\mathbb{Z}_q^{k_{\mathsf{fft}}}$. This solves one source of suboptimality of this approach.
On the negative side, it increases the noise term in the right-hand side of (1.15) expressing the quantity $\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\mathbf{s}_{\mathsf{enu}}\rangle - \left\langle \mathbf{u}_{\mathsf{lsc}}, \mathbf{G}^{\top}\mathbf{s}_{\mathsf{fft}}\right\rangle$. This is due to the additional term $\langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}}\rangle$ which appears there. We therefore expect $F_{\mathbf{s}_{\mathsf{enu}}}^{(\mathsf{lsc})}\left(\mathbf{G}^{\top}\mathbf{s}_{\mathsf{fft}}\right)$ to be smaller than $F_{\mathbf{s}_{\mathsf{enu}}}(\mathbf{s}_{\mathsf{fft}})$ and will need to make $\mathscr{S}$ bigger to distinguish it from other values of $F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}^{(\mathsf{lsc})}(\mathbf{z})$.

The last point above makes it clear that we want to make the additional noise term $\langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}}\rangle$ as small as possible. But of course for a given k_{fft}, there is a lower bound of what we can achieve. It can namely be proven (see Equation (2.11) in Section 2) that the best we can do is to choose, for a given k_{fft}, the decoding distance d_{lsc} such that

$$d_{\mathsf{lsc}} \approx q^{1-\frac{k_{\mathsf{fft}}}{n_{\mathsf{fft}}}} \cdot \sqrt{\frac{n_{\mathsf{fft}}}{2\pi e}}. \tag{1.22}$$

This can be compared to the modulus switching approach. For a same FFT complexity, meaning roughly that k_{fft} is such that $p^{n_{\mathsf{fft}}} = q^{k_{\mathsf{fft}}}$, we have

$$d_{\mathsf{ms}} \approx \frac{q}{p}\sqrt{\frac{n_{\mathsf{fft}}}{12}} \approx 0.28867\frac{q\sqrt{n_{\mathsf{fft}}}}{p} \tag{1.23}$$

$$d_{\mathsf{lsc}} \approx \frac{q}{p}\sqrt{\frac{n_{\mathsf{fft}}}{2\pi e}} \approx 0.24197\frac{q\sqrt{n_{\mathsf{fft}}}}{p} \tag{1.24}$$

Equations (1.23) and (1.24) show that the lossy source coding approach yields a smaller value for $\|\langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}}\rangle\|$ than modulus switching, as $d_{\mathsf{lsc}} < d_{\mathsf{ms}}$. However, unlike modulus switching, we need to construct a code that can be decoded efficiently up to the optimal decoding distance d_{lsc}. A classical approach involves using a Cartesian product of small random codes. This solution achieves asymptotically the optimal decoding distance d_{lsc} but with a sub-exponential complexity that is not so negligible (super-polynomial). In Section 3.3, we propose an alternative solution using polar codes, which can achieve a decoding distance very close to d_{lsc} in quasi-linear time.

It should be noted that the modulus switching strategy can really be viewed as a quantizing approach. In the modulus switching technique, we approximate $\mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x}$ by $\frac{q}{p}\left\lfloor \frac{p}{q}\mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x}\right\rceil$, *i.e.* we quantize/approximate a point in $\mathbb{R}^{n_{\mathsf{fft}}}$ by a point in the lattice $\frac{q}{p}\mathbb{Z}^{n_{\mathsf{fft}}}$. On the other hand, in the case of the lossy source code approach we approximate a point in $\mathbb{Z}_q^{n_{\mathsf{fft}}}$ by a codeword in $\mathcal{C}_{\mathsf{lsc}}$. If we express this as a quantizer, this means that we quantize/approximate a point in $\mathbb{R}^{n_{\mathsf{fft}}}$ by a lattice point in Construction A applied to $\mathcal{C}_{\mathsf{lsc}}$. The second quantizer just turns out to be much better than the first quantizer in terms of the distortion/quantizing distance which is achieved.

Getting Rid of Independence Assumptions in the Analysis and Results. In [MAT22], it is argued that dual attacks can substantially lower the security level of certain NIST candidates, such as Kyber. However, this claim remains contested, as their analysis relies on assumptions that, according to [DP23b], have been proven to be incorrect. Specifically, [DP23b] highlights a flawed independence assumption, which can be stated as follows:

Assumption 1.1 (Independence Assumption). *Let Λ be a full-rank lattice of dimension n, and let $\mathbf{r}$ be a random variable distributed according to* $\mathrm{Unif}(\mathbb{R}^n/\Lambda)$ *or ${\mathcal{B}_\alpha}^n$, where the $\mathcal{B}_\alpha$ are i.i.d. centered binomial variables. Assume that $\left(e^{2i\pi\langle \mathbf{w},\mathbf{r}\rangle}\right)_{\mathbf{w}\in\Lambda}$ are mutually independent.*

In dual attacks, evaluation functions involve sums of terms like those described in the assumption. Assuming independence when the terms are not independent can lead to significant miscalculations in estimating false positives passing the evaluation function. This issue is highlighted in [DP23b], which shows that predictions of such scoring functions are inaccurate in certain regimes.

Recent papers [DP23a,MT23,CDMT24] present new approaches to analyze dual attacks accurately, without relying on the independence assumption. Specifically, let

$$F(\mathbf{x}) \triangleq \sum_{\mathbf{w}\in\Lambda\cap\mathscr{B}} e^{2i\pi\langle \mathbf{w},\mathbf{r}(\mathbf{x})\rangle} \tag{1.25}$$

define a scoring function, where Λ is a full-rank lattice of dimension n, $\mathscr{B} \subseteq \mathbb{R}^n$ a set of small vectors, and $\mathbf{r}(\mathbf{x}) \in \mathbb{R}^n$, all depending on the specific dual attack method under consideration. It can be observed that $F(\mathbf{x})$ remains invariant when any vector from the dual lattice of Λ is added to $\mathbf{r}(\mathbf{x})$. Indeed, for any $\mathbf{w}^\vee \in \Lambda^\vee$, we have

$$\langle \mathbf{w}, \mathbf{r}(\mathbf{x}) + \mathbf{w}^\vee\rangle = \langle \mathbf{w}, \mathbf{r}(\mathbf{x})\rangle + \langle \mathbf{w}, \mathbf{w}^\vee\rangle = \langle \mathbf{w}, \mathbf{r}(\mathbf{x})\rangle\,. \tag{1.26}$$

Thus, it is reasonable to conclude that $\langle \mathbf{w}, \mathbf{r}(\mathbf{x})\rangle$ depends on the structure of the coset $\Lambda^\vee + \mathbf{r}(\mathbf{x})$, and in particular, on its shortest vector. One way to see this is by invoking the Poisson summation formula:

$$F(\mathbf{x}) = \frac{1}{\mathsf{Vol}\,(\Lambda)} \cdot \sum_{\mathbf{w}^\vee\in\Lambda^\vee+\mathbf{r}(\mathbf{x})} \widehat{\mathbb{1}_{\mathscr{B}}}\,(\mathbf{w}^\vee)\,. \tag{1.27}$$

In [DP23a,CDMT24], it was observed that the Fourier transform of the indicator function can be expressed in terms of Bessel functions, depending only on the input's length. Consequently, $F(\mathbf{x})$ is essentially related to the length enumerator of the vectors in $\Lambda^\vee + \mathbf{r}(\mathbf{x})$, particularly the shortest. We distinguish between correct and incorrect candidates by noting that $\mathbf{r}(\mathbf{x})$ is notably small when $\mathbf{x}$ is the desired vector, and random otherwise. Therefore, for the correct guess, the shortest vector in $\Lambda^\vee + \mathbf{r}(\mathbf{x})$ is $\mathbf{r}(\mathbf{x})$, while for incorrect guesses, it corresponds to the typical shortest vector in a random coset of a random lattice.

This approach allowed us to analyze our evaluation functions thoroughly (see Section 4). However, to simplify calculations, we make approximations that we validate through simulations. Finally, we control the number of false positives in our attack, ensuring they remain negligible. By doing so, we achieve results close to those of MATZOV (see Section 5). Furthermore, as they did previously, we apply our attack to the parameters of Kyber and confirm the claim in [MAT22], which asserts that Kyber's security does not meet the NIST's requirements.

2 Notation and Preliminaries

Basic Notation. We denote vectors by bold lowercase letters and matrices by bold uppercase letters, *e.g.* $\mathbf{v}$ and $\mathbf{M}$. We consider the vectors as column vectors and therefore row vectors are denoted $\mathbf{v}^\top$. The concatenation of vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted as $(\mathbf{x}, \mathbf{y})$. The components of the vector $\mathbf{x} \in \mathbb{R}^n$ are denoted by x_i for $i \in [\![1, n]\!]$ where $[\![a, b]\!]$ are the integers between a and b.

For any $x \in \mathbb{Z}_q$, denote by $\widehat{x} \in x + q\mathbb{Z}$ the unique integer such that $|\widehat{x}| \leq \frac{q-1}{2}$. We extend this notion to vectors $\mathbf{x} \in \mathbb{Z}_q^n$ componentwise. In other words, $\widehat{\mathbf{x}}$ is the lift from $\mathbb{Z}_q$ to $\mathbb{Z}$ centered on $\mathbf{0}$. We define $\|\mathbf{x}\|$ for $\mathbf{x} \in \mathbb{Z}_q^n$ as

$$\|\mathbf{x}\| \overset{\triangle}{=} \|\widehat{\mathbf{x}}\| \overset{\triangle}{=} \sqrt{\sum_{i=1}^{n} \widehat{x}_i^{\,2}} \overset{\triangle}{=} \sqrt{\sum_{i=1}^{n} \underset{j \in \mathbb{Z}}{\operatorname{argmin}} \left((x_i + jq)^2\right)} \tag{2.1}$$

The Learning With Error Problem. Let us define more formally the LWE problem here. It starts by defining an LWE oracle that produces samples according to the following distribution:

Definition 2.1 (LWE oracle). *Let $q, n, m \in \mathbb{N}$, and let χ_s, χ_e be distributions over $\mathbb{Z}_q$. We first draw $\mathbf{s} \in \mathbb{Z}_q^n$ with coordinates drawn independently from each other according to the distribution χ_s and then draw m LWE samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where the $\mathbf{a}_i$'s are drawn uniformly at random in $\mathbb{Z}_q^n$ and $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i$ where the e_i's are drawn independently according to the distribution χ_e. We let $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ be the matrix where the i-th row is $\mathbf{a}_i$. The pair $(\mathbf{A}, \mathbf{b})$ is the output of the oracle and satisfies $\mathbf{b} \overset{\triangle}{=} \mathbf{A}\mathbf{s} + \mathbf{e}$.*

We then define the Search-LWE problem as follows:

Problem 2.2 (Search-LWE). *Given a sample $(\mathbf{A}, \mathbf{b})$ drawn from an LWE$(q, n, m, \chi_s, \chi_e)$-oracle, the goal is to recover the secret vector $\mathbf{s}$.*

In the literature, the LWE oracle is sometimes defined by replacing the matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ by a vector $\mathbf{a} \in \mathbb{Z}_q^n$. Then the LWE problems are stated for an arbitrary number of calls to the oracle. If the LWE oracle is called m times, then the situation is actually the same as above.

The Centered Binomial Distribution. In the LWE oracle, the distributions χ_s and χ_e depend on the context. Historically, the secret vector $\mathbf{s}$ was distributed uniformly in $\mathbb{Z}_q^n$ and the noise vector $\mathbf{e}$ was short. It is quite common today to consider the case where $\mathbf{s}$ is also short; we are then talking about Small-LWE problem. In recent cryptosystems, particularly those involved in the NIST Post-Quantum Standardization Process, the distributions for χ_s and χ_e are centered binomial distributions. Note that in Kyber, the distribution for the secret vector and the error vector is the same.

Definition 2.3 (Centered Binomial Distribution). *The centered binomial distribution $\mathcal{B}_\alpha$ of parameter $\alpha \in [\![0, \frac{q-1}{2}]\!]$ is defined as $\mathcal{B}_\alpha \sim \sum_{i=1}^{\alpha}(X_i - Y_i)$ where the X_i's and Y_i's are i.i.d. as uniform over $\{0, 1\}$. In particular, for all $i \in [\![-\alpha, \alpha]\!]$, we have $\mathbb{P}(\mathcal{B}_\alpha = i) = 2^{-2\alpha}\binom{2\alpha}{\alpha+i}$. Note that $\mathcal{B}_\alpha$ has mean 0 and standard deviation $\sigma \overset{\triangle}{=} \sqrt{\frac{\alpha}{2}}$.*

Coding Background. To simplify the discussion, we assume that q is a prime number implying that $\mathbb{Z}_q$ has the structure of a finite field. A linear code $\mathcal{C}$ of length n and dimension k is a subspace of $\mathbb{Z}_q^n$ of dimension k. We say that $\mathcal{C}$ is an $[n, k]_q$-code and its rate is $R \overset{\triangle}{=} \frac{k}{n}$. A generator matrix $\mathbf{G}$ for $\mathcal{C}$ is a full rank matrix in $\mathbb{Z}_q^{n \times k}$ such that $\mathcal{C} = \left\{\mathbf{G}\mathbf{u} \ : \ \mathbf{u} \in \mathbb{Z}_q^k\right\}$, and a parity-check matrix $\mathbf{H}$ for $\mathcal{C}$ is a full rank matrix in $\mathbb{Z}_q^{(n-k) \times n}$ such that $\mathcal{C} = \left\{\mathbf{c} \in \mathbb{Z}_q^n \ : \ \mathbf{H}\mathbf{c} = \mathbf{0}\right\}$.

Definition 2.4 (Dual Code). *Let $\mathcal{C}$ be an $[n, k]_q$-code. The dual code $\mathcal{C}^\perp$ of $\mathcal{C}$ is defined as the following $[n, n-k]_q$-code:*

$$\mathcal{C}^\perp \overset{\triangle}{=} \left\{\mathbf{h} \in \mathbb{Z}_q^n \ : \ \forall \mathbf{c} \in \mathcal{C}, \ \langle \mathbf{c}, \mathbf{h} \rangle = 0\right\} \tag{2.2}$$

Lattice Background. A lattice Λ is a discrete subgroup of $\mathbb{R}^d$. In particular, for a matrix $\mathbf{B} \in \mathbb{R}^{d\times d}$, the lattice $\Lambda(\mathbf{B})$ is defined as:

$$\Lambda(\mathbf{B}) \stackrel{\triangle}{=} \left\{\mathbf{Bx} \ : \ \mathbf{x} \in \mathbb{Z}^d\right\} \tag{2.3}$$

The matrix $\mathbf{B}$ is called a basis if it has full column rank.

The volume of a full-rank lattice $\Lambda \subset \mathbb{R}^d$ is the Euclidean volume of its fundamental Voronoï region $\mathsf{Vor}(\Lambda) \stackrel{\triangle}{=} \{\mathbf{v} \in \mathbb{R}^d : \|\mathbf{v}\| \leq \|\mathbf{v} - \mathbf{x}\|\ , \ \forall \mathbf{x} \in \Lambda\}$. For a full-rank lattice $\Lambda(\mathbf{B})$, it is the volume of the fundamental d-dimensional parallelepiped defined by the column vectors of $\mathbf{B}$ (or any other basis of Λ). It is also the determinant of the matrix $\mathbf{B}$:

$$\mathsf{Vol}\left(\Lambda(\mathbf{B})\right) \stackrel{\triangle}{=} \mathsf{Vol}\left(\mathsf{Vor}\left(\Lambda(\mathbf{B})\right)\right) = \det\left(\mathbf{B}\right) \tag{2.4}$$

The dual $\Lambda^\vee$ of a lattice Λ is defined as

Definition 2.5 (Dual Lattice). *The dual $\Lambda^\vee$ of a lattice $\Lambda \subset \mathbb{R}^d$ is*

$$\Lambda^\vee \stackrel{\triangle}{=} \{\mathbf{x} \in \operatorname{span}(\Lambda) \ : \ \forall \mathbf{y} \in \Lambda, \ \langle \mathbf{x}, \mathbf{y}\rangle \in \mathbb{Z}\}\,. \tag{2.5}$$

We can construct a lattice from a linear code through Construction A:

Definition 2.6 (Construction A). *Let $\mathcal{C}$ be an $[n,k]_q$-code. The lattice Λ obtained by Construction A applied to $\mathcal{C}$ is given by*

$$\Lambda(\mathcal{C}) \stackrel{\triangle}{=} \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{x} \equiv \mathbf{c} \bmod q, \ \mathbf{c} \in \mathcal{C}\}\,. \tag{2.6}$$

Note that if $\mathcal{C}$ is defined by a systematic generator matrix $\mathbf{G} \stackrel{\triangle}{=} \begin{bmatrix}\mathbf{Id}_k \\ \mathbf{A}\end{bmatrix} \in \mathbb{Z}_q^{n\times k}$, then the lattice $\Lambda(\mathcal{C})$ that is obtained through Construction A is also the lattice $\Lambda(\mathbf{B})$ generated by

$$\mathbf{B} \stackrel{\triangle}{=} \left[\begin{array}{c|c}\mathbf{Id}_k & \mathbf{0} \\ \hline \mathbf{A} & q\mathbf{Id}_{n-k}\end{array}\right]. \tag{2.7}$$

Clearly finding the closest point in Euclidean distance to some $\mathbf{y} \in \mathbb{Z}_q^n$ in $\mathcal{C}$ also amounts to find the closest lattice point in $\Lambda(\mathcal{C})$ of $\mathbf{y}$. The algorithm for performing this task when $\mathbf{y}$ belongs to $\mathbb{R}^n$ is known as a mean-squared-error (MSE) quantizer for $\Lambda(\mathcal{C})$. To analyze its performance, first notice that the fundamental Voronoï region V of a lattice $\Lambda(\mathcal{C})$ associated to code $\mathcal{C}$ of dimension k over $\mathbb{Z}_q^n$ has volume $\mathsf{Vol}\left(V\right) \stackrel{\triangle}{=} \mathsf{Vol}\left(\Lambda(\mathcal{C})\right) = q^{n-k}$ [CS88]. The average decoding distance ω provided by the mean-square quantizer for Λ can be assessed by the normalized second moment $G \stackrel{\triangle}{=} G(\Lambda(\mathcal{C}))$, which is defined as

$$G \stackrel{\triangle}{=} \frac{1}{n \cdot \mathsf{Vol}\left(V\right)^{\frac{2}{n}}} \int_V \frac{\|\mathbf{v}\|^2}{\mathsf{Vol}\left(V\right)} d\mathbf{v}. \tag{2.8}$$

It is known that

$$G \geq \frac{1}{2\pi e} \tag{2.9}$$

and is achieved asymptotically for lattices generated by Construction A from q-ary random codes when q gets big [ZF96]. The case where the equality is achieved in Equation (2.9) really corresponds to the case when the equality

$$|\mathcal{C}| \cdot \mathsf{Vol}\left(\mathsf{Ball}_\omega^n\right) = q^n \tag{2.10}$$

is met, *i.e.* when the average decoding distance is

$$\omega \stackrel{\triangle}{=} \sqrt{\int_V \frac{\|\mathbf{v}\|^2}{\mathsf{Vol}\left(V\right)} d\mathbf{v}} = \sqrt{\frac{n}{2\pi e}} \cdot q^{1-\frac{k}{n}} + o(1)\,. \tag{2.11}$$

It corresponds therefore to the lattice analogue of the Gilbert-Varshamov distance.

Short Vector Sampler. Dual attacks heavily depend on lattice reduction algorithms, such as BKZ, to find short vectors in a lattice or its dual. Our improvements in this paper do not address these algorithms. Instead, we will use Algorithm 2.1 to obtain short vectors and refer the reader to [GJ21,MAT22] for implementation details.

Algorithm 2.1 Short Vectors Sampling Procedure [GJ21]

```
    Input: A basis $\mathbf{B} = [\mathbf{b}_0 \dots \mathbf{b}_{d-1}]$ for a lattice and $2 \leq \beta_{\mathsf{bkz}}, \beta_{\mathsf{sieve}} \in \mathbb{Z} \leq d$.
    Output: A list of $N_{\mathsf{sieve}}(\beta_{\mathsf{sieve}}) \triangleq \left(\sqrt{\frac{4}{3}}\right)^{\beta_{\mathsf{sieve}}}$ vectors from the lattice.
Randomize the basis $\mathbf{B}$.
Run $\mathsf{BKZ}$-$\beta_{\mathsf{bkz}}$ to obtain a reduced basis $\mathbf{b}'_0, \dots, \mathbf{b}'_{d-1}$.
Run a sieve in dimension $\beta_{\mathsf{sieve}}$ on the sublattice spanned by $\mathbf{b}'_0, \dots, \mathbf{b}'_{\beta_{\mathsf{sieve}}-1}$ to obtain a list $L$ of $N_{\mathsf{sieve}}(\beta_{\mathsf{sieve}})$ vectors.
return $L$
```

The parameter β_{bkz} controls the block size in the BKZ algorithm, with the cost scaling exponentially with it. The sieving algorithm outputs $N_{\mathsf{sieve}}(\beta_{\mathsf{sieve}})$ short vectors in the lattice and its complexity also scales exponentially with β_{sieve}. The magnitude $N_{\mathsf{sieve}}(\beta_{\mathsf{sieve}})$ also grows exponentially with β_{sieve} but slower than the cost of sieving. We will write $T_{\mathsf{BKZ}}(d, \beta_{\mathsf{bkz}})$ for the cost of running BKZ-β_{bkz} in dimension d and $T_{\mathsf{sieve}}(\beta_{\mathsf{sieve}})$ for the cost of sieving in dimension β_{sieve}. One possible instantiation of the lattice sieve algorithm is [BDGL16] which has a cost of $2^{0.292\,\beta_{\mathsf{sieve}}+o(\beta_{\mathsf{sieve}})}$. Thus, according to the best known algorithms we have $T_{\mathsf{BKZ}}(d, \beta_{\mathsf{bkz}}) \in \text{poly}\,(d) \cdot 2^{\Theta(\beta_{\mathsf{bkz}})}$ and $T_{\mathsf{sieve}}(\beta_{\mathsf{sieve}}) \in 2^{\Theta(\beta_{\mathsf{sieve}})}$. More specifically, we take these complexities from [MAT22, Lemma 4.1, Assumption 7.3].

Lemma 2.7 (Short Vectors Sampling Complexity). *Let $\mathbf{B}$ be a basis of a d-dimensional lattice. Then, the running time T_{sample} of Algorithm 2.1 for outputting $N_{\mathsf{sieve}}(\beta_{\mathsf{sieve}})$ short vectors is*

$$T_{\mathsf{sample}}\,(d, \beta_{\mathsf{bkz}}, \beta_{\mathsf{sieve}}) = T_{\mathsf{BKZ}}(d, \beta_{\mathsf{bkz}}) + T_{\mathsf{sieve}}(\beta_{\mathsf{sieve}}) \tag{2.12}$$

where

- ▷ $T_{\mathsf{BKZ}}(d, \beta_{\mathsf{bkz}}) = C_{\mathsf{prog}}^2 \cdot (d - \beta_{\mathsf{bkz}} + 1) \cdot T_{\mathsf{NNS}}({\beta_{\mathsf{bkz}}}^{\mathsf{eff}})$,
- ▷ $T_{\mathsf{sieve}}(\beta_{\mathsf{sieve}}) = C_{\mathsf{prog}} \cdot T_{\mathsf{NNS}}\,(\beta_{\mathsf{sieve}})$,
- ▷ $N_{\mathsf{sieve}}(\beta) = \left(\sqrt{\frac{4}{3}}\right)^{\beta}$ *is the expected number of sieve results,*
- ▷ $T_{\mathsf{NNS}}\,(\beta)$ *is the time complexity for finding all close pairs in dimension β (see [AGPS20a] with improvement of MATZOV [MAT22, Section 6]),*
- ▷ $C_{\mathsf{prog}} = 1/\left(1 - 2^{-0.292}\right)$ *is the number of close pairs search to run,*
- ▷ *and β^{eff} is the optimal sieve dimension to use for solving the Shortest Vector Problem (SVP) for lattices in dimension β.*

Note that in [Duc18], it is estimated that $\beta^{\mathsf{eff}} = \beta - \frac{\beta \log(4/3)}{\log(\beta/(2\pi e))}$.

The lengths of the short vectors produced by Algorithm 2.1. Assuming that the Gaussian Heuristic (GH) and the Geometric Series Assumption[6] (GSA) [Sch03] hold for a d-dimensional lattice, applying BKZ-β to it produces vectors $\mathbf{x}$ of average length [Che13]:

$$\|\mathbf{x}\| \approx \delta(\beta)^d \cdot \mathsf{Vol}\,(\Lambda)^{\frac{1}{d}}\,, \tag{2.13}$$

where $\delta(\beta) = \left(\frac{\beta}{2\pi e}\,(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}}$ is the root-Hermite factor[7]. With these assumptions, the expected length of the returned short vectors is given by:

[6] The GSA may lead us to underestimate the final complexity of a few bits (see [DP23b, Appendix A.3]).

[7] Experiments in [AD221] show that these assumptions hold for $d > \beta$ and $\beta \to \infty$ and in particular, it hold with good accuracy for $\beta > 50$.

Lemma 2.8 (Length of the sampled short vectors [MAT22, Lemma 4.2]). *Let Λ be a d-dimensional lattice. Then, Algorithm 2.1 outputs at least N vectors of expected length ℓ given by*

$$\ell \triangleq \mathsf{Vol}\left(\Lambda\right)^{1/d} \cdot \sqrt{\frac{4}{3}} \cdot \delta(\beta_{\mathsf{sieve}})^{\beta_{\mathsf{sieve}}-1} \cdot \delta(\beta_{\mathsf{bkz}})^{d-\beta_{\mathsf{sieve}}}. \tag{2.14}$$

In our attack, we select β_{bkz} and β_{sieve} such that $T_{\mathsf{BKZ}}(d, \beta_{\mathsf{bkz}}) = T_{\mathsf{sieve}}(\beta_{\mathsf{sieve}})$. Under the same GH and GSA assumptions, we derive the following lemma:

Lemma 2.9. *The short vectors produced by Algorithm 2.1 are in a sublattice Λ' of dimension β_{sieve} and expected volume*

$$\mathsf{Vol}\left(\Lambda'\right) = \left(\mathsf{Vol}\left(\Lambda\right)^{\frac{1}{d}} \cdot \delta(\beta_{\mathsf{bkz}})^{d-\beta_{\mathsf{sieve}}}\right)^{\beta_{\mathsf{sieve}}}. \tag{2.15}$$

Fast Fourier Transform. Dual lattice attacks widely use discrete Fourier transforms:

Definition 2.10 (Discrete Fourier Transform). *The discrete Fourier transform $\widehat{f}$ of a function $f \colon \mathbb{Z}_q^n \longrightarrow \mathbb{C}$ is defined as*

$$\widehat{f}\left(\mathbf{x}\right) = \sum_{\mathbf{a} \in \mathbb{Z}_q^n} f(\mathbf{a}) e^{-\frac{2i\pi}{q}\langle \mathbf{x}, \mathbf{a}\rangle}. \tag{2.16}$$

The effectiveness of our dual attacks is heavily dependent on the speed at which we can compute discrete Fourier transforms. In [MAT22, Ass. 7.4], it is estimated that a Fast Fourier Transform (FFT) over $\mathbb{Z}_q^n$ requires $n \cdot q^{n+1}$ multiplications. Note that this is clearly suboptimal for large prime q's: while this is not a problem in [MAT22] as q is reduced thanks to modulus switching, here, we drop modulus switching resulting in q being big and imposed by the original LWE instance (*i.e.* $q = 3329$ for Kyber). This motivates us to give the following finer estimation for the cost of an FFT over $\mathbb{Z}_{3329}^n$.

Proposition 2.11 (Complexity of the FFT). *Let $q = 3329$. There exists an FFT over $\mathbb{Z}_q$ with complexity given by:*

$$N_{\mathsf{FFT}}^{(\mathsf{add})} = 240500 \quad \textit{and} \quad N_{\mathsf{FFT}}^{(\mathsf{mul})} = 115928, \tag{2.17}$$

the number of additions and multiplications, respectively.

By using the algorithm in [DM90, §2.3.2] that reduces the calculation of an FFT over $\mathbb{Z}_q^n$ to FFT's over $\mathbb{Z}_q$, we deduce that the total cost to perform a discrete Fourier transform over $\mathbb{Z}_q^n$ is

$$n\, q^{n-1}\, N_{\mathsf{FFT}}^{(\mathsf{add})} \quad \textit{and} \quad n\, q^{n-1}\, N_{\mathsf{FFT}}^{(\mathsf{mul})} \tag{2.18}$$

additions and multiplications, respectively. Finally, by supposing as in [MAT22, Ass. 7.4] that the cost of an addition and a multiplication are

$$C_{\mathsf{add}} = 160 \quad \textit{and} \quad C_{\mathsf{mul}} = 1024, \tag{2.19}$$

respectively, the total cost of an FFT over $\mathbb{Z}_q^n$ is

$$C_{\mathsf{FFT}} = C_{\mathsf{add}}\, n\, q^{n-1}\, N_{\mathsf{FFT}}^{(\mathsf{add})} + C_{\mathsf{mul}}\, n\, q^{n-1}\, N_{\mathsf{FFT}}^{(\mathsf{mul})}. \tag{2.20}$$

We obtained the number of additions and multiplications required for an FFT over $\mathbb{Z}_{3329}$, namely $N_{\mathsf{FFT}}^{(\mathsf{add})}$ and $N_{\mathsf{FFT}}^{(\mathsf{mul})}$, by slightly modifying the FFTW software. This software basically allows one to enumerate a large number of FFT's and we simply selected the one that minimized the cost C_{FFT}[8].

[8] We give more details on how this result was obtained in https://github.com/kevin-carrier/CodedDualAttack/tree/main/claimFFT

3 Our Dual Attack Algorithm

In this section, we present our algorithm for solving the search LWE problem. Unlike MATZOV, we do not use modulus switching to reduce the modulus size. As noted in the introduction, we expect that a good lossy source code/quantizer results in a smaller additional noise term $\langle \mathbf{e}_{\mathsf{fft}}, \mathbf{s}_{\mathsf{fft}} \rangle$ compared to modulus switching. However, modulus switching may still offer benefits: it has certain advantages, such as improved FFT efficiency when the modulus is a power of two, which is not the case in the starting LWE problem for Kyber. Additionally, modulus switching provides more flexibility in choosing the parameter k_{fft}. Although a hybrid approach combining both techniques is possible, we do not pursue it in this paper due to the complexity it would add to the analysis.

3.1 Overview of the Algorithm

Our algorithm begins by partitioning the set of coordinates $I \stackrel{\triangle}{=} [\![1, n]\!]$ of the secret $\mathbf{s}$ into I_{enu}, I_{fft} and I_{lat}, with sizes n_{enu}, n_{fft} and n_{lat}, respectively. The three parts of the vector $\mathbf{s}$ are denoted as $\mathbf{s}_{\mathsf{enu}}$, $\mathbf{s}_{\mathsf{fft}}$ and $\mathbf{s}_{\mathsf{lat}}$ respectively. Similarly, we divide the columns of $\mathbf{A}$ to obtain $\mathbf{A}_{\mathsf{enu}}$, $\mathbf{A}_{\mathsf{fft}}$ and $\mathbf{A}_{\mathsf{lat}}$. As described in the introduction, our algorithm basically tries to guess a value $\widetilde{\mathbf{s}_{\mathsf{enu}}}$ for $\mathbf{s}_{\mathsf{enu}}$ by computing an associated score for $\widetilde{\mathbf{s}_{\mathsf{enu}}}$ namely the maximum value of the score function, $\max_{\mathbf{z} \in \mathbb{Z}_q^{k_{\mathsf{fft}}}} F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}(\mathbf{z})$ and making a decision on this value. Apart from the lossy source code approach described above, we depart from the MATZOV algorithm in the two following ways: firstly, we will make the bet that $\mathbf{s}_{\mathsf{enu}} = \mathbf{0}$, that is, we only consider $\widetilde{\mathbf{s}_{\mathsf{enu}}} = \mathbf{0}$ whereas MATZOV enumerates and tests several values $\widetilde{\mathbf{s}_{\mathsf{enu}}}$ taken by decreasing likelihood. Secondly, in our algorithm, we will select I_{lat} once and for all (instead of changing it at each iteration), and then iterate R times with different choices for I_{enu} and I_{fft} on the remaining positions, this allows us to reuse the computed dual vectors for each iteration. The skeleton of the whole algorithm is given in Algorithm 3.1.

Algorithm 3.1 The code based dual attack to solve LWE

Input: a sample $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ produced by an $\mathsf{LWE}\,(q, n, m, \mathcal{B}_\alpha, \mathcal{B}_\alpha)$ oracle.
Parameters: some positive integers R, T, β_{bkz}, β_{sieve}, n_{enu}, n_{fft}, k_{fft}, n_{lat}, d_{lsc} and an $[n_{\mathsf{fft}}, k_{\mathsf{fft}}]_q$ linear code with generator matrix $\mathbf{G}$.
Output: the secret vector $\mathbf{s}$.

```
choose $I_{\mathsf{lat}} \subseteq [\![1, n]\!]$ such that $|I_{\mathsf{lat}}| = n_{\mathsf{lat}}$;
$\mathscr{S} \leftarrow$ SET_OF_SHORT_LATTICE_VECTORS$(\mathbf{A}, I_{\mathsf{lat}})$;
repeat $R$ times
    choose a partition $I_{\mathsf{enu}} \cup I_{\mathsf{fft}}$ of $[\![1, n]\!] \setminus I_{\mathsf{lat}}$ with $|I_{\mathsf{enu}}| = n_{\mathsf{enu}}$, $|I_{\mathsf{fft}}| = n_{\mathsf{fft}}$;
    $\mathbf{A}_{\mathsf{enu}}, \mathbf{A}_{\mathsf{fft}} \leftarrow$ select the columns of $\mathbf{A}$ indexed by $I_{\mathsf{enu}}$ and $I_{\mathsf{fft}}$ respectively;
    $\mathscr{L} \leftarrow$ LWE_SAMPLES$(\mathscr{S}, \mathbf{A}_{\mathsf{fft}}, \mathbf{G})$;
    $\mathrm{V} \leftarrow$ SOLVE_LWE_WITH_FFT$(\mathscr{L})$;
    if $\mathrm{V} \geq T$ then
        $\mathbf{s}_{\mathsf{enu}} \leftarrow \mathbf{0}$
        $(\mathbf{s}_{\mathsf{fft}}, \mathbf{s}_{\mathsf{lat}}) \leftarrow$ SUB_LWE_SOLVER$(\begin{bmatrix}\mathbf{A}_{\mathsf{fft}} & \mathbf{A}_{\mathsf{lat}}\end{bmatrix}, \mathbf{b})$
        return $(\mathbf{s}_{\mathsf{enu}}, \mathbf{s}_{\mathsf{fft}}, \mathbf{s}_{\mathsf{lat}}, I_{\mathsf{enu}}, I_{\mathsf{fft}}, I_{\mathsf{lat}})$
end repeat
```

We now give details on the procedures Set_of_short_lattice_vectors, LWE_Samples, Solve_LWE_with_FFT and sub_LWE_solver.

The Lattice Reduction Function Set_of_short_lattice_vectors. This function outputs a set $\mathscr{S}$ of pairs of short vectors $(\mathbf{x}, \mathbf{y})$ obtained by performing the short vectors sampler Algorithm 2.1 with the appropriate choice of parameters on the lattice $\Lambda(\mathbf{B})$ where the lattice basis $\mathbf{B} \in \mathbb{R}^{(m+n_{\mathsf{lat}})\times(m+n_{\mathsf{lat}})}$ is given by Construction A:

$$\mathbf{B} \stackrel{\triangle}{=} \left[\begin{array}{c|c} \mathbf{Id}_m & \mathbf{0} \\ \hline {\mathbf{A}_{\mathsf{lat}}}^{\top} & q\mathbf{Id}_{n_{\mathsf{lat}}} \end{array}\right]. \tag{3.1}$$

This gives pairs $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}^m \times \mathbb{Z}^{n_{\mathsf{lat}}}$ for which both $\mathbf{x}$ and $\mathbf{y}$ are short and satisfy $\mathbf{y} = \mathbf{A}_{\mathsf{lat}}^{\top}\mathbf{x} \mod q$. For the rest of this paper we denote by N the expected size of $\mathscr{S}$. From Lemma 2.7, we have

Notation 3.1.

$$N \stackrel{\triangle}{=} N_{\mathsf{sieve}}\left(\beta_{\mathsf{sieve}}\right) \stackrel{\triangle}{=} \left(\sqrt{\frac{4}{3}}\right)^{\beta_{\mathsf{sieve}}}. \tag{3.2}$$

Producing New LWE Samples with LWE_Samples. Each pair $(\mathbf{x}, \mathbf{y})$ of short vectors in $\mathscr{S}$ yields one LWE sample by decoding $\mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x}$ in the code $\mathcal{C}_{\mathsf{lsc}}$ generated by $\mathbf{G}$. This yields an output $\mathbf{u}_{\mathsf{lsc}} \in \mathbb{Z}_q^{k_{\mathsf{fft}}}$ such that $\mathbf{e}_{\mathsf{lsc}} \stackrel{\triangle}{=} \mathbf{A}_{\mathsf{fft}}^{\top}\mathbf{x} - \mathbf{G}\mathbf{u}_{\mathsf{lsc}}$ is of small norm, say close to some d_{lsc}.The new LWE sample is given by the pair $(\mathbf{u}_{\mathsf{lsc}}, \langle \mathbf{x}, \mathbf{b}\rangle)$. It is readily seen that, when $\mathbf{s}_{\mathsf{enu}} = \mathbf{0}$ we have

$$\langle \mathbf{x}, \mathbf{b}\rangle = \left\langle \mathbf{u}_{\mathsf{lsc}}, \mathbf{G}^{\top}\mathbf{s}_{\mathsf{fft}}\right\rangle + \langle \mathbf{x}, \mathbf{e}\rangle + \langle \mathbf{y}, \mathbf{s}_{\mathsf{lat}}\rangle + \langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}}\rangle \tag{3.3}$$

which corresponds to an LWE sample with secret $\mathbf{G}^{\top}\mathbf{s}_{\mathsf{fft}}$ and noise $e' = \langle \mathbf{x}, \mathbf{e}\rangle + \langle \mathbf{y}, \mathbf{s}_{\mathsf{lat}}\rangle + \langle \mathbf{e}_{\mathsf{lsc}}, \mathbf{s}_{\mathsf{fft}}\rangle$. All these samples $(\mathbf{u}_{\mathsf{lsc}}, \langle \mathbf{x}, \mathbf{b}\rangle)$ are then put in a list $\mathscr{L}$ which is the output of LWE_Samples.

Solving the Induced LWE Instance with Solve_LWE_with_FFT. This procedure outputs a real number V which indicates to us how noisy the aforementioned LWE samples are. This is done by searching exhaustively for the solution $\mathbf{G}^{\top}\mathbf{s}_{\mathsf{fft}}$ by computing the score function for all $\mathbf{z} \in \mathbb{Z}_q^{k_{\mathsf{fft}}}$:

$$F_{\mathbf{0}}^{(\mathsf{lsc})}(\mathbf{z}) \stackrel{\triangle}{=} \sum_{(\mathbf{u}_{\mathsf{lsc}}, b)\in\mathscr{L}} \cos\left(\tfrac{2\pi}{q}\left(b - \langle \mathbf{u}_{\mathsf{lsc}}, \mathbf{z}\rangle\right)\right), \tag{3.4}$$

and returning its maximum value, $V = \max_{\mathbf{z}\in\mathbb{Z}_q^{k_{\mathsf{fft}}}} F_{\mathbf{0}}^{(\mathsf{lsc})}(\mathbf{z})$. If $\mathbf{s}_{\mathsf{enu}} = \mathbf{0}$ we expect that this maximum value is achieved for $\mathbf{z} = \mathbf{G}^{\top}\mathbf{s}_{\mathsf{fft}}$. The score function is efficiently computed with an FFT as follows. First we compute a function $f_{\mathbf{0}}^{(\mathsf{lsc})}$ defined for $\mathbf{a} \in \mathbb{Z}_q^{k_{\mathsf{fft}}}$ as

$$f_{\mathbf{0}}^{(\mathsf{lsc})}(\mathbf{a}) \stackrel{\triangle}{=} \sum_{(\mathbf{a}, b)\,:\,(\mathbf{a}, b)\in\mathscr{L}} e^{\frac{2i\pi}{q}\, b}, \tag{3.5}$$

then we compute the FFT of $f_{\mathbf{0}}^{(\mathsf{lsc})}$ and take its real part. It is readily seen that

$$F_{\mathbf{0}}^{(\mathsf{lsc})} = \mathfrak{Re}\left(\widehat{f_{\mathbf{0}}^{(\mathsf{lsc})}}\right). \tag{3.6}$$

Recovering the Rest of the Secret with sub_LWE_solver. At this point, we expect that, under the condition that our parameters are well-chosen, if $V \geq T$ (here T will be chosen around the expected value of $F_{\mathbf{0}}^{(\mathsf{lsc})}(\mathbf{G}^{\top}\mathbf{s}_{\mathsf{fft}})$ when $\mathbf{s}_{\mathsf{enu}} = \mathbf{0}$) then with overwhelming probability, $\mathbf{s}_{\mathsf{enu}} = \mathbf{0}$. In other words, we recovered n_{enu} positions of the secret from the original LWE problem $(\mathbf{A}, \mathbf{b})$ of dimension $m \times n$. Note that when $\mathbf{s}_{\mathsf{enu}} = \mathbf{0}$ we can write

$$\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} = \left[\mathbf{A}_{\mathsf{fft}}\ \mathbf{A}_{\mathsf{lat}}\right]\begin{pmatrix}\mathbf{s}_{\mathsf{fft}} \\ \mathbf{s}_{\mathsf{lat}}\end{pmatrix} + \mathbf{e}. \tag{3.7}$$

As such, if indeed $\mathbf{s}_{\mathsf{enu}} = \mathbf{0}$, we can recover $\mathbf{s}_{\mathsf{fft}}$ and $\mathbf{s}_{\mathsf{lat}}$ by solving the new LWE problem given by $\left(\left[\mathbf{A}_{\mathsf{fft}}\ \mathbf{A}_{\mathsf{lat}}\right], \mathbf{b}\right)$, which has strictly smaller dimension $m \times (n - n_{\mathsf{enu}})$. We don't specify a particular algorithm for the sub_LWE_solver routine, since it will be called at most once, and solving an LWE instance with smaller dimension and with error distribution $\mathcal{B}_{\alpha}$ has complexity negligible compared to the other computations in Algorithm 3.1.

3.2 Correctness of the Algorithm

The following lemma provides the conditions on the parameters for Algorithm 3.1 to succeed.

Lemma 3.2 (Correctness). *Suppose* $(\mathbf{A}, \mathbf{b})$ *is sampled from an* $\mathsf{LWE}(q, n, m, \mathcal{B}_\alpha, \mathcal{B}_\alpha)$ *oracle, and* SUB_LWE_SOLVER *returns* $\mathbf{s}_{\mathsf{fft}}$ *and* $\mathbf{s}_{\mathsf{lat}}$ *with probability* $1-\mu$ *when the bet* $\mathbf{s}_{\mathsf{enu}} = \mathbf{0}$ *is valid and the secret meets the threshold, namely* $F_{\mathbf{0}}^{(\mathsf{lsc})}(\mathbf{G}^\top \mathbf{s}_{\mathsf{fft}}) \geq T$*. The probability that our algorithm succeeds in recovering the secret* $\mathbf{s}$*, is lower bounded by*

$$P_{\text{success}} \triangleq \eta \cdot P_{\mathsf{good}} \cdot (1-\mu) - \varepsilon \tag{3.8}$$

where

$$\varepsilon \triangleq R \cdot q^{k_{\mathsf{fft}}} \cdot P_{\mathsf{wrong}}, \tag{3.9}$$

$$\eta \triangleq \left(1 - \sum_{t=0}^{n_{\mathsf{enu}}+n_{\mathsf{fft}}} \left(1 - \frac{\binom{t}{n_{\mathsf{enu}}}}{\binom{n_{\mathsf{enu}}+n_{\mathsf{fft}}}{n_{\mathsf{enu}}}}\right)^R \binom{n_{\mathsf{enu}}+n_{\mathsf{fft}}}{t} p_0^t (1-p_0)^{n_{\mathsf{enu}}+n_{\mathsf{fft}}-t}\right), \tag{3.10}$$

$$P_{\mathsf{good}} \triangleq \mathbb{P}\left(F_{\mathbf{0}}^{(\mathsf{lsc})}\left(\mathbf{G}^\top \mathbf{s}_{\mathsf{fft}}\right) \geq T \middle| \mathbf{s}_{\mathsf{enu}} = \mathbf{0}\right), \tag{3.11}$$

$$P_{\mathsf{wrong}} \triangleq \mathbb{P}\left(F_{\mathbf{0}}^{(\mathsf{lsc})}\left(\mathbf{z}\right) \geq T \middle| \mathbf{s}_{\mathsf{enu}} \neq \mathbf{0}\right), \tag{3.12}$$

$$p_0 \triangleq \mathbb{P}\left(\mathcal{B}_\alpha = 0\right) = 2^{-2\alpha}\binom{2\alpha}{\alpha} \tag{3.13}$$

and where $\mathbf{z}$ *is taken uniformly at random in* $\mathbb{Z}_q^{k_{\mathsf{fft}}}$*. We will use Approximations 4.8 and 4.9 to estimate* P_{good} *and* P_{wrong}*, respectively.*

Proof. For $i \in [\![1, R]\!]$, let us denote the following events for each iteration i:

- A_i: "$s_{\mathsf{enu}} = \mathbf{0}$";
- B_i: "$F_{\mathbf{0}}^{(\mathsf{lsc})}(\mathbf{G}^\top \mathbf{s}_{\mathsf{fft}}) \geq T$";
- C_i: "sub-LWE finds the secret";
- D_i: "$\exists \mathbf{z} \in \mathbb{Z}_q^{k_{\mathsf{fft}}},\ F_{\mathbf{0}}^{(\mathsf{lsc})}(\mathbf{z}) \geq T$";

Using the union bound on the probability that the algorithm fails and taking the complement of this event yields that the probability that our algorithm succeeds is lower bounded by

$$\begin{aligned}
&\mathbb{P}\left(\left(\bigcup_i (A_i \cap B_i \cap C_i)\right) \cap \left(\bigcap_i (A_i \cup \overline{D_i})\right)\right) \\
&\qquad \geq \mathbb{P}\left(\bigcup_i (A_i \cap B_i \cap C_i)\right) - \mathbb{P}\left(\bigcup_i (\overline{A_i} \cap D_i)\right)
\end{aligned}$$

Next, applying the union bound again and considering that we run R iterations, we can upper bound $\mathbb{P}\left(\bigcup_i \left(\overline{A_i} \cap D_i\right)\right)$ by $Rq^{k_{\mathsf{fft}}} P_{\mathsf{wrong}}$, which gives the $-\varepsilon$ term in the inequality stated in the lemma. On the other hand, it is straightforward to lower bound the first term:

$$\begin{aligned}
\mathbb{P}\left(\bigcup_i (A_i \cap B_i \cap C_i)\right) &= \mathbb{P}\left(\bigcup_i (A_i \cap B_i \cap C_i) \mid \bigcup_i A_i\right) \cdot \mathbb{P}\left(\bigcup_i A_i\right) \\
&\geq \mathbb{P}\left(B_{i_0} \cap C_{i_0} \mid A_{i_0}\right) \cdot \mathbb{P}\left(\bigcup_i A_i\right) \\
&= \mathbb{P}\left(C_{i_0} \mid A_{i_0}, B_{i_0}\right) \cdot \mathbb{P}\left(B_{i_0} \mid A_{i_0}\right) \cdot \mathbb{P}\left(\bigcup_i A_i\right) \\
&= (1-\mu) \cdot P_{\mathsf{good}} \cdot \mathbb{P}\left(\bigcup_i A_i\right)
\end{aligned}$$

Let n_0 denote the number of zeros in $\mathbf{s}_{[\![1,n]\!]\setminus I_{\mathsf{lat}}}$. Then, we have

$$\begin{aligned}\mathbb{P}\left(\bigcup_i A_i\right) &= 1-\mathbb{P}\left(\bigcap_i \overline{A_i}\right)\\ &= 1-\sum_{t=0}^{n_{\mathsf{enu}}+n_{\mathsf{fft}}} \mathbb{P}\left(\bigcap_i \overline{A_i} \mid n_0=t\right)\cdot \mathbb{P}\left(n_0=t\right)\\ &= 1-\sum_{t=0}^{n_{\mathsf{enu}}+n_{\mathsf{fft}}} \left(1-\frac{\binom{t}{n_{\mathsf{enu}}}}{\binom{n_{\mathsf{enu}}+n_{\mathsf{fft}}}{n_{\mathsf{enu}}}}\right)^R \cdot \mathbb{P}\left(n_0=t\right)\end{aligned}$$

Clearly, n_0 follows a binomial distribution with parameter $n_{\mathsf{enu}}+n_{\mathsf{fft}}$ and p_0. This concludes the proof. □

Note that Lemma 3.2 does not impose any constraints on N and T. However, we must choose them carefully. First, if $R\cdot q^{k_{\mathsf{fft}}}\cdot P_{\mathsf{wrong}}\geq 1$ or if P_{good} is too small, then the lower bound we obtain for the probability of success of our dual attack is 0, which means we cannot actually guarantee a success. This is why, in Section 5, we select N and T that ensure a high probability of success. In particular, we set $\eta \geq 0.62$ and ε close to 0 (see Appendix C.2), ensuring a success probability of at least $0.3(1-\mu)$. Furthermore, it is worth noting that the condition "$R\cdot q^{k_{\mathsf{fft}}}\cdot P_{\mathsf{wrong}}$ is much smaller than 1" resolves the indistinguishability issue raised in [DP23b].

3.3 Which Codes Should We Use?

In Algorithm 3.1, one could ask: which code should we use for $\mathcal{C}_{\mathsf{lsc}}$? In terms of the decoding distance alone, the answer would be, just use a random code of dimension k_{fft} in $\mathbb{Z}_q^{n_{\mathsf{fft}}}$. In this case, we would obtain the decoding distance $d\approx q^{1-\frac{k_{\mathsf{fft}}}{n_{\mathsf{fft}}}}\sqrt{\frac{n_{\mathsf{fft}}}{2\pi e}}$ (see Equation (2.11)) attaining the bound (2.9) or (2.10). However, the decoding algorithm we could use in this case would be too complex for our purpose. We could instead use a product code structure as in [BDGL16]. Contrarily to what happens in the latter case, where spherical codes can be used, we are in a situation where more structured codes could do the job better. A natural answer is given here by polar codes.

The generator matrix of such a code in length n which is a power of 2 and an arbitrary code dimension k is obtained as follows. We define a generator matrix $\mathbf{G}$ for our code as $\mathbf{G}\triangleq \mathbf{K}_1\otimes\cdots\otimes\mathbf{K}_{\log_2(n)}\cdot\mathbf{F}$ where $\otimes$ stands for the Kronecker product, $\mathbf{K}_i\triangleq\begin{bmatrix}1&1\\\alpha_i&0\end{bmatrix}$ and α_i's are some invertible elements chosen uniformly at random in $\mathbb{Z}_q^*$. The matrix $\mathbf{F}\in\mathbb{Z}_q^{n\times k}$ is an expansion matrix such that for all $\mathbf{m}\in\mathbb{Z}_q^k$, $\mathbf{Fm}$ is exactly $\mathbf{m}$ on k positions and 0 on the others (that are the *frozen* positions). Since the code length we need is not necessarily a power of 2, we adjust it by puncturing the code (*i.e.* we remove as many rows of the generator matrix as needed).

Furthermore, the Successive Cancellation (SC) decoder which is classically used to decode polar codes in the error correction scenario can be turned with a simple modification into an algorithm for finding a close codeword (but not necessarily the closest one) [KU10]. This is precisely what is needed in our context. It needs a noise model to instantiate it, and this can be done for our Euclidean metric by using the Gaussian noise model. To get even closer to the optimal decoding distance, we have improved the decoding process: firstly we turn the SC algorithm into a probabilistic decoder, then we call it several times to get a list of codeword candidates and choose the closest one from this list (see Appendix A again to get more details about our list decoder). This procedure induces a new small constant factor L in the whole complexity which is the size of the decoding list; but in return this additional cost allows us to achieve a better decoding distance. The complexity for decoding is given by:

Lemma 3.3 (Decoding polar codes). *List decoding an $[n_{\mathsf{fft}},k_{\mathsf{fft}}]_q$ polar code by using L probabilistic SC decoders can be done with time complexity of order*

$$T_{\mathsf{dec}}\left(q,n_{\mathsf{fft}},k_{\mathsf{fft}}\right)=3\cdot L\cdot\left(C_{\mathrm{mul}}\cdot N_{FFT}^{(\mathrm{mul})}(q)+C_{\mathrm{add}}\cdot N_{FFT}^{(\mathrm{add})}(q)\right)\cdot n'_{\mathsf{fft}}\log_2(n'_{\mathsf{fft}}) \tag{3.14}$$

where n'_{fft} is the smallest power of 2 greater than n_{fft} and $N^{(\mathrm{mul})}_{\mathsf{FFT}}(q)$ is the number of multiplications we need to achieve a discrete Fourier transform over $\mathbb{Z}_q$.

The proof of this lemma directly follows from Lemma A.4.

By using similar arguments as in [KU10], it can be proven that for n_{fft} tending to infinity and constant $\frac{k_{\mathsf{fft}}}{n_{\mathsf{fft}}}$, the average distance achieved by our decoder is

$$d_{\mathsf{lsc}} = \sqrt{\frac{n_{\mathsf{fft}}}{2\pi e}} \cdot q^{1-\frac{k_{\mathsf{fft}}}{n_{\mathsf{fft}}}} \cdot (1 + o(1)) \tag{3.15}$$

This result is essentially due to the polarization phenomenon (see Appendix A). However, Equation (3.15) is not precise enough to accurately estimate the full complexity of our dual attack due to the $o(1)$ term. For this reason, we provide a C implementation[9] and present experimental results demonstrating that polar codes are perfectly suited to our case. The experiments we conducted use the exact codes required for our dual attacks, and our optimization suggests choosing $L = 1$. To justify the use of polar codes in this context, we verified that the total complexity of our dual attack closely matches the ideal scenario, where decoding at the distance $d_{\mathsf{lsc}} \triangleq \sqrt{\frac{n_{\mathsf{fft}}}{2\pi e}} \cdot q^{1-\frac{k_{\mathsf{fft}}}{n_{\mathsf{fft}}}}$ would incur the same cost as using polar codes.

4 Analysis of our Dual Attack

4.1 Complexity of our Dual Attack

A general formula for the complexity of our algorithm using polar codes for $\mathcal{C}_{\mathsf{lsc}}$ is given by the following theorem:

Theorem 4.1 (Complexity of Algorithm 3.1). *Using the same notations as in the correctness Lemma 3.2 and by supposing that the cost of one call to* Sub_LWE_Solver *is negligible, the average time complexity of Algorithm 3.1 is upper bounded by*

$$T_{\mathsf{sample}} + R \cdot (N \cdot T_{\mathsf{dec}} + T_{\mathsf{FFT}}) \tag{4.1}$$

where

- ▷ $T_{\mathsf{sample}} \triangleq T_{\mathsf{sample}}(m + n_{\mathsf{lat}}, \beta_{\mathsf{bkz}}, \beta_{\mathsf{sieve}})$ *is the cost to produce* $N = N_{\mathsf{sieve}}(\beta_{\mathsf{sieve}})$ *short vectors in* $\Lambda(\mathbf{B})$*. This cost is given by Lemma 2.7;*
- ▷ $T_{\mathsf{dec}} \triangleq T_{\mathsf{dec}}(q, n_{\mathsf{fft}}, k_{\mathsf{fft}})$ *is the cost for decoding a random vector in* $\mathbb{Z}_q^{n_{\mathsf{fft}}}$ *in* $\mathcal{C}_{\mathsf{lsc}}$ *generated by* $\mathbf{G}$*. This cost is given by Lemma 3.3;*
- ▷ T_{FFT} *is the cost of an* FFT *over* $\mathbb{Z}_q^{k_{\mathsf{fft}}}$ *and is given by Proposition 2.11;*

4.2 Computing an Accurate Approximation of P_{good} and P_{wrong}

In this subsection, we provide an accurate estimation of the probabilities P_{good} and P_{wrong} that appear in Theorem 4.1. First, let's recall the expression of the score function:

$$F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^\top \widetilde{\mathbf{s}_{\mathsf{fft}}}\right) \triangleq \sum_{(\mathbf{x},\mathbf{y}) \in \mathscr{S}} \cos\left(\frac{2\pi}{q}\left(\left\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\widetilde{\mathbf{s}_{\mathsf{enu}}}\right\rangle - \left\langle \mathbf{u}_{\mathsf{lsc}}, \mathbf{G}^\top \widetilde{\mathbf{s}_{\mathsf{fft}}}\right\rangle\right)\right) \tag{4.2}$$

$$= \sum_{(\mathbf{x},\mathbf{y}) \in \mathscr{S}} \cos\left(\frac{2\pi}{q}\left(\left\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\mathsf{enu}}\widetilde{\mathbf{s}_{\mathsf{enu}}}\right\rangle - \left\langle \mathbf{c}_{\mathsf{lsc}}, \widetilde{\mathbf{s}_{\mathsf{fft}}}\right\rangle\right)\right) \tag{4.3}$$

where $\mathscr{S}$ is a set of N short vectors drawn from $\{(\mathbf{x},\mathbf{y}) \in \Lambda(\mathbf{B}) \;:\; \|(\mathbf{x},\mathbf{y})\| \leq d_{\mathsf{lat}}\}$. Here, $\mathbf{c}_{\mathsf{lsc}} \triangleq \mathbf{G}\mathbf{u}_{\mathsf{lsc}}$ is a codeword in $\mathcal{C}_{\mathsf{lsc}} \subseteq \mathbb{Z}_q^{n_{\mathsf{fft}}}$, obtained by decoding $\mathbf{A}_{\mathsf{fft}}^\top \mathbf{x}$ using our polar code decoder. It is important to note that, according to Lemma 2.9, the short lattice vectors in $\mathscr{S}$ generated by Algorithm 2.1 are not uniformly distributed within $\Lambda(\mathbf{B}) \cap \mathsf{Ball}^{m+n_{\mathsf{lat}}}_{d_{\mathsf{lat}}}$; instead, they belong to a β_{sieve}-dimensional sublattice $\Lambda(\mathbf{B}') \subseteq \Lambda(\mathbf{B})$, where $\mathbf{B}' \in \mathbb{R}^{(m+n_{\mathsf{lat}}) \times \beta_{\mathsf{sieve}}}$. More precisely, we make the following assumption regarding the distribution of vectors in $\mathscr{S}$:

[9] https://github.com/kevin-carrier/CodedDualAttack/tree/main/PolarCodeOverZq

Assumption 4.2. *We assume that the set $\mathscr{S}$ consists of N vectors uniformly sampled from*

$$\{\mathbf{w} \in \Lambda(\mathbf{B}') \ : \ \|\mathbf{w}\| \leq d_{\mathsf{lat}}\} \, ,$$

where $\Lambda(\mathbf{B}')$ is a β_{sieve}-dimensional sublattice of $\Lambda(\mathbf{B})$ with basis $\mathbf{B}' \in \mathbb{R}^{(m+n_{\mathsf{lat}})\times\beta_{\mathsf{sieve}}}$, and volume as stated in Lemma 2.9. The radius $d_{\mathsf{lat}} \triangleq \frac{\ell(\beta_{\mathsf{sieve}}+1)}{\beta_{\mathsf{sieve}}}$ corresponds to that of a β_{sieve}-dimensional Euclidean ball, within which the average vector length ℓ is given in Lemma 2.8.

Additionally, we assume that the probability distribution of the output of our polar code decoder has radial symmetry. This means that the probability depends only on the distance between the returned codeword and the word to be decoded, not on the specific direction. Through experimentation, we observed that this distance closely follows a normal distribution. Based on these observations, we make the following assumption:

Assumption 4.3. *Let $\mathbf{u}$ be any vector from $\mathbb{Z}_q^{n_{\mathsf{fft}}}$, and let $\mathrm{Dec}(\mathbf{u})$ represent the random variable corresponding to the output of our polar code decoding algorithm. The distribution of $\|\mathbf{u} - \mathrm{Dec}(\mathbf{u})\|$ does not depend on $\mathbf{u}$ and we assume that it can be smoothed and approximated by the normal distribution $\mathcal{N}(\mu_{\mathsf{lsc}}, \sigma_{\mathsf{lsc}}^2)$, where μ_{lsc} is the mean and σ_{lsc} is the standard deviation, both determined through simulations. We also assume that the conditional distribution of $\mathbf{u} - \mathrm{Dec}(\mathbf{u})$ given that $\|\mathbf{u} - \mathrm{Dec}(\mathbf{u})\| = d_{\mathsf{lsc}}$ is uniform over $(\Lambda(\mathcal{C}_{\mathsf{lsc}}) + \mathbf{u}) \cap \mathsf{Sphere}_{d_{\mathsf{lsc}}}^{n_{\mathsf{fft}}}$.*

First-Level Approximation of the Score Function. The score function $F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}^{(\mathsf{lsc})}\left(\mathbf{G}^\top\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)$, as recalled in Equation (4.3), is a random variable influenced by multiple sources of randomness:

(i) the randomness in the short vector sampling Algorithm 2.1 that generates $\mathscr{S}$,
(ii) the inherent randomness in the polar code decoder Dec,
(iii) the randomness in the LWE instance, particularly in the choice of the matrix $\mathbf{A}$,
(iv) the randomness of the guess $\widetilde{\mathbf{s}_{\mathsf{enu}}}$, which primarily arises from the selection of I_{enu},
(v) the randomness of the guess $\widetilde{\mathbf{s}_{\mathsf{fft}}}$.

We denote by $\mathbb{E}_{\mathscr{S},\mathrm{Dec}}\left(\cdot\right)$ the conditional expectation over the randomness sources (i) and (ii). Thus, a first-level approximation of the score function is:

Approximation 4.4 (First-Level Approximation). *The score function $F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}^{(\mathsf{lsc})}\left(\mathbf{G}^\top\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)$ can be approximated by its conditional expectation*

$$\mathbb{E}_{\mathscr{S},\mathrm{Dec}}\left(F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}^{(\mathsf{lsc})}\left(\mathbf{G}^\top\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)\right) \tag{4.4}$$

where the expectation is taken over the randomness of the polar code decoder and the short vector sampler.

Lemma 4.5. *The conditional expectation in Approximation 4.4 can be expressed as*

$$\mathbb{E}_{\mathscr{S},\mathrm{Dec}}\left(F_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}^{(\mathsf{lsc})}\left(\mathbf{G}^\top\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)\right) = \sum_{\substack{\left(\mathbf{w}_{\mathsf{lat}}^\vee, \mathbf{w}_{\mathsf{lsc}}^\vee\right) \\ \in \Lambda(\mathbf{B}_{\mathsf{global}}^{\mathsf{tmp}})^\vee + \frac{\left(\mathbf{B}'^\top\mathbf{r}_{\mathsf{lat}}, \mathbf{B}_{\mathsf{lsc}}^\top\mathbf{r}_{\mathsf{lsc}}\right)}{q}}} \widehat{f_{\mathsf{lat}}}\left(\mathbf{w}_{\mathsf{lat}}^\vee\right) \cdot \widehat{f_{\mathsf{lsc}}}\left(\mathbf{w}_{\mathsf{lsc}}^\vee\right) , \tag{4.5}$$

where

$$f_{\mathsf{lat}}\left(\mathbf{w}_{\mathsf{lat}}\right) \triangleq \mathbb{P}\left(\mathbf{B}'\mathbf{w}_{\mathsf{lat}} \in \mathscr{S}\right) , \quad f_{\mathsf{lsc}}\left(\mathbf{w}_{\mathsf{lsc}}\right) \triangleq \mathbb{P}\left(\mathbf{u} - \mathrm{Dec}\left(\mathbf{u}\right) = \mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\right) , \tag{4.6}$$

$$\left(\mathbf{r}_{\mathsf{lat}}, \mathbf{r}_{\mathsf{lsc}}\right) \triangleq \mathbf{r} \triangleq \left(\mathbf{e} + \mathbf{A}_{\mathsf{enu}}(\mathbf{s}_{\mathsf{enu}} - \widetilde{\mathbf{s}_{\mathsf{enu}}}) + \mathbf{A}_{\mathsf{fft}}(\mathbf{s}_{\mathsf{fft}} - \widetilde{\mathbf{s}_{\mathsf{fft}}}), \mathbf{s}_{\mathsf{lat}}, \widetilde{\mathbf{s}_{\mathsf{fft}}}\right) , \tag{4.7}$$

and

$$\mathbf{B}_{\mathsf{global}}^{\mathsf{tmp}} \triangleq \begin{bmatrix} \mathbf{Id}_{\beta_{\mathsf{sieve}}} & \mathbf{0} \\ \mathbf{A}_{\mathsf{fft}}^\top\mathbf{B}'_{[m]} & \mathbf{Id}_{n_{\mathsf{fft}}} \end{bmatrix} . \tag{4.8}$$

Here, $\mathbf{r}_{\mathsf{lat}}$ corresponds to the first $m + n_{\mathsf{lat}}$ coordinates of $\mathbf{r}$, while $\mathbf{r}_{\mathsf{lsc}}$ corresponds to its last n_{fft} coordinates. Additionally, under Assumption 4.3, we have that the random variable $\mathbf{u} - \mathrm{Dec}(\mathbf{u})$ is independent of $\mathbf{u}$ and follows the distribution induced by the polar code decoder.

Proof. For any vector in $\varLambda(\mathbf{B}')$, let $\mathbf{x}$ and $\mathbf{y}$ denote its first m coordinates and its next n_{lat} coordinates, respectively. Taking the conditional expectation over the randomness of both the polar code decoder and the short vector sampler, we obtain:

$$\begin{aligned}\mathbb{E}_{\mathscr{S},\mathrm{Dec}}\left(F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^\top\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)\right) &= \sum_{\substack{(\mathbf{x},\mathbf{y})\in\varLambda(\mathbf{B}')\\ \mathbf{c}_{\mathsf{lsc}}\in\varLambda(\mathcal{C}_{\mathsf{lsc}})}} \mathbb{P}\left((\mathbf{x},\mathbf{y})\in\mathscr{S} \text{ and } \mathrm{Dec}\left(\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}\right)=\mathbf{c}_{\mathsf{lsc}}\right)\\ &\qquad\cdot\cos\left(\tfrac{2\pi}{q}\left(\left\langle\mathbf{x},\mathbf{b}-\mathbf{A}_{\mathsf{enu}}\widetilde{\mathbf{s}_{\mathsf{enu}}}\right\rangle-\left\langle\mathbf{c}_{\mathsf{lsc}},\widetilde{\mathbf{s}_{\mathsf{fft}}}\right\rangle\right)\right)\\ &= \sum_{\substack{(\mathbf{x},\mathbf{y})\in\varLambda(\mathbf{B}')\\ \mathbf{w}_{\mathsf{lsc}}\in\varLambda(\mathcal{C}_{\mathsf{lsc}})+\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}}} \mathbb{P}\left((\mathbf{x},\mathbf{y})\in\mathscr{S} \text{ and } \mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}-\mathrm{Dec}\left(\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}\right)=\mathbf{w}_{\mathsf{lsc}}\right)\\ &\qquad\cdot\cos\left(\tfrac{2\pi}{q}\left(\left\langle\mathbf{x},\mathbf{b}-\mathbf{A}_{\mathsf{enu}}\widetilde{\mathbf{s}_{\mathsf{enu}}}\right\rangle-\left\langle\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}-\mathbf{w}_{\mathsf{lsc}},\widetilde{\mathbf{s}_{\mathsf{fft}}}\right\rangle\right)\right)\\ &= \sum_{\substack{(\mathbf{x},\mathbf{y})\in\varLambda(\mathbf{B}')\\ \mathbf{w}_{\mathsf{lsc}}\in\varLambda(\mathcal{C}_{\mathsf{lsc}})+\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}}} \mathbb{P}\left((\mathbf{x},\mathbf{y})\in\mathscr{S}\right)\cdot\mathbb{P}\left(\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}-\mathrm{Dec}\left(\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}\right)=\mathbf{w}_{\mathsf{lsc}}\right)\\ &\qquad\cdot\cos\left(\tfrac{2\pi}{q}\left(\left\langle\mathbf{x},\mathbf{b}-\mathbf{A}_{\mathsf{enu}}\widetilde{\mathbf{s}_{\mathsf{enu}}}\right\rangle-\left\langle\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}-\mathbf{w}_{\mathsf{lsc}},\widetilde{\mathbf{s}_{\mathsf{fft}}}\right\rangle\right)\right)\end{aligned}$$

where the last equality holds because of the independence of the decoding noise $\mathbf{u}-\mathrm{Dec}(\mathbf{u})$ from $\mathbf{u}$. Next, we simplify the inner expression inside the cosine function:

$$\begin{aligned}&\left\langle\mathbf{x},\mathbf{b}-\mathbf{A}_{\mathsf{enu}}\widetilde{\mathbf{s}_{\mathsf{enu}}}\right\rangle-\left\langle\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}-\mathbf{w}_{\mathsf{lsc}},\widetilde{\mathbf{s}_{\mathsf{fft}}}\right\rangle\\ &=\left\langle\mathbf{x},\mathbf{e}+\mathbf{A}_{\mathsf{enu}}(\mathbf{s}_{\mathsf{enu}}-\widetilde{\mathbf{s}_{\mathsf{enu}}})+\mathbf{A}_{\mathsf{lat}}\mathbf{s}_{\mathsf{lat}}+\mathbf{A}_{\mathsf{fft}}\mathbf{s}_{\mathsf{fft}}\right\rangle-\left\langle\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}-\mathbf{w}_{\mathsf{lsc}},\widetilde{\mathbf{s}_{\mathsf{fft}}}\right\rangle\\ &=\left\langle\mathbf{x},\mathbf{e}+\mathbf{A}_{\mathsf{enu}}(\mathbf{s}_{\mathsf{enu}}-\widetilde{\mathbf{s}_{\mathsf{enu}}})+\mathbf{A}_{\mathsf{fft}}(\mathbf{s}_{\mathsf{fft}}-\widetilde{\mathbf{s}_{\mathsf{fft}}})\right\rangle+\left\langle\mathbf{A}_{\mathsf{lat}}^\top\mathbf{x},\mathbf{s}_{\mathsf{lat}}\right\rangle+\left\langle\mathbf{w}_{\mathsf{lsc}},\widetilde{\mathbf{s}_{\mathsf{fft}}}\right\rangle\\ &=\left\langle\left(\mathbf{x},\mathbf{y},\mathbf{w}_{\mathsf{lsc}}\right),\mathbf{r}\right\rangle \bmod q\end{aligned}$$

where $\mathbf{r}$ is as defined in Equation (4.7). So, we obtain

$$\mathbb{E}_{\mathscr{S},\mathrm{Dec}}\left(F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^\top\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)\right) = \sum_{\substack{(\mathbf{x},\mathbf{y})\in\varLambda(\mathbf{B}')\\ \mathbf{w}_{\mathsf{lsc}}\in\varLambda(\mathcal{C}_{\mathsf{lsc}})+\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}}} \begin{aligned}&\mathbb{P}\left((\mathbf{x},\mathbf{y})\in\mathscr{S}\right)\cdot\mathbb{P}\left(\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}-\mathrm{Dec}\left(\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}\right)=\mathbf{w}_{\mathsf{lsc}}\right)\\ &\qquad\cdot\cos\left(2\pi\left\langle\left(\mathbf{x},\mathbf{y},\mathbf{w}_{\mathsf{lsc}}\right),\tfrac{\mathbf{r}}{q}\right\rangle\right)\end{aligned}.$$

We observe that if $(\mathbf{x},\mathbf{y})\in\varLambda(\mathbf{B}')$ and $\mathbf{w}_{\mathsf{lsc}}\in\varLambda(\mathcal{C}_{\mathsf{lsc}})+\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}$, then it follows that $(-\mathbf{x},-\mathbf{y})\in\varLambda(\mathbf{B}')$ and $-\mathbf{w}_{\mathsf{lsc}}\in\varLambda(\mathcal{C}_{\mathsf{lsc}})-\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}$. Furthermore, the probabilities involved in the formula are invariant under negation. This allows us to express the cosine function in its exponential form, which yields

$$\begin{aligned}\mathbb{E}_{\mathscr{S},\mathrm{Dec}}\left(F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^\top\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)\right) &= \sum_{\substack{(\mathbf{x},\mathbf{y})\in\varLambda(\mathbf{B}')\\ \mathbf{w}_{\mathsf{lsc}}\in\varLambda(\mathcal{C}_{\mathsf{lsc}})+\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}}} \mathbb{P}\left((\mathbf{x},\mathbf{y})\in\mathscr{S}\right)\cdot\mathbb{P}\left(\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}-\mathrm{Dec}\left(\mathbf{A}_{\mathsf{fft}}^\top\mathbf{x}\right)=\mathbf{w}_{\mathsf{lsc}}\right)\cdot e^{2i\pi\left\langle(\mathbf{x},\mathbf{y},\mathbf{w}_{\mathsf{lsc}}),\frac{\mathbf{r}}{q}\right\rangle}\\ &= \sum_{(\mathbf{w}_{\mathsf{lat}},\mathbf{w}_{\mathsf{lsc}})\in\varLambda(\mathbf{B}^{\mathsf{tmp}}_{\mathsf{global}})} f_{\mathsf{lat}}\left(\mathbf{w}_{\mathsf{lat}}\right)\cdot f_{\mathsf{lsc}}\left(\mathbf{w}_{\mathsf{lsc}}\right)\cdot e^{2i\pi\left\langle\left(\mathbf{B}'\mathbf{w}_{\mathsf{lat}},\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\right),\frac{\mathbf{r}}{q}\right\rangle}.\end{aligned}$$

Note that the function f_{lsc} is defined in terms of a vector $\mathbf{u}$, which in this case depends on $\mathbf{w}_{\mathsf{lat}}$. However, the error vector produced by our polar code decoder is independent of the vector being decoded; that means the distribution f_{lsc} does not depend on $\mathbf{u}$. Finally, following [CDMT24,DP23a], we can apply the Poisson summation formula to obtain Equation (4.5). □

Second-Level Approximation of the Score Function. By estimating the Fourier transforms $\widehat{f_{\mathsf{lat}}}$ and $\widehat{f_{\mathsf{lsc}}}$ from Lemma 4.5, we derive a new approximation of the score function:

Approximation 4.6 (Second-Level Approximation). *Based on Approximation 4.4, Assumptions 4.2 and 4.3, and assuming the Gaussian Heuristic holds, we have*

$$F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^\top\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)\approx N\cdot\sum_{\substack{i\geq 0\\ j\geq 0}} N_{i,j}\cdot\int_0^\infty \psi_{\mathsf{lsc}}(d_{\mathsf{lsc}})\cdot\varPhi_{d_{\mathsf{lsc}}}(i,j)dd_{\mathsf{lsc}} \tag{4.9}$$

where

$$\Phi_{d_{\mathsf{lsc}}}(i,j) \overset{\triangle}{=} \Upsilon_{\frac{\beta_{\mathsf{sieve}}}{2}}\left(\tfrac{2\pi}{q} d_{\mathsf{lat}} i\right) \cdot \Upsilon_{\frac{n_{\mathsf{fft}}}{2}-1}\left(\tfrac{2\pi}{q} d_{\mathsf{lsc}} j\right), \tag{4.10}$$

$$\Upsilon_n(x) \overset{\triangle}{=} \frac{\Gamma(n+1)\, J_n(x)}{(x/2)^n} = \sum_{\ell=0}^{+\infty} \frac{(-1)^\ell (x/2)^{2\ell}}{\ell! \prod_{s=1}^{\ell}(n+s)}, \tag{4.11}$$

$$\begin{aligned} N_{i,j} \overset{\triangle}{=} \Big|\Big\{ (\mathbf{w}_{\mathsf{lat}}, \mathbf{w}_{\mathsf{lsc}}) \in q\Lambda\left(\mathbf{B}_{\mathsf{global}}\right)^{\vee} + \mathbf{r}_{\mathsf{proj}} \subseteq \operatorname{span}\left(\mathbf{B}'\right) \times \mathbb{R}^{n_{\mathsf{fft}}} \\ : \ \|\mathbf{w}_{\mathsf{lat}}\| = i \ \textit{and} \ \|\mathbf{w}_{\mathsf{lsc}}\| = j \Big\}\Big|, \end{aligned} \tag{4.12}$$

with $\mathbf{r}_{\mathsf{proj}}$, *the orthogonal projection on* $\operatorname{span}(\mathbf{B}_{\mathsf{global}}) = \operatorname{span}(\mathbf{B}') \times \mathbb{R}^{n_{\mathsf{fft}}}$ *of*

$$\mathbf{r} \overset{\triangle}{=} \left(\mathbf{e} + \mathbf{A}_{\mathsf{enu}}(\mathbf{s}_{\mathsf{enu}} - \widetilde{\mathbf{s}_{\mathsf{enu}}}) + \mathbf{A}_{\mathsf{fft}}(\mathbf{s}_{\mathsf{fft}} - \widetilde{\mathbf{s}_{\mathsf{fft}}}), \mathbf{s}_{\mathsf{lat}}, \widetilde{\mathbf{s}_{\mathsf{fft}}}\right), \tag{4.13}$$

and

$$\mathbf{B}_{\mathsf{global}} \overset{\triangle}{=} \begin{array}{c|c|l} \beta_{\mathsf{sieve}} & n_{\mathsf{fft}} & \\ \hline \mathbf{B}' & \mathbf{0} & m + n_{\mathsf{lat}} \\ \hline \mathbf{A}_{\mathsf{fft}}^{\top} \cdot \mathbf{B}'_{[m]} & \mathbf{B}_{\mathsf{lsc}} & n_{\mathsf{fft}} \end{array}$$

Here:

- $\mathbf{B}' \in \mathbb{R}^{(m+n_{\mathsf{lat}}) \times \beta_{\mathsf{sieve}}}$ *is a basis of the sublattice where the sampled short vectors lie,*
- $\mathbf{B}'_{[m]}$ *consists of the first* m *rows of* $\mathbf{B}'$,
- *and* $\mathbf{B}_{\mathsf{lsc}}$ *is a basis of the lattice* $\Lambda(\mathcal{C}_{\mathsf{lsc}})$.

To justify the above approximation, we first show that under Assumptions 4.2 and 4.3, and assuming the Gaussian Heuristic holds, the following two approximations hold:

$$\widehat{f_{\mathsf{lat}}}\left(\mathbf{w}_{\mathsf{lat}}^{\vee}\right) \approx N \cdot \Upsilon_{\frac{\beta_{\mathsf{sieve}}}{2}}\left(2\pi d_{\mathsf{lat}} \left\|\mathbf{B}'\left(\mathbf{B}'^{\top}\mathbf{B}'\right)^{-1} \mathbf{w}_{\mathsf{lat}}^{\vee}\right\|\right), \tag{4.14}$$

and

$$\widehat{f_{\mathsf{lsc}}}\left(\mathbf{w}_{\mathsf{lsc}}^{\vee}\right) \approx \int_0^{\infty} \psi_{\mathsf{lsc}}(d_{\mathsf{lsc}}) \cdot \Upsilon_{\frac{n_{\mathsf{fft}}}{2}-1}\left(\tfrac{2\pi}{q} d_{\mathsf{lsc}} \left\|\mathbf{B}_{\mathsf{lsc}}^{-\top} \mathbf{w}_{\mathsf{lsc}}^{\vee}\right\|\right) d d_{\mathsf{lsc}}. \tag{4.15}$$

On the one hand, under Assumption 4.2 and the Gaussian Heuristic, we can smooth and approximate f_{lat} by

$$f_{\mathsf{lat}}\left(\mathbf{w}_{\mathsf{lat}}\right) \approx \mathbb{1}_{\leq d_{\mathsf{lat}}}\left(\mathbf{B}'\mathbf{w}_{\mathsf{lat}}\right) \cdot N \cdot \frac{\mathsf{Vol}\big(\Lambda(\mathbf{B}')\big)}{\mathsf{Vol}\Big(\mathsf{Ball}_{d_{\mathsf{lat}}}^{\beta_{\mathsf{sieve}}}\Big)}. \tag{4.16}$$

Using the radial nature of both $\mathbb{1}_{\leq d_{\mathsf{lat}}}$ and $\widehat{\mathbb{1}_{\leq d_{\mathsf{lat}}}}$, and the facts that $\|\mathbf{B}'\mathbf{w}_{\mathsf{lat}}\| = \left\|\sqrt{\mathbf{B}'^{\top}\mathbf{B}'}\mathbf{w}_{\mathsf{lat}}\right\|$ and $\left\|\sqrt{\mathbf{B}'^{\top}\mathbf{B}'}^{-\top} \mathbf{w}_{\mathsf{lat}}^{\vee}\right\| = \left\|\mathbf{B}'\left(\mathbf{B}'^{\top}\mathbf{B}'\right)^{-1} \mathbf{w}_{\mathsf{lat}}^{\vee}\right\|$, we have

$$\begin{aligned} \widehat{f_{\mathsf{lat}}}\left(\mathbf{w}_{\mathsf{lat}}^{\vee}\right) &\approx \frac{N \cdot \mathsf{Vol}\big(\Lambda(\mathbf{B}')\big)}{\mathsf{Vol}\Big(\mathsf{Ball}_{d_{\mathsf{lat}}}^{\beta_{\mathsf{sieve}}}\Big)} \int_{\mathbb{R}^{\beta_{\mathsf{sieve}}}} \mathbb{1}_{\leq d_{\mathsf{lat}}}\left(\mathbf{B}'\mathbf{w}_{\mathsf{lat}}\right) e^{-2i\pi\left\langle \mathbf{w}_{\mathsf{lat}}, \mathbf{w}_{\mathsf{lat}}^{\vee}\right\rangle} d\mathbf{w}_{\mathsf{lat}} \\ &= \frac{N \cdot \mathsf{Vol}\big(\Lambda(\mathbf{B}')\big)}{\mathsf{Vol}\Big(\mathsf{Ball}_{d_{\mathsf{lat}}}^{\beta_{\mathsf{sieve}}}\Big)} \int_{\mathbb{R}^{\beta_{\mathsf{sieve}}}} \mathbb{1}_{\leq d_{\mathsf{lat}}}\left(\sqrt{\mathbf{B}'^{\top}\mathbf{B}'}\mathbf{w}_{\mathsf{lat}}\right) e^{-2i\pi\left\langle \mathbf{w}_{\mathsf{lat}}, \mathbf{w}_{\mathsf{lat}}^{\vee}\right\rangle} d\mathbf{w}_{\mathsf{lat}} \\ &= \frac{N \cdot \mathsf{Vol}\big(\Lambda(\mathbf{B}')\big)}{\mathsf{Vol}\Big(\mathsf{Ball}_{d_{\mathsf{lat}}}^{\beta_{\mathsf{sieve}}}\Big) \cdot \sqrt{\det(\mathbf{B}'^{\top}\mathbf{B}')}} \cdot \int_{\mathbb{R}^{\beta_{\mathsf{sieve}}}} \mathbb{1}_{\leq d_{\mathsf{lat}}}\left(\mathbf{v}\right) e^{-2i\pi\left\langle \sqrt{\mathbf{B}'^{\top}\mathbf{B}'}^{-1}\mathbf{v}, \mathbf{w}_{\mathsf{lat}}^{\vee}\right\rangle} d\mathbf{v} \\ &= \frac{N}{\mathsf{Vol}\Big(\mathsf{Ball}_{d_{\mathsf{lat}}}^{\beta_{\mathsf{sieve}}}\Big)} \cdot \widehat{\mathbb{1}_{\leq d_{\mathsf{lat}}}}\left(\sqrt{\mathbf{B}'^{\top}\mathbf{B}'}^{-\top} \mathbf{w}_{\mathsf{lat}}^{\vee}\right) \\ &= \frac{N}{\mathsf{Vol}\Big(\mathsf{Ball}_{d_{\mathsf{lat}}}^{\beta_{\mathsf{sieve}}}\Big)} \cdot \widehat{\mathbb{1}_{\leq d_{\mathsf{lat}}}}\left(\mathbf{B}'\left(\mathbf{B}'^{\top}\mathbf{B}'\right)^{-1} \mathbf{w}_{\mathsf{lat}}^{\vee}\right) \end{aligned}$$

where the Fourier transform of the indicator function of a ball can be expressed in term of the Bessel function:

$$\begin{aligned}\widehat{\mathbb{1}_{\le d_{\mathsf{lat}}}}\left(\mathbf{w}\right) &= \left(\frac{d_{\mathsf{lat}}}{\|\mathbf{w}\|}\right)^{\frac{\beta_{\mathsf{sieve}}}{2}} \cdot J_{\frac{\beta_{\mathsf{sieve}}}{2}}\left(2\pi d_{\mathsf{lat}}\left\|\mathbf{w}\right\|\right) \\ &= \mathsf{Vol}\left(\mathsf{Ball}^{\beta_{\mathsf{sieve}}}_{d_{\mathsf{lat}}}\right) \cdot \Upsilon_{\frac{\beta_{\mathsf{sieve}}}{2}}\left(2\pi d_{\mathsf{lat}}\left\|\mathbf{w}\right\|\right).\end{aligned}$$

On the other hand, based on Assumption 4.3, we state the following approximation for f_{lsc}:

$$\begin{aligned}f_{\mathsf{lsc}}\left(\mathbf{w}_{\mathsf{lsc}}\right) &\overset{\triangle}{=} \mathbb{P}\left(\mathbf{u} - \mathrm{Dec}\left(\mathbf{u}\right) = \mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\right) \\ &= \mathbb{P}\left(\left\|\mathbf{u} - \mathrm{Dec}\left(\mathbf{u}\right)\right\| = \left\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\right\|\right) \cdot \mathbb{P}\left(\mathbf{u} - \mathrm{Dec}\left(\mathbf{u}\right) = \mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}} \;\middle|\; \left\|\mathbf{u} - \mathrm{Dec}\left(\mathbf{u}\right)\right\| = \left\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\right\|\right) \\ &\approx \left(\int_{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|}^{\sqrt{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|^2+1}} \psi_{\mathsf{lsc}}\left(d_{\mathsf{lsc}}\right) dd_{\mathsf{lsc}}\right) \cdot \frac{\mathsf{Vol}(\varLambda(\mathcal{C}_{\mathsf{lsc}}))}{\mathsf{Vol}\left(\mathsf{Ball}^{n_{\mathsf{fft}}}_{\sqrt{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|^2+1}}\right) - \mathsf{Vol}\left(\mathsf{Ball}^{n_{\mathsf{fft}}}_{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|}\right)}\end{aligned}$$

where ψ_{lsc} is the probability density function of the normal distribution $\mathcal{N}\left(\mu_{\mathsf{lsc}}, \sigma^2_{\mathsf{lsc}}\right)$. In the context of our dual attack on Kyber, the decoding distance $\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|$ takes values in the hundreds or even thousands, as shown in Appendix C, Table C.1. Consequently, the difference $\sqrt{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|^2 + 1} - \|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|$ is small, allowing us to approximate $f_{\mathsf{lsc}}\left(\mathbf{w}_{\mathsf{lsc}}\right)$ by

$$\begin{aligned}f_{\mathsf{lsc}}\left(\mathbf{w}_{\mathsf{lsc}}\right) &\approx \frac{\left(\sqrt{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|^2 + 1} - \|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|\right) \cdot \psi_{\mathsf{lsc}}\left(\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|\right) \cdot \mathsf{Vol}\left(\varLambda(\mathcal{C}_{\mathsf{lsc}})\right)}{\left(\sqrt{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|^2 + 1} - \|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|\right) \cdot \mathsf{Vol}\left(\mathsf{Sphere}^{n_{\mathsf{fft}}}_{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|}\right)} \\ &= \psi_{\mathsf{lsc}}\left(\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|\right) \cdot \frac{\mathsf{Vol}\left(\varLambda(\mathcal{C}_{\mathsf{lsc}})\right)}{\mathsf{Vol}\left(\mathsf{Sphere}^{n_{\mathsf{fft}}}_{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|}\right)}\end{aligned}$$

Note that we must have

$$\sum_{\mathbf{w}_{\mathsf{lsc}} \in \mathbb{Z}^{n_{\mathsf{fft}}}} \mathbb{P}\left(\mathbf{u} - \mathrm{Dec}\left(\mathbf{u}\right) = \mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\right) = 1.$$

Thus, the smoothed approximation of f_{lsc} should represent a probability density function, as we can verify by the following[10]:

$$\begin{aligned}\int_{\mathbb{R}^{n_{\mathsf{fft}}}} \psi_{\mathsf{lsc}}\left(\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|\right) \cdot \frac{\mathsf{Vol}(\varLambda(\mathcal{C}_{\mathsf{lsc}}))}{\mathsf{Vol}\left(\mathsf{Sphere}^{n_{\mathsf{fft}}}_{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|}\right)} d\mathbf{w}_{\mathsf{lsc}} &= \int_{\mathbb{R}^{n_{\mathsf{fft}}}} \psi_{\mathsf{lsc}}\left(\|\mathbf{w}\|\right) \cdot \frac{\mathsf{Vol}(\varLambda(\mathcal{C}_{\mathsf{lsc}}))}{\mathsf{Vol}\left(\mathsf{Sphere}^{n_{\mathsf{fft}}}_{\|\mathbf{w}\|}\right)} \cdot \frac{1}{\det(\mathbf{B}_{\mathsf{lsc}})} d\mathbf{w} \\ &= \int_{\mathbb{R}^{n_{\mathsf{fft}}}} \psi_{\mathsf{lsc}}\left(\|\mathbf{w})\|\right. \cdot \frac{1}{\mathsf{Vol}\left(\mathsf{Sphere}^{n_{\mathsf{fft}}}_{\|\mathbf{w}\|)}\right)} d\mathbf{w} \\ &= \int_0^\infty \psi_{\mathsf{lsc}}\left(d_{\mathsf{lsc}}\right) \int_{\mathsf{Sphere}^{n_{\mathsf{fft}}}_{d_{\mathsf{lsc}}}} \frac{1}{\mathsf{Vol}\left(\mathsf{Sphere}^{n_{\mathsf{fft}}}_{d_{\mathsf{lsc}}}\right)} d\sigma(\mathbf{s}) dd_{\mathsf{lsc}} \\ &= \int_0^\infty \psi_{\mathsf{lsc}}\left(d_{\mathsf{lsc}}\right) dd_{\mathsf{lsc}} \\ &\approx 1\end{aligned}$$

[10] We verified that $\int_{-\infty}^{0} \psi_{\mathsf{lsc}}(r)dr$ is negligible.

where $d\sigma(\mathbf{s})$ represents the classical Lebesgue measure on the sphere $\mathsf{Sphere}^{n_{\mathsf{fft}}}_{d_{\mathsf{lsc}}}$. Finally, by using [CE01, Prop.2.1], we have

$$\begin{aligned}
\widehat{f_{\mathsf{lsc}}}\left(\mathbf{w}_{\mathsf{lsc}}^{\vee}\right) &\approx \int_{\mathbb{R}^{n_{\mathsf{fft}}}} \psi_{\mathsf{lsc}}\left(\left\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\right\|\right) \cdot \frac{\mathsf{Vol}(\varLambda(\mathcal{C}_{\mathsf{lsc}}))}{\mathsf{Vol}\left(\mathsf{Sphere}^{n_{\mathsf{fft}}}_{\|\mathbf{B}_{\mathsf{lsc}}\mathbf{w}_{\mathsf{lsc}}\|}\right)} \cdot e^{-2i\pi\left\langle \mathbf{w}_{\mathsf{lsc}}^{\vee}, \mathbf{w}_{\mathsf{lsc}}\right\rangle} d\mathbf{w}_{\mathsf{lsc}} \\
&= \int_{\mathbb{R}^{n_{\mathsf{fft}}}} \psi_{\mathsf{lsc}}\left(\|\mathbf{w}\|\right) \cdot \frac{\mathsf{Vol}(\varLambda(\mathcal{C}_{\mathsf{lsc}}))}{\mathsf{Vol}\left(\mathsf{Sphere}^{n_{\mathsf{fft}}}_{\|\mathbf{w}\|}\right)} \cdot e^{-2i\pi\left\langle \mathbf{w}_{\mathsf{lsc}}^{\vee}, \mathbf{B}_{\mathsf{lsc}}^{-1}\mathbf{w}_{\mathsf{lsc}}\right\rangle} \cdot \frac{1}{\det(\mathbf{B}_{\mathsf{lsc}})} d\mathbf{w} \\
&= \int_{\mathbb{R}^{n_{\mathsf{fft}}}} \psi_{\mathsf{lsc}}\left(\|\mathbf{w}\|\right) \cdot \frac{1}{\mathsf{Vol}\left(\mathsf{Sphere}^{n_{\mathsf{fft}}}_{\|\mathbf{w}\|}\right)} \cdot e^{-2i\pi\left\langle \mathbf{B}_{\mathsf{lsc}}^{-\top}\mathbf{w}_{\mathsf{lsc}}^{\vee}, \mathbf{w}_{\mathsf{lsc}}\right\rangle} d\mathbf{w} \\
&= \frac{2\pi}{\left\|\mathbf{B}_{\mathsf{lsc}}^{-\top}\mathbf{w}_{\mathsf{lsc}}^{\vee}\right\|^{\frac{n_{\mathsf{fft}}}{2}-1}} \int_0^{\infty} \psi_{\mathsf{lsc}}(d_{\mathsf{lsc}}) \cdot \frac{\mathsf{Vol}(\varLambda(\mathcal{C}_{\mathsf{lsc}}))}{\mathsf{Vol}\left(\mathsf{Sphere}^{n_{\mathsf{fft}}}_{d_{\mathsf{lsc}}}\right)} \cdot {d_{\mathsf{lsc}}}^{\frac{n_{\mathsf{fft}}}{2}} \cdot J_{\frac{n_{\mathsf{fft}}}{2}-1}\left(2\pi d_{\mathsf{lsc}}\left\|\mathbf{B}_{\mathsf{lsc}}^{-\top}\mathbf{w}_{\mathsf{lsc}}^{\vee}\right\|\right) dd_{\mathsf{lsc}} \\
&= \int_0^{\infty} \psi_{\mathsf{lsc}}(d_{\mathsf{lsc}}) \cdot \varUpsilon_{\frac{n_{\mathsf{fft}}}{2}-1}\left(2\pi d_{\mathsf{lsc}}\left\|\mathbf{B}_{\mathsf{lsc}}^{-\top}\mathbf{w}_{\mathsf{lsc}}^{\vee}\right\|\right) dd_{\mathsf{lsc}}.
\end{aligned}$$

Finally, by leveraging Approximation 4.4 and Lemma 4.5, together with Approximations (4.14) and (4.15) for $\widehat{f_{\mathsf{lat}}}$ and $\widehat{f_{\mathsf{lsc}}}$, we can complete the justification for Approximation 4.6. First, we observe that for all $(\mathbf{w}_{\mathsf{lat}}^{\vee}, \mathbf{w}_{\mathsf{lsc}}^{\vee}) \in \varLambda(\mathbf{B}_{\mathsf{global}}^{\mathsf{tmp}})^{\vee} + \frac{\left(\mathbf{B}'^{\top}\mathbf{r}_{\mathsf{lat}}, \mathbf{B}_{\mathsf{lsc}}^{\top}\mathbf{r}_{\mathsf{lsc}}\right)}{q}$, we have

$$q\left(\mathbf{B}'\left(\mathbf{B}'^{\top}\mathbf{B}'\right)^{-1}\mathbf{w}_{\mathsf{lat}}^{\vee}, \mathbf{B}_{\mathsf{lsc}}^{-\top}\mathbf{w}_{\mathsf{lsc}}^{\vee}\right) \in q\varLambda(\mathbf{B}_{\mathsf{global}})^{\vee} + \mathbf{r}_{\mathsf{proj}},$$

And so, combining all the terms and making the appropriate variable change, we obtain

$$F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^{\top}\widetilde{\mathbf{s}_{\mathsf{fft}}}\right) \approx N \cdot \int_0^{\infty} \psi_{\mathsf{lsc}}(d_{\mathsf{lsc}}) \cdot \sum_{\substack{\left(\mathbf{w}_{\mathsf{lat}}^{\vee}, \mathbf{w}_{\mathsf{lsc}}^{\vee}\right) \\ \in q\varLambda(\mathbf{B}_{\mathsf{global}})^{\vee} + \mathbf{r}_{\mathsf{proj}}}} \varUpsilon_{\frac{\beta_{\mathsf{sieve}}}{2}}\left(\frac{2\pi}{q} d_{\mathsf{lat}}\left\|\mathbf{w}_{\mathsf{lat}}^{\vee}\right\|\right) \cdot \varUpsilon_{\frac{n_{\mathsf{fft}}}{2}-1}\left(\frac{2\pi}{q} d_{\mathsf{lsc}}\left\|\mathbf{w}_{\mathsf{lsc}}^{\vee}\right\|\right) dd_{\mathsf{lsc}}.$$

We conclude by noting that the inner sum depends only on the lengths of $\mathbf{w}_{\mathsf{lat}}^{\vee}$ and $\mathbf{w}_{\mathsf{lsc}}^{\vee}$, that we denote i and j, respectively.

Third-Level Approximation of the Score Function. In Appendix B, we provide an initial intuition for why the good guess can be distinguished from the wrong ones, based on Approximation 4.6. However, here we present more precise calculations.

Recall that Approximation 4.6 gives

$$F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^{\top}\widetilde{\mathbf{s}_{\mathsf{fft}}}\right) \approx N \cdot \int_0^{\infty} \psi_{\mathsf{lsc}}(d_{\mathsf{lsc}}) \cdot \sum_{i,j} N_{i,j} \cdot \varPhi_{d_{\mathsf{lsc}}}(i,j) dd_{\mathsf{lsc}} \tag{4.17}$$

where $N_{i,j}$ is a random variable representing the number of pairs $(\mathbf{w}_{\mathsf{lat}}^{\vee}, \mathbf{w}_{\mathsf{lsc}}^{\vee}) \in q\varLambda\left(\mathbf{B}_{\mathsf{global}}\right)^{\vee} + \mathbf{r}_{\mathsf{proj}} \subseteq \mathrm{span}\left(\mathbf{B}'\right) \times \mathbb{R}^{n_{\mathsf{fft}}}$ such that $\|\mathbf{w}_{\mathsf{lat}}^{\vee}\| = i$ and $\|\mathbf{w}_{\mathsf{lsc}}^{\vee}\| = j$.

Describing the distribution in Equation (4.17) is quite complex. To simplify this, we propose the following model:

Model 4.7. *We assume that* $F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^{\top}\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)$ *approximately follows the same distribution as* $\mathcal{D} + \mathcal{N}(0, N/2)$*, where* $\mathcal{N}(0, N/2)$ *denotes a normal distribution with mean* 0 *and standard deviation* $\sqrt{N/2}$*, and*

$$\mathcal{D} \triangleq N \cdot \int_0^{\infty} \psi_{\mathsf{lsc}}(d_{\mathsf{lsc}}) \cdot \left(\max_{i,j \,:\, N_{i,j}=1}\left(\varPhi_{d_{\mathsf{lsc}}}(i,j)\right)\right) dd_{\mathsf{lsc}}. \tag{4.18}$$

We recall that ψ_{lsc} *refers to the probability density function of* $\mathcal{N}(\mu_{\mathsf{lsc}}, \sigma_{\mathsf{lsc}}^2)$*.*

Based on Approximation 4.6 and Model 4.7, we can make the following two approximations, with probabilities calculated over the randomness of the guesses I_{enu} and $\widetilde{\mathbf{s}_{\mathsf{fft}}}$, as well as over the randomness of the LWE instance:

Approximation 4.8 (Good Guess). *If we make the good guess* $(\widetilde{\mathbf{s}_{\mathsf{enu}}}, \widetilde{\mathbf{s}_{\mathsf{fft}}}) = (\mathbf{s}_{\mathsf{enu}}, \mathbf{s}_{\mathsf{fft}})$ *and if we choose* T *around the expectation of* $F^{(\mathsf{lsc})}_{\mathbf{s}_{\mathsf{enu}}}\left(\mathbf{G}^\top \mathbf{s}_{\mathsf{fft}}\right)$*, namely by defining*

$$\frac{T}{N} = \frac{\exp\left(\frac{-\alpha(\pi\mu_{\mathsf{lsc}}/q)^2}{1+2\alpha(\pi\sigma_{\mathsf{lsc}}/q)^2}\right)}{\sqrt{1+2\alpha(\pi\sigma_{\mathsf{lsc}}/q)^2}} \cdot \int_0^1 \beta_{\mathsf{sieve}} \cdot t^{\beta_{\mathsf{sieve}}-1} \cdot e^{-\alpha\left(\frac{\pi d_{\mathsf{lat}} t}{q}\right)^2} dt, \tag{4.19}$$

then

$$P_{\mathsf{good}} \triangleq \mathbb{P}\left(F^{(\mathsf{lsc})}_{\mathbf{s}_{\mathsf{enu}}}\left(\mathbf{G}^\top \mathbf{s}_{\mathsf{fft}}\right) \geq T\right) \approx 0.5 \tag{4.20}$$

Approximation 4.9 (Wrong Guess). *If we make the wrong guess* $(\widetilde{\mathbf{s}_{\mathsf{enu}}}, \widetilde{\mathbf{s}_{\mathsf{fft}}}) \neq (\mathbf{s}_{\mathsf{enu}}, \mathbf{s}_{\mathsf{fft}})$*, then*

$$P_{\mathsf{wrong}} \triangleq \mathbb{P}\left(F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^\top \widetilde{\mathbf{s}_{\mathsf{fft}}}\right) \geq T\right) \tag{4.21}$$

$$\approx \int_{-\infty}^{+\infty}\int_0^{+\infty} \min\left(1, \int_{\mathcal{E}(T-t)} \lambda(x)\mu(y) d(x,y)\right) \cdot \frac{e^{\frac{-t^2}{N} - \frac{(d_{\mathsf{lsc}}-\mu_{\mathsf{lsc}})^2}{2\sigma^2_{\mathsf{lsc}}}}}{\pi\sigma_{\mathsf{lsc}}\sqrt{2N}} dd_{\mathsf{lsc}}\, dt \tag{4.22}$$

where

$$\mathcal{E}(T-t) \triangleq \left\{(x,y) \in \mathbb{R}^2_+ \ : \ N \cdot \varPhi_{d_{\mathsf{lsc}}}(x,y) \geq T-t\right\}, \tag{4.23}$$

$$\lambda(x) \triangleq \frac{2 \cdot \delta\left(\beta_{\mathsf{bkz}}\right)^{\beta_{\mathsf{sieve}}(m+n_{\mathsf{lat}}-\beta_{\mathsf{sieve}})} \cdot \pi^{\frac{\beta_{\mathsf{sieve}}}{2}} \cdot x^{\beta_{\mathsf{sieve}}-1}}{q^{\beta_{\mathsf{sieve}} \cdot \frac{m}{m+n_{\mathsf{lat}}}} \cdot \varGamma\left(\frac{\beta_{\mathsf{sieve}}}{2}\right)} \qquad \textit{and} \qquad \mu(y) \triangleq \frac{2 \cdot \pi^{\frac{n_{\mathsf{fft}}}{2}} \cdot y^{n_{\mathsf{fft}}-1}}{q^{k_{\mathsf{fft}}} \cdot \varGamma\left(\frac{n_{\mathsf{fft}}}{2}\right)}. \tag{4.24}$$

Justification of Approximation 4.8. The rationale behind Approximation 4.8 is that if we make the good guess $(\widetilde{\mathbf{s}_{\mathsf{enu}}}, \widetilde{\mathbf{s}_{\mathsf{fft}}}) = (\mathbf{s}_{\mathsf{enu}}, \mathbf{s}_{\mathsf{fft}})$, then there exists an element

$$\mathbf{r}_{\mathsf{proj}} \triangleq \left(\mathbf{P}(\mathbf{e}, \mathbf{s}_{\mathsf{lat}}), \mathbf{s}_{\mathsf{fft}}\right) \in q\varLambda\left(\mathbf{B}_{\mathsf{global}}\right)^\vee + \mathbf{r}_{\mathsf{proj}} \tag{4.25}$$

which has particularly small length. Specifically, the quantities $\frac{\|\mathbf{P}(\mathbf{e},\mathbf{s}_{\mathsf{lat}})\|}{\sqrt{\frac{\alpha}{2}}}$ and $\frac{\|\mathbf{s}_{\mathsf{fft}}\|}{\sqrt{\frac{\alpha}{2}}}$ approximately follow a χ-distribution[11] with degrees of freedom β_{sieve} and n_{fft}, respectively. Therefore, we obtain

$$\begin{aligned}\mathbb{E}_{\chi_{\beta_{\mathsf{sieve}}},\chi_{n_{\mathsf{fft}}}}\left(F^{(\mathsf{lsc})}_{\mathbf{s}_{\mathsf{enu}}}\left(\mathbf{G}^\top \mathbf{s}_{\mathsf{fft}}\right)\right) &\approx \mathbb{E}_{\chi_{\beta_{\mathsf{sieve}}},\chi_{n_{\mathsf{fft}}}}\left(\mathcal{D}\right) \\ &\approx N \cdot \mathbb{E}_{d_{\mathsf{lsc}}}\left(\mathbb{E}_{\chi_{\beta_{\mathsf{sieve}}},\chi_{n_{\mathsf{fft}}}}\left(\varPhi_{d_{\mathsf{lsc}}}\left(\sqrt{\tfrac{\alpha}{2}}\chi_{\beta_{\mathsf{sieve}}}, \sqrt{\tfrac{\alpha}{2}}\chi_{n_{\mathsf{fft}}}\right)\right)\right) \\ &= N \cdot \mathbb{E}_{\chi_{\beta_{\mathsf{sieve}}}}\left(\varUpsilon_{\frac{\beta_{\mathsf{sieve}}}{2}}\left(\tfrac{2\pi}{q} d_{\mathsf{lat}}\sqrt{\tfrac{\alpha}{2}}\chi_{\beta_{\mathsf{sieve}}}\right)\right) \\ &\qquad \cdot \mathbb{E}_{\chi_{n_{\mathsf{fft}}},d_{\mathsf{lsc}}}\left(\varUpsilon_{\frac{n_{\mathsf{fft}}}{2}-1}\left(\tfrac{2\pi}{q} d_{\mathsf{lsc}}\sqrt{\tfrac{\alpha}{2}}\chi_{n_{\mathsf{fft}}}\right)\right)\end{aligned}$$

where $\mathbb{E}_{d_{\mathsf{lsc}}}(\cdot)$ denotes the expectation with respect to the random variable d_{lsc}, which follows a normal distribution $\mathcal{N}(\mu_{\mathsf{lsc}}, \sigma^2_{\mathsf{lsc}})$.

[11] Strictly speaking, it is not an exact χ-distribution since the coordinates of $\mathbf{r}_{\mathsf{proj}}$ are not precisely normally distributed.

Using Equation (4.11), each term in the above equation can be expressed in terms of the moments of the χ^2-distribution:

$$\begin{aligned}
\mathbb{E}_{\chi_{n_{\mathsf{fft}}}, d_{\mathsf{lsc}}}\left(\Upsilon_{\frac{n_{\mathsf{fft}}}{2}-1}\left(\frac{2\pi}{q} d_{\mathsf{lsc}}\sqrt{\frac{\alpha}{2}}\chi_{n_{\mathsf{fft}}}\right)\right) &= \mathbb{E}_{d_{\mathsf{lsc}}}\left(\sum_{\ell=0}^{+\infty}\frac{\left(-\alpha\left(\frac{\pi d_{\mathsf{lsc}}}{q}\right)^2\right)^\ell}{\ell!}\cdot\frac{\mathbb{E}\left(\chi_{n_{\mathsf{fft}}}^{2\ell}\right)}{2^\ell\prod_{s=1}^{\ell}\left(\frac{n_{\mathsf{fft}}}{2}-1+s\right)}\right)\\
&= \mathbb{E}_{d_{\mathsf{lsc}}}\left(\sum_{\ell=0}^{+\infty}\frac{\left(-\alpha\left(\frac{\pi d_{\mathsf{lsc}}}{q}\right)^2\right)^\ell}{\ell!}\right)\\
&= \mathbb{E}_{d_{\mathsf{lsc}}}\left(e^{-\alpha\left(\frac{\pi d_{\mathsf{lsc}}}{q}\right)^2}\right)\\
&= \frac{\exp\left(\frac{-\alpha(\pi\mu_{\mathsf{lsc}}/q)^2}{1+2\alpha(\pi\sigma_{\mathsf{lsc}}/q)^2}\right)}{\sqrt{1+2\alpha(\pi\sigma_{\mathsf{lsc}}/q)^2}}
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}_{\chi_{\beta_{\mathsf{sieve}}}}\left(\Upsilon_{\frac{\beta_{\mathsf{sieve}}}{2}}\left(\frac{2\pi}{q} d_{\mathsf{lat}}\sqrt{\frac{\alpha}{2}}\chi_{\beta_{\mathsf{sieve}}}\right)\right) &= \sum_{\ell=0}^{+\infty}\frac{\left(-\alpha\left(\frac{\pi d_{\mathsf{lat}}}{q}\right)^2\right)^\ell}{\ell!}\cdot\frac{\mathbb{E}\left(\chi_{\beta_{\mathsf{sieve}}}^{2\ell}\right)}{2^\ell\prod_{s=1}^{\ell}\left(\frac{\beta_{\mathsf{sieve}}}{2}+s\right)}\\
&= \sum_{\ell=0}^{+\infty}\frac{\left(-\alpha\left(\frac{\pi d_{\mathsf{lat}}}{q}\right)^2\right)^\ell}{\ell!}\cdot\frac{\prod_{s=0}^{\ell-1}\left(\frac{\beta_{\mathsf{sieve}}}{2}+s\right)}{\prod_{s=0}^{\ell-1}\left(\frac{\beta_{\mathsf{sieve}}}{2}+1+s\right)}\\
&= \int_0^1 \beta_{\mathsf{sieve}}\cdot t^{\beta_{\mathsf{sieve}}-1}\cdot e^{-\alpha\left(\frac{\pi d_{\mathsf{lat}} t}{q}\right)^2}\,dt
\end{aligned}$$

where the last equality is a well-known result concerning generalized hypergeometric functions[12].

Finally, in Equation (4.19), we chose T as the expected score $\mathbb{E}\left(F^{(\mathsf{lsc})}_{\mathbf{s}_{\mathsf{enu}}}\left(\mathbf{G}^\top\mathbf{s}_{\mathsf{fft}}\right)\right)$ of the good guess. Through experimentation, we verified that the expectation of this score is approximately equal to its median, which justifies Approximation 4.8.

Justification of Approximation 4.9. On the other hand, Approximation 4.9 is obtained by estimating the length of the short vectors in $q\Lambda\left(\mathbf{B}_{\mathsf{global}}\right)^\vee+\mathbf{r}_{\mathsf{proj}}$, where $\mathbf{r}_{\mathsf{proj}}$ is no longer the shortest vector in the lattice coset. For i and j small enough, we can make the approximation that

$$\mathbb{P}\left(N_{i,j}>0\right)\approx\mathbb{P}\left(N_{i,j}=1\right). \tag{4.26}$$

Thus, the survival function of $\mathcal{D}$, knowing that the achieved decoding distance is d_{lsc}, can be approximated by

$$\begin{aligned}
\mathbb{P}\left(\mathcal{D}>T\mid d_{\mathsf{lsc}}\right) &\approx \mathbb{P}\left(\exists(i,j)\in\mathcal{E}(T)\ :\ N_{i,j}>0\right) && (4.27)\\
&\approx \min\left(1,\mathbb{E}\left(\sum_{\substack{(i,j)\in\mathcal{E}(T)\\(i^2,j^2)\in\mathbb{N}^2}} N_{i,j}\right)\right) && (4.28)
\end{aligned}$$

[12] Note that $\int_0^1 \beta_{\mathsf{sieve}}\cdot t^{\beta_{\mathsf{sieve}}-1}e^{-\alpha\left(\frac{\pi d_{\mathsf{lat}} t}{q}\right)^2}\,dt=\mathbb{E}_{t\sim\mathrm{Unif}\left(\mathsf{Ball}^{\beta_{\mathsf{sieve}}}_{d_{\mathsf{lat}}}\right)}\left(e^{-\alpha\left(\frac{\pi t}{q}\right)^2}\right)$.

where

$$\mathcal{E}(T) \triangleq \left\{(i,j) \in \mathbb{R}_+^2 \ : \ N \cdot \Phi_{d_{\mathsf{lsc}}}(i,j) \geq T\right\}. \tag{4.29}$$

We have already observed that when we make the correct guess, the probability $\mathbb{P}\left(N_{i,j} > 0\right)$ is particularly high for a pair (i,j) close to $\left(\sqrt{\frac{\alpha\beta_{\mathsf{sieve}}}{2}}, \sqrt{\frac{\alpha n_{\mathsf{lsc}}}{2}}\right)$. Now, in the case where we make a wrong guess – that is $(\widetilde{\mathbf{s}_{\mathsf{enu}}}, \widetilde{\mathbf{s}_{\mathsf{fft}}}) \neq (\mathbf{s}_{\mathsf{enu}}, \mathbf{s}_{\mathsf{fft}})$ – then we have

$$\mathbb{E}\left(\sum_{\substack{(i,j)\in\mathcal{E}(T)\\(i^2,j^2)\in\mathbb{N}^2}} N_{i,j}\right) \approx \frac{\int_{\mathcal{E}(T)} \mathsf{Vol}\left(\mathsf{Sphere}_x^{\beta_{\mathsf{sieve}}}\right) \cdot \mathsf{Vol}\left(\mathsf{Sphere}_y^{n_{\mathsf{fft}}}\right) d(x,y)}{\mathsf{Vol}\left(q\Lambda(\mathbf{B}_{\mathsf{global}})^{\vee}\right)}$$

$$= \int_{\mathcal{E}(T)} \frac{\mathsf{Vol}\left(\mathsf{Sphere}_x^{\beta_{\mathsf{sieve}}}\right) \cdot \mathsf{Vol}\left(\Lambda(\mathbf{B}')\right)}{q^{\beta_{\mathsf{sieve}}}} \cdot \frac{\mathsf{Vol}\left(\mathsf{Sphere}_y^{n_{\mathsf{fft}}} \cdot \mathsf{Vol}(\Lambda(\mathbf{B}_{\mathsf{lsc}}))\right)}{q^{n_{\mathsf{fft}}}} d(x,y)$$

The volume of $\Lambda(\mathbf{B}')$ is provided in Lemma 2.9, while the volume of $\Lambda(\mathbf{B}_{\mathsf{lsc}})$ is $q^{n_{\mathsf{fft}}-k_{\mathsf{fft}}}$. Meanwhile, the integral can be evaluated numerically. Note that the Gaussian Heuristic is necessary here, as q-ary lattices only behave approximately like random lattices.

Finally, under Model 4.7, P_{wrong} is the convolution product of the probability density function of the normal distribution $\mathcal{N}(0, N/2)$ and the survival function of $\mathcal{D}$, that is given by

$$\mathbb{P}\left(\mathcal{D} > T\right) = \int_0^{\infty} \psi_{\mathsf{lsc}}(d_{\mathsf{lsc}}) \cdot \mathbb{P}\left(\mathcal{D} > T \mid d_{\mathsf{lsc}}\right) dd_{\mathsf{lsc}}$$

Validating our Analysis Through Simulations. We verify here the soundness of Approximation 4.9 for P_{wrong}, which is crucial for estimating the number of false candidates. To this end, we implemented and ran Algorithm 3.1, computing an experimental value for P_{wrong}, namely $\frac{\left|\{\mathbf{z}\in\mathbb{Z}_q^{k_{\mathsf{fft}}} : F_{\mathbf{0}}^{(\mathsf{lsc})}(\mathbf{z})\geq T\}\right|}{q^{k_{\mathsf{fft}}}}$ for different values of T. We plotted it against its theoretical approximation in Figure 4.1. Notably, we found that the experimental and theoretical estimates are in agreement, though the plot on the right suggests that our analysis may be slightly optimistic.

We used the g6k library [ADH+19] to generate short vectors in a lattice and used polar codes, along with the decoder described in Section 3.3, for the auxiliary code $\mathcal{C}_{\mathsf{lsc}}$. We provide the program used to generate Figure 4.1 in the GitHub repository[13].

5 Application

In this section we give estimates for the cost of our attack against LWE problems from the literature. In particular, we consider NIST PQC standardized candidate KYBER [SAB+20]. We summarize in Table 5.1 the parameters of KYBER and outline the security level required by NIST, along with the attack complexities claimed by MATZOV. Note that MATZOV findings are not widely agreed upon in the cryptographic community as the analysis is based on independence assumptions which were strongly questioned in [DP23b].

The models C0, CC, and CN refer to different cost models for lattice reduction. These models are consistent with those presented in [AS22], and they can be described as follows:

C0. Cost estimates in the "Core-SVP" cost model [ADPS16] for Algorithm 2.1 using [BDGL16] as the sieving oracle. This model assumes a single SVP call suffices to reduce a lattice. It furthermore assumes that all lower-order terms in the exponent are zero.

CC. Cost estimates in a classical circuit model [AGPS20b,SAB+20,MAT22] for Algorithm 2.1 using [BDGL16] as the sieving oracle. We derive these estimates by implementing the cost estimates from [MAT22], those tagged "asymptotic" (cf. [MAB+22]). This is the most detailed

[13] https://github.com/kevin-carrier/CodedDualAttack/tree/main/verifyModel

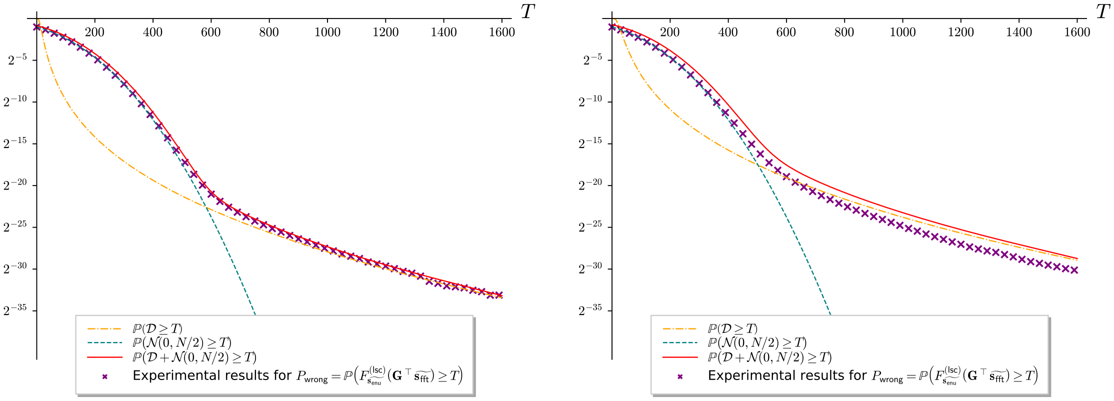

Fig. 4.1: Experimental validation of Approximation 4.9 for P_{wrong}. The solid lines represent our theoretical model, while the crosses indicate results obtained from simulations. The theoretical model is drawn using the observed values for d_{lat} and d_{lsc}. In particular, d_{lsc} follows a normal distribution $\mathcal{N}(\mu_{\mathsf{lsc}}, \sigma^2_{\mathsf{lsc}})$ with mean μ_{lsc} and standard deviation σ_{lsc} that are derived from the observed decoding distances of the $[n_{\mathsf{fft}}, k_{\mathsf{fft}}]_q$ polar codes used in the experiments. The experimental data were obtained by running 4000 iterations of Algorithm 3.1, with each iteration using an input $(\mathbf{A}, \mathbf{b})$ taken uniformly at random in $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$. The parameters used are:
– on the left: $q = 241$, $m = 40$, $n = 43$, $n_{\mathsf{lat}} = 35$, $n_{\mathsf{enu}} = 0$, $n_{\mathsf{fft}} = 8$, $k_{\mathsf{fft}} = 3$, $N = 25971$, $\beta_{\mathsf{bkz}} = 32$, $\beta_{\mathsf{sieve}} = 44$, $d_{\mathsf{lat}} = 42.00$, $\mu_{\mathsf{lsc}} = 23.94$ and $\sigma_{\mathsf{lsc}} = 3.38$,
– on the right: $q = 241$, $m = 40$, $n = 50$, $n_{\mathsf{lat}} = 42$, $n_{\mathsf{enu}} = 0$, $n_{\mathsf{fft}} = 8$, $k_{\mathsf{fft}} = 3$, $N = 25970$, $\beta_{\mathsf{bkz}} = 35$, $\beta_{\mathsf{sieve}} = 41$, $d_{\mathsf{lat}} = 58.60$, $\mu_{\mathsf{lsc}} = 23.87$ and $\sigma_{\mathsf{lsc}} = 3.30$.

cost estimate available in the literature. However, we caution that these estimates, too, ignore the cost of memory access and thus may significantly underestimate the true cost. That is RAM access is not “free” (cf. [MAB+22]). This cost model is called “list_decoding-classical” in [AGPS20b].

CN. Cost estimates in a query model for Algorithm 2.1 using [BDGL16] as the sieving oracle. We include this cost model for completeness. This cost model is called “list_decoding-naive_classical” in [AGPS20b].

These various models rely on [BDGL16] for nearest-neighbor search, using fuzzy hashing functions based on product codes. It should be added that [Duc22] pointed out that the original analysis of [BDGL16] assumes an ideal case, underestimating the decoding cost and overlooking the suboptimal rate-distortion of product codes. Specifically, [Duc22] estimates that for a sieving dimension of 380, the complexity of the nearest-neighbor search increases by a factor of about 2^6. Here we kept the same way of computing the costs as in Matzov's paper [MAT22] to keep a fair comparison with their work. Delving further into this topic and replacing for instance the simple product codes with a better approach is outside the scope of this work.

For the same reasons, we only consider the classical RAM model. Adapting our analysis to a more realistic memory access model, such as the one presented in [Jaq24], is left as future work. Furthermore, we do not compare our results with the state-of-the-art primal attacks. A fair comparison would require substantial effort, as it involves accounting for many confounding factors, such as those listed in [DP23b, Appendix A]. Achieving a more precise estimate – down to the gate count level, as done for instance in [SAB+20] – would demand a significantly more in-depth analysis, which we leave as future work given the already considerable length of this paper. Once again, our goal here is to provide a proof of concept for dual lattice attacks and to help resolve the controversy surrounding them. In line with this objective, we deliberately chose to work with a relatively simple and likely suboptimal dual attack algorithm. There are several potential avenues for improving it, including: (i) allowing false positives and adding a verification step to filter them out, (ii) combining modulus switching with our technique, as it would enable computations in characteristic 2, thereby allowing for a much more efficient fast Fourier transform, (iii) replacing the naive product code in the sieving procedure by a better quantizer... We leave these improvements for future work, as they fall outside the scope of this paper.

Evaluating the Complexity of our Dual Attack. We optimize the time complexity of Algorithm 3.1, as established in Theorem 4.1, under the assumptions outlined in Lemma 3.2. We assume that $1-\mu \approx 1$, and we constrain the parameters N and T such that $\varepsilon := R\,P_{\mathsf{wrong}}\,q^{k_{\mathsf{fft}}}$ remains close to 1; in particular, in our setting, η is always greater than 0.62. Additionally, we select T following Approximation 4.8, ensuring that $P_{\mathsf{good}} \approx \frac{1}{2}$. At the same time, we ensure that ε remains close to 0. These choices guarantee that the overall success probability[14] of our algorithm is lower-bounded by approximately $0.31-\varepsilon \approx 0.31$.

Our complexity results are summarized in the last three columns of Table 5.1. The associated parameters are given in Appendix C, Table C.1 and some relevant intermediate quantities are summarized in Appendix C, Table C.2. Note that the quantity d_{lat} is defined from the other parameters as in Lemma 2.8. The mean μ_{lsc} of the decoding distance d_{lsc} and its standard deviation σ_{lsc} are computed by choosing an $[n_{\mathsf{fft}}, k_{\mathsf{fft}}]_q$ polar code and by decoding many (a thousand) random words of $\mathbb{Z}_q^{n_{\mathsf{fft}}}$. For all instances of Kyber, we used a list size of $L=1$ in the decoder (see Section 3.3, Lemma 3.3). This list size allows us to achieve a decoding distance close enough to the optimal decoding distance $d_{\mathrm{GV}} \triangleq \sqrt{\frac{n_{\mathsf{fft}}}{2\pi e}}\,q^{1-\frac{k_{\mathsf{fft}}}{n_{\mathsf{fft}}}}$ without incurring additional cost in the complexity. Finally, we provide in the GitHub repository[15] a file verifying that our parameters achieve the complexity claims and verify all the constraints.

Scheme	LWE parameters			Security level required by NIST	MATZOV Complexity			Complexity of our Algorithm 3.1		
	q	n	α		**C0**	**CC**	**CN**	**C0**	**CC**	**CN**
Kyber-512	3329	512	3	AES-128 (143 bits)	115.4	139.2	134.4	**121.8**	**139.5**	**134.5**
Kyber-768	3329	768	2	AES-192 (207 bits)	173.7	196.1	190.6	**173.0**	**195.1**	**189.8**
Kyber-1024	3329	1024	2	AES-256 (272 bits)	241.8	262.4	256.1	**239.0**	**259.7**	**254.6**

Table 5.1: The LWE parameters for Kyber, the security level required by NIST, the claimed $\log_2$ complexity of MATZOV attack as given in [AS22, Table 2], and the $\log_2$ complexity of our dual attack Algorithm 3.1.

Acknowledgment

We would like to thank Léo Ducas for helpful discussions and for pointing out, in an older version of the paper, that we made an error by overlooking the fact that MATZOV sieving is performed on a sublattice rather than the full lattice.

[14] Note that in [MAT22], this success probability was approximately 0.25.

[15] https://github.com/kevin-carrier/CodedDualAttack/tree/main/OptimizeCodedDualAttack

References

AD221. *Lattice Attacks on NTRU and LWE: A History of Refinements*, page 15–40. London Mathematical Society Lecture Note Series. Cambridge University Press, 2021.

ADH$^+$19. Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 717–746, Cham, 2019. Springer International Publishing.

ADPS16. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, August 2016.

AFFP14. Martin R. Albrecht, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. modulus switching for the BKW algorithm on LWE. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 429–445, Heidelberg, March 2014. Springer.

AGPS20a. Martin Albrecht, Vlad Gheorghiu, Eamonn Postlethwaite, and John Schanck. *Estimating Quantum Speedups for Lattice Sieves*, pages 583–613. 12 2020.

AGPS20b. Martin R. Albrecht, Vlad Gheorghiu, Eamonn W. Postlethwaite, and John M. Schanck. Estimating quantum speedups for lattice sieves. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 583–613. Springer, Heidelberg, December 2020.

Alb17. Martin R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 103–129, 2017.

AS22. Martin R. Albrecht and Yixin Shen. Quantum augmented dual attack. Cryptology ePrint Archive, Paper 2022/656, 2022. https://eprint.iacr.org/2022/656.

BDGL16. Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 10–24. SIAM, 2016.

BV11. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106. IEEE Computer Society, 2011.

CDMT22. Kevin Carrier, Thomas Debris-Alazard, Charles Meyer-Hilfiger, and Jean-Pierre Tillich. Statistical decoding 2.0: Reducing decoding to LPN. In *Advances in Cryptology - ASIACRYPT 2022*, LNCS. Springer, 2022.

CDMT24. Kévin Carrier, Thomas Debris-Alazard, Charles Meyer-Hilfiger, and Jean-Pierre Tillich. Reduction from sparse LPN to LPN, dual attack 3.0. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, volume 14656 of *LNCS*, pages 286–315. Springer, 2024.

CE01. Henry Cohn and Noam Elkies. New upper bounds on sphere packings i. *Annals of Mathematics*, 157, 10 2001.

Che13. Yuanmi Chen. Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe. 2013.

Chi14. Mao-Ching Chiu. Non-binary polar codes with channel symbol permutations. In *2014 International Symposium on Information Theory and its Applications*, pages 433–437, 2014.

CS88. John H. Conway and Neil J. A. Sloane. *Sphere Packings, Lattices and Groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1988.

DM90. Dan E. Dudgeon and Russell M. Mersereau. *Multidimensional Digital Signal Processing*. Prentice Hall Professional Technical Reference, 1990.

DP23a. Léo Ducas and Ludo N. Pulles. Accurate score prediction for dual attacks. preprint, November 2023. preprint.

DP23b. Léo Ducas and Ludo N. Pulles. Does the dual-sieve attack on learning with errors even work? In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023*, volume 14083 of *LNCS*, pages 37–69, Santa Barbara, CA, USA, August 2023. Springer.

Duc18. Léo Ducas. Shortest vector from lattice sieving: A few dimensions for free. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 125–145, Cham, 2018. Springer International Publishing.

Duc22. Léo Ducas. Estimating the hidden overheads in the bdgl lattice sieving algorithm. In *Post-Quantum Cryptography: 13th International Workshop, PQCrypto 2022, Virtual Event, September 28–30, 2022, Proceedings*, page 480–497, Berlin, Heidelberg, 2022. Springer-Verlag.

EJK20. Thomas Espitau, Antoine Joux, and Natalia Kharchenko. On a dual/hybrid approach to small secret LWE - A dual/enumeration technique for learning with errors and application to security estimates of FHE schemes. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology - INDOCRYPT 2020 - 21st International Conference on Cryptology in India, Bangalore, India, December 13-16, 2020, Proceedings*, volume 12578 of *Lecture Notes in Computer Science*, pages 440–462. Springer, 2020.

FJ05. M. Frigo and S.G. Johnson. The design and implementation of fftw3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

GJ21. Qian Guo and Thomas Johansson. Faster dual lattice attacks for solving LWE with applications to CRYSTALS. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part IV*, volume 13093 of *Lecture Notes in Computer Science*, pages 33–62. Springer, 2021.

Jab01. Abdulrahman Al Jabri. A statistical decoding algorithm for general linear block codes. In Bahram Honary, editor, *Cryptography and coding. Proceedings of the 8^{th} IMA International Conference*, volume 2260 of *LNCS*, pages 1–8, Cirencester, UK, December 2001. Springer.

Jaq24. Samuel Jaques. Memory adds no cost to lattice sieving for computers in 3 or more spatial dimensions. *IACR Communications in Cryptology*, 10 2024.

KU10. Satish B. Korada and Rüdiger Urbanke. Polar codes are optimal for lossy source coding. *IEEE Trans. Inform. Theory*, 56(4):1751–1768, 2010.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology - EUROCRYPT2010*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.

MAB$^+$22. Matzov, Daniel Apon, Daniel J. Bernstein, Carl Mitchell, Léo Ducas, Martin Albrecht, and Chris Peikert. Improved Dual Lattice Attack. https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/Fm4cDfsx65s, 2022.

MAT22. MATZOV. Report on the Security of LWE: Improved Dual Lattice Attack, April 2022.

MR09. Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

MT23. Charles Meyer-Hilfiger and Jean-Pierre Tillich. Rigorous foundations for dual attacks in coding theory. In *Theory of Cryptography Conference, TCC 2023*, volume 14372 of *LNCS*, pages 3–32. Springer Verlag, December 2023.

PS24. Amaury Pouly and Yixin Shen. Provable dual attacks on learning with errors. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, volume 14656 of *LNCS*, pages 256–285. Springer, 2024.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93, 2005.

SAB$^+$20. Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

Sav21. Valentin Savin. Non-binary polar codes for spread-spectrum modulations. In *2021 11th International Symposium on Topics in Coding (ISTC)*, pages 1–5, 2021.

Sch03. Claus Peter Schnorr. Lattice reduction by random sampling and birthday methods. In Helmut Alt and Michel Habib, editors, *STACS 2003*, pages 145–156, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

ŞTA09. Eren Şaşoğlu, Emre Telatar, and Erdal Arıkan. Polarization for arbitrary discrete memoryless channels. In *Proc. IEEE Inf. Theory Workshop- ITW*, pages 144–149, October 2009.

ZF96. Ram Zamir and Meir Feder. On lattice quantization noise. *IEEE Trans. Inform. Theory*, 42(4):1152–1159, 1996.

Appendix

A Quantizers, Lossy Source Coding and Polar Codes

In this section we give more details about the construction of polar codes over $\mathbb{Z}_q$ that is mentioned in Subsection 3.3. We then verify that, when decoding random words, it is possible, to achieve a typical decoding distance which is very close to the lattice analogue of the Gilbert-Varshamov distance that we recall to be

$$\omega \approx \sqrt{\frac{n}{2\pi e}} q^{1-\frac{k}{n}}. \tag{A.1}$$

where n and k are respectively the length and the dimension of the polar code.

Construction of Polar Codes over $\mathbb{Z}_q$. Let assume a codeword in $\mathbb{Z}_q^n$ of which each symbol is transmitted through a Gaussian channel of standard deviation $\sigma \triangleq \frac{\omega}{\sqrt{n}}$. The polar code construction basically consists of transforming those n Gaussian channels into n virtual channels that are, for most of them, either of maximal or minimal entropy. The idea then is to fix (or in other words *freeze*) the information that will transit via the bad channels and involve the good channels for the k symbols of useful information.

Our construction essentially follows the papers [ŞTA09,Chi14,Sav21]. We refer to those articles for more details about polar codes. It is a recursive construction which can be described as follows.

Definition A.1 **($(U+V, \alpha U)$-construction).** *Let U and V be two linear codes of the same length n over $\mathbb{Z}_q$ and let $\alpha \in \mathbb{Z}_q^*$ be an invertible scalar. The $(U+V, \alpha V)$ is a $\mathbb{Z}_q$-linear code of length $2n$ defined by*

$$(U+V, \alpha U) \triangleq \{(\mathbf{u}+\mathbf{v}, \alpha \mathbf{u}) : \mathbf{u} \in U \text{ and } \mathbf{v} \in V\} \tag{A.2}$$

A polar code of length $n \triangleq 2^m$ and dimension k is then defined by

Definition A.2 **(polar code).** *Let F be a subset of $\{0,1\}^m$ of size $2^m - k$ and let α be a function mapping the binary words of length $< m$ to $\mathbb{Z}_q^*$. The polar code of length 2^m associated to F and α is defined recursively by*

$$C \triangleq U_\varepsilon \tag{A.3}$$

where the $U_\mathbf{x}$ are codes of length 1 for all $\mathbf{x} \in \{0,1\}^m$ (we denote by ε the empty binary word) and are given by

$$U_\mathbf{x} = \begin{cases} \{0\} & \text{if } \mathbf{x} \in F \\ \mathbb{Z}_q & \text{otherwise} \end{cases} \tag{A.4}$$

and the other $U_\mathbf{x}$'s where $\mathbf{x}$ is a binary word of length $< m$ are defined recursively by

$$U_\mathbf{x} \triangleq \left(U_{0||\mathbf{x}} + U_{1||\mathbf{x}}, \alpha(\mathbf{x}) U_{0||\mathbf{x}}\right). \tag{A.5}$$

Thus, a polar code is fully defined by the set F of *frozen* positions and the $\alpha(\mathbf{x})$'s. In [Chi14], it is admitted that choosing the $\alpha(\mathbf{x})$'s uniformly at random in $\mathbb{Z}_q^*$ is good enough. However, in [Sav21], it is shown that those coefficients can be optimized. Thereafter, we do not use the optimization technique from [Sav21] but simply try several polar codes then choose the best of them. On another hand, we classically determine the optimal frozen positions F using Monte-Carlo simulation: we run many times a genie-aided decoder for estimating the probability distribution of each virtual channel then selecting the worst of them; that are the $2^m - k$ virtual channels for which the error probabilities are the highest.

Decoding Polar Codes over $\mathbb{Z}_q$. The Successive Cancelation (SC) decoding algorithm (see [ŞTA09,Chi14,Sav21]) can be described as a recursive decoding algorithm. For each code $U_{\mathbf{x}} \subseteq \mathbb{Z}_q^{2^{m-t}}$ such that $\mathbf{x} \in \{0,1\}^t$ and $t \in [\![0, m-1]\!]$, we decode a noisy codeword in this code by using recursively the decoders of $U_{0||\mathbf{x}}$ and $U_{1||\mathbf{x}}$.

Let $\mathbf{c} \overset{\triangle}{=} (c_1, \cdots, c_{2^{m-t}}) \overset{\triangle}{=} (\mathbf{u}+\mathbf{v}, \alpha(\mathbf{x})\mathbf{u})$ be a codeword in $U_{\mathbf{x}}$; *i.e.* $\mathbf{u} \overset{\triangle}{=} (u_1, \cdots, u_{2^{m-t-1}})$ and $\mathbf{v} \overset{\triangle}{=} (v_1, \cdots, v_{2^{m-t-1}})$ are respectively in $U_{0||\mathbf{x}}$ and $U_{1||\mathbf{x}}$. Let assume that $\mathbf{c}$ is transmitted through a channel $W^{(\mathbf{x})}$; let

$$\mathbf{y} \overset{\triangle}{=} (y_1, \cdots, y_{2^{m-t}}) \overset{\triangle}{=} (\mathbf{y}_\ell, \mathbf{y}_r) \in \mathbb{Z}_q^{2^{m-t-1}} \times \mathbb{Z}_q^{2^{m-t-1}} \tag{A.6}$$

be the received word. We assume that for each position $i \in [\![1, 2^{m-t}]\!]$ and symbol $s \in \mathbb{Z}_q$, we know the probability that the transmitted symbol is s knowing that the received one is y_i:

$$\Pi_i^{(\mathbf{x})}(s) \overset{\triangle}{=} \mathbb{P}\left(c_i = s | y_i\right) \tag{A.7}$$

Instead of decoding directly $\mathbf{y}$, we decode first $\mathbf{y}_\ell - \alpha(\mathbf{x})^{-1}\mathbf{y}_r$ expecting to find $\mathbf{v} \in U_{1||\mathbf{x}}$. The virtual channel through which $\mathbf{v}$ has transited is then the serialization of two $W^{(\mathbf{x})}$ channels that we denote by $W^{(1||\mathbf{x})}$. Thus for each coordinate $i \in \left[\!\left[1, 2^{m-t-1}\right]\!\right]$ and symbol $s \in \mathbb{Z}_q$, we have the probability

$$\begin{aligned}
\Pi_i^{(1||\mathbf{x})}(s) &\overset{\triangle}{=} \mathbb{P}\left(v_i = s | y_i, y_{i+2^{m-t-1}}\right) && \text{(A.8)}\\
&= \sum_{s' \in \mathbb{Z}_q} \Pi_i^{(\mathbf{x})}(s+s') \cdot \Pi_{i+2^{m-t-1}}^{(\mathbf{x})}(\alpha(\mathbf{x}) \cdot s') && \text{(A.9)}\\
&= \sum_{s' \in \mathbb{Z}_q} \Pi_i^{(\mathbf{x})}(s-s') \cdot \Pi_{i+2^{m-t-1}}^{(\mathbf{x})}(-\alpha(\mathbf{x}) \cdot s') && \text{(A.10)}\\
&= \left(\Pi_i^{(\mathbf{x})} * \overline{\Pi}_{i+2^{m-t-1}}^{(\mathbf{x})}\right)(s) && \text{(A.11)}
\end{aligned}$$

where $\overline{\Pi}_i^{(\mathbf{x})}(s) \overset{\triangle}{=} \Pi_i^{(\mathbf{x})}(-\alpha(\mathbf{x}) \cdot s)$.

On another hand, let us assume that the decoding of $\mathbf{y}_\ell - \alpha(\mathbf{x})^{-1}\mathbf{y}_r$ has led us to the vector $\widetilde{\mathbf{v}}$ that we expect to be $\mathbf{v}$ (for the genie-aided decoder used for the construction of the code, we actually take $\widetilde{\mathbf{v}} = \mathbf{v}$, regardless of the result of the decoding of $\mathbf{y}_\ell - \alpha(\mathbf{x})^{-1}\mathbf{y}_r$). We now have two independent noisy versions of the same vector $\mathbf{u}$ that are $\alpha(\mathbf{x})^{-1}\mathbf{y}_r$ and $\mathbf{y}_\ell - \widetilde{\mathbf{v}}$. In other words, supposing $\widetilde{\mathbf{v}} = \mathbf{v}$, the vector $\mathbf{u}$ has been sent twice through the channel $W^{(\mathbf{x})}$; we denote by $W^{(0||\mathbf{x})}$ the resulting channel and for each coordinate $i \in \left[\!\left[1, 2^{m-t-1}\right]\!\right]$ and symbol $s \in \mathbb{Z}_q$, we have the probability

$$\begin{aligned}
\Pi_i^{(0||\mathbf{x})}(s) &\overset{\triangle}{=} \mathbb{P}\left(u_i = s | y_i, y_{i+2^{m-t-1}}\right) && \text{(A.12)}\\
&= \frac{1}{\eta} \cdot \Pi_i^{(\mathbf{x})}(s+\widetilde{v}_i) \cdot \Pi_{i+2^{m-t-1}}^{(\mathbf{x})}(\alpha(\mathbf{x}) \cdot s) && \text{(A.13)}
\end{aligned}$$

where $\eta \overset{\triangle}{=} \sum_{s' \in \mathbb{Z}_q} \Pi_i^{(\mathbf{x})}(s'+\widetilde{v}_i) \cdot \Pi_{i+2^{m-t-1}}^{(\mathbf{x})}(\alpha(\mathbf{x}) \cdot s')$ is a normalization factor.

Finally, for decoding a received word $\mathbf{y} \in \mathbb{Z}_q^{2^m}$ in the code U_ε that has been sent through a Gaussian channel of standard deviation σ, one essentially has to compute recursively the vector probabilities $\Pi_i^{(\mathbf{x})}$ for all $t \in [\![1, m]\!]$, $\mathbf{x} \in \{0,1\}^t$ and $i \in [\![1, 2^{m-t}]\!]$ using the Equations (A.11) and (A.13). Note that the initial channel $W^{(\varepsilon)}$ is the original Gaussian channel; so for all $i \in [\![1, 2^m]\!]$ and $s \in \mathbb{Z}_q$, we have

$$\Pi_i^{(\varepsilon)}(s) = \mathbb{P}\left(\mathcal{G}_{\mathbb{Z}_q,\sigma} = y_i - s\right) \tag{A.14}$$

where $\mathcal{G}_{\mathbb{Z}_q,\sigma}$ is the modular Gaussian distribution defined as follows:

Definition A.3 (Discrete Gaussian Distribution). *Let $\sigma > 0$ and let $\mathcal{S} \subset \mathbb{R}$ be a discrete set. The discrete Gaussian distribution $\mathcal{G}_{\mathcal{S},\sigma}$ over $\mathcal{S}$ is defined by:*

$$\mathbb{P}\left(\mathcal{G}_{\mathcal{S},\sigma} = x\right) \stackrel{\triangle}{=} \frac{\rho_\sigma(x)}{\sum_{y \in \mathcal{S}} \rho_\sigma(y)} \tag{A.15}$$

where $\rho_\sigma(x) \stackrel{\triangle}{=} \exp(-x^2/2\sigma^2)$ is the probability density function of the normal distribution $N(0, \sigma^2)$.

In particular, if $\mathcal{S} \stackrel{\triangle}{=} \mathbb{Z}_q$ then we speak of modular Gaussian distribution and for all $x \in \mathbb{Z}_q$, we have

$$\mathbb{P}\left(\mathcal{G}_{\mathbb{Z}_q,\sigma} = x\right) = \mathbb{P}\left(\mathcal{G}_{\mathbb{Z},\sigma} \in x + q\mathbb{Z}\right) = \frac{\sum_{u \in x+q\mathbb{Z}} \rho_\sigma(u)}{\sum_{y \in \mathbb{Z}} \rho_\sigma(y)} \tag{A.16}$$

where x is assimilated to any of its representatives.

When arriving to the codes on the leaves – that are the codes $U_{\mathbf{x}}$ such that $\mathbf{x} \in \{0,1\}^m$ – then we can exhaustively decode $U_{\mathbf{x}}$:

1. if $\mathbf{x} \in F$ (meaning the corresponding symbol is *frozen*) then the only possible codeword in $U_{\mathbf{x}}$ is the symbol 0,
2. if $\mathbf{x} \notin F$, then we choose the maximum likelihood codeword in $U_{\mathbf{x}}$ that is the symbol s for which $\mathit{\Pi}_1^{(\mathbf{x})}(s)$ is the greatest.

The running time of Successive Cancelation decoding is given by the following lemma:

Lemma A.4 (Complexity of the SC decoder). *Assuming q is a power of 2. The running time for decoding a word in a polar code of length 2^m and dimension k over $\mathbb{Z}_q$ is:*

$$T_{\text{SC}} \leq 3 \cdot \left(C_{\text{add}} \cdot N_{\mathsf{FFT}}^{(\text{add})}(q) + C_{\text{mul}} \cdot N_{\mathsf{FFT}}^{(\text{mul})}(q)\right) \cdot m \cdot 2^m \tag{A.17}$$

where C_{add} and C_{mul} are the costs of an addition and a multiplication, respectively, and $N_{\mathsf{FFT}}^{(\text{add})}(q)$ and $N_{\mathsf{FFT}}^{(\text{mul})}(q)$ are the number of additions and multiplications needed to achieve a discrete Fourier transform over $\mathbb{Z}_q$ (Proposition 2.11 provides those numbers for $q = 3329$, which were computed using the FFTW software [FJ05]).

Proof. For all $t \in [\![1, m]\!]$, $\mathbf{x} \in \{0,1\}^t$ and $i \in [\![1, 2^{m-t}]\!]$ – *i.e.* for $m \cdot 2^m$ triplets $(t, \mathbf{x}, i)$ – we can compute the vector of probabilities $\mathit{\Pi}_i^{(\mathbf{x})}$ with at most $3 \cdot q \cdot \log_2(q)$ multiplications. Indeed, we either have to compute Equation (A.11) or Equation (A.13). In the first case, it is a convolution; this can be done with the help of three fast Fourier transforms, with each FFT requiring $N_{\mathsf{FFT}}^{(\text{add})}(q)$ additions and $N_{\mathsf{FFT}}^{(\text{mul})}(q)$ multiplications. In the second case, we only have to do $2.q$ multiplications and q additions, which is less than the cost of a convolution.

Remark A.5. We could reduce the cost of the SC decoder by considering the vectors of LLR (*Log Likelihood Ratio*) instead of the vectors of probabilities. This trick allows to transform multiplications into additions.

List Decoding Using Probabilistic SC Decoder. We can modify the above SC decoder to obtain a probabilistic decoder. To this end, when decoding non-frozen symbols in the codes on the leaves $U_{\mathbf{x}}$ where $\mathbf{x} \in \{0,1\}^m \setminus F$, then output the symbol s according to the distribution $\mathit{\Pi}_1^{(\mathbf{x})}$ instead of returning the one with the best probability. Note that as m tends to infinity and $\frac{k}{2^m}$ remains constant, for $\mathbf{x} \in \{0,1\}^m \setminus F$, the channels $W^{(\mathbf{x})}$'s have capacity very close to 1 and for $\mathbf{x} \in F$, they have capacity very close to 1. Because of this polarization phenomenon, we can prove similarly to [KU10] that our probabilistic SC decoder achieves an average decoding distance

$$d = \sqrt{\tfrac{2^m}{2\pi e}} \cdot q^{1-\frac{k}{2^m}} \cdot (1 + o(1)) \tag{A.18}$$

when decoding a random vector in $\mathbb{Z}_q^{2^m}$.

To turn this probabilistic SC decoder into a list decoder, one only has to running it L times then choosing the codeword that minimizes the decoding distance. The complexity of a such algorithm is essentially L times the complexity of the SC decoder given by Lemma A.4. Note that, contrary to some more classical list decoders of polar codes, the L decoding procedures of our algorithm can be trivially parallelized.

Punctured Polar Code. The polar codes construction above is about codes of length that are a power of 2. In our case, we may require codes of other length. A simple way for reducing the length of a code without changing its dimension is to puncture it. Let n, k be two positive integers. We build a linear code of length n and dimension k by puncturing a polar code of length 2^m and dimension k where $m \triangleq \lceil \log_2(n) \rceil$. Let denote by $\ell \triangleq 2^m - n$ the number of symbol to puncture. The puncturing operation essentially consists of ignoring the ℓ first symbols of the codeword; that is equivalent to suppose that the ℓ first physical channels through which transit the codewords are of maximal entropy:

$$\Pi_i^{(\varepsilon)}(s) \triangleq \frac{1}{q} \qquad \forall i \in [\![1, \ell]\!] \tag{A.19}$$

Note that we made this assumption both for the decoder and also for the genie-aided decoder used to determine the frozen positions.

B An Intuition about the Distinguishability of the Good Guess.

An important insight from Approximation 4.6 is that the score $F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^{\top}\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)$ depends on the length enumerator of the vectors in the coset $q\Lambda\left(\mathbf{B}_{\mathsf{global}}\right)^{\vee}+\mathbf{r}_{\mathsf{proj}}$. Specifically, for the good guess $(\widetilde{\mathbf{s}_{\mathsf{enu}}},\widetilde{\mathbf{s}_{\mathsf{fft}}})=(\mathbf{s}_{\mathsf{enu}},\mathbf{s}_{\mathsf{fft}})$, this lattice coset contains a particularly short vector $\mathbf{r}_{\mathsf{proj}}=(\mathbf{P}(\mathbf{e},\mathbf{s}_{\mathsf{lat}}),\mathbf{s}_{\mathsf{fft}})$ where $\mathbf{P}\overset{\triangle}{=}\mathbf{B}'\left(\mathbf{B}'^{\top}\mathbf{B}'\right)^{-1}\mathbf{B}'^{\top}$ is the orthogonal projection onto span$(\mathbf{B}')$. In contrast, for wrong guesses where $(\widetilde{\mathbf{s}_{\mathsf{enu}}},\widetilde{\mathbf{s}_{\mathsf{fft}}})\neq(\mathbf{s}_{\mathsf{enu}},\mathbf{s}_{\mathsf{fft}})$, the shortest vector is no longer $\mathbf{r}_{\mathsf{proj}}$. This observation provides a preliminary answer to the indistinguishability question posed in [DP23b]. Indeed, we cannot distinguish the good guess from the wrong ones if the length of $(\mathbf{P}(\mathbf{e},\mathbf{s}_{\mathsf{lat}}),\mathbf{s}_{\mathsf{fft}})$ is greater than the length of the shortest vector in $q\Lambda\left(\mathbf{B}_{\mathsf{global}}\right)^{\vee}+\mathbf{r}_{\mathsf{proj}}$. Given that the coordinates of $(\mathbf{e},\mathbf{s}_{\mathsf{lat}},\mathbf{s}_{\mathsf{fft}})$ are i.i.d. random variables following a centered binomial distribution with parameter α, we estimate the length of $(\mathbf{P}(\mathbf{e},\mathbf{s}_{\mathsf{lat}}),\mathbf{s}_{\mathsf{fft}})$ to be

$$\|(\mathbf{P}(\mathbf{e},\mathbf{s}_{\mathsf{lat}}),\mathbf{s}_{\mathsf{fft}})\|\approx\sqrt{\tfrac{\alpha(\beta_{\mathsf{sieve}}+n_{\mathsf{fft}})}{2}}. \tag{B.1}$$

On the other hand, if $\Lambda\left(\mathbf{B}_{\mathsf{global}}\right)$ is treated as a random lattice with volume $V_{\mathsf{global}}\overset{\triangle}{=}\mathsf{Vol}\left(\Lambda(\mathbf{B}_{\mathsf{global}})\right)=\mathsf{Vol}\left(\Lambda(\mathbf{B}')\right)\cdot\mathsf{Vol}\left(\Lambda(\mathbf{B}_{\mathsf{lsc}})\right)$, then, using the Gaussian Heuristic, we can estimate the length of the shortest vector in the lattice coset to be

$$\lambda_1\left(q\Lambda\left(\mathbf{B}_{\mathsf{global}}\right)^{\vee}+\mathbf{r}_{\mathsf{proj}}\right)\approx\frac{q}{V_{\mathsf{global}}^{\frac{1}{\beta_{\mathsf{sieve}}+n_{\mathsf{fft}}}}}\cdot\sqrt{\tfrac{\beta_{\mathsf{sieve}}+n_{\mathsf{fft}}}{2\pi e}}. \tag{B.2}$$

However, we don't consider just one such coset, but rather $M\overset{\triangle}{=}R\cdot q^{k_{\mathsf{fft}}}$. Therefore, the probability of having an even smaller shortest vector in one of the cosets is not negligible. In [DP23b, Section 4.3], the shortest vector, across all cosets, is estimated to be

$$\approx\frac{\lambda_1\left(q\Lambda\left(\mathbf{B}_{\mathsf{global}}\right)^{\vee}+\mathbf{r}_{\mathsf{proj}}\right)}{M^{\frac{1}{\beta_{\mathsf{sieve}}+n_{\mathsf{fft}}}}}. \tag{B.3}$$

Table B.1 compares (B.1) and (B.3) for the parameters derived for Kyber in Section 5.

Scheme	C0		CC		CN	
	Eq. (B.1)	Eq.(B.3)	Eq. (B.1)	Eq.(B.3)	Eq. (B.1)	Eq.(B.3)
Kyber-512	25.42	26.61	25.66	27.21	25.57	26.58
Kyber-768	25.63	26.70	25.82	27.23	25.75	27.19
Kyber-1024	30.25	31.17	30.38	31.99	30.48	32.34

Table B.1: Comparison between the estimated length of $(\mathbf{P}(\mathbf{e},\mathbf{s}_{\mathsf{lat}}),\mathbf{s}_{\mathsf{fft}})$ and the estimated length of the shortest vector in all the cosets $q\Lambda\left(\mathbf{B}_{\mathsf{global}}\right)^{\vee}+\mathbf{r}_{\mathsf{proj}}$, based on the parameters provided in Table C.1 (Appendix C). We are outside the contradictory regime raised in [DP23b] as soon as *Eq.*(B.1) $<$ *Eq.*(B.3).

In Subsection 4.2, we refine these calculations to provide an accurate approximation of $F^{(\mathsf{lsc})}_{\widetilde{\mathbf{s}_{\mathsf{enu}}}}\left(\mathbf{G}^{\top}\widetilde{\mathbf{s}_{\mathsf{fft}}}\right)$, which we validate through simulations. In particular, we no longer assume that $\Lambda\left(\mathbf{B}_{\mathsf{global}}\right)$ is a random lattice of volume V_{global}; instead, we separately analyze the first β_{sieve} coordinates and the last n_{fft} coordinates.

C Parameters Tables Related to the Complexities Given In Table 5.1.

C0:

Scheme	m	β_{bkz}	β_{sieve}	n_{enu}	n_{fft}	k_{fft}	n_{lat}	d_{lat}	μ_{lsc}	σ_{lsc}	$\log_2(N)$	$\log_2(T)$
Kyber-512	488	393	393	1	38	8	473	2882.14	975.17	42.11	81.55	43.82
Kyber-768	667	588	588	6	69	12	693	4401.02	1741.80	48.28	122.02	64.69
Kyber-1024	920	815	815	9	100	17	915	5190.00	2134.08	45.25	169.13	88.47

CC:

Scheme	m	β_{bkz}	β_{sieve}	n_{enu}	n_{fft}	k_{fft}	n_{lat}	d_{lat}	μ_{lsc}	σ_{lsc}	$\log_2(N)$	$\log_2(T)$
Kyber-512	475	384	387	5	52	9	455	2706.86	1528.73	47.38	80.31	43.31
Kyber-768	636	581	574	6	93	14	669	4015.56	2410.56	46.47	119.12	63.01
Kyber-1024	802	811	792	10	131	19	883	4683.71	2956.65	50.05	164.35	85.86

CN:

Scheme	m	β_{bkz}	β_{sieve}	n_{enu}	n_{fft}	k_{fft}	n_{lat}	d_{lat}	μ_{lsc}	σ_{lsc}	$\log_2(N)$	$\log_2(T)$
Kyber-512	457	384	388	4	48	9	460	2830.06	1292.55	39.26	80.51	43.42
Kyber-768	682	583	577	8	86	13	674	4083.09	2309.88	57.72	119.74	63.50
Kyber-1024	782	816	797	6	132	19	886	4678.80	2997.51	46.56	165.39	86.37

Table C.1: Parameters to obtain Table 5.1.

C0:

Scheme	$\log_2(P_{\mathsf{wrong}})$	$\log_2(R)$	$\log_2(T_{\mathsf{sample}})$	$\log_2(N \cdot T_{\mathsf{dec}})$	$\log_2(T_{\mathsf{FFT}})$	η	$log_2(\varepsilon)$
Kyber-512	-103.25	2.84	115.76	118.95	112.13	0.91	-6.81
Kyber-768	-220.73	9.49	172.70	160.64	159.52	0.69	-70.83
Kyber-1024	-295.30	13.74	238.98	207.75	218.53	0.62	-82.65

CC:

Scheme	$\log_2(P_{\mathsf{wrong}})$	$\log_2(R)$	$\log_2(T_{\mathsf{sample}})$	$\log_2(N \cdot T_{\mathsf{dec}})$	$\log_2(T_{\mathsf{FFT}})$	η	$log_2(\varepsilon)$
Kyber-512	-119.57	9.39	139.51	117.71	124.00	0.66	-4.87
Kyber-768	-177.79	9.49	194.81	157.74	183.15	0.73	-4.49
Kyber-1024	-244.03	15.15	259.35	204.17	242.09	0.63	-6.57

CN:

Scheme	$\log_2(P_{\mathsf{wrong}})$	$\log_2(R)$	$\log_2(T_{\mathsf{sample}})$	$\log_2(N \cdot T_{\mathsf{dec}})$	$\log_2(T_{\mathsf{FFT}})$	η	$log_2(\varepsilon)$
Kyber-512	-120.51	7.71	143.30	117.91	124.00	0.71	-7.49
Kyber-768	-225.52	12.32	189.78	158.36	171.34	0.63	-61.09
Kyber-1024	-240.59	9.49	254.44	205.21	242.09	0.76	-8.78

Table C.2: Intermediate results for Table 5.1. We recall that $P_{\mathsf{good}} \approx 0.5$. η and ε are defined in Lemma 3.2.