Improved Estimation of Key Enumeration with Applications to Solving LWE

Alessandro Budroni and Erik Mårtensson
Cryptography Research Center, Technology Innovation Institute, Abu Dhabi, UAE
alessandro.budroni@tii.ae
Selmer Center, Department of Informatics, University of Bergen, Norway
Department of Electrical and Information Technology, Lund University, Sweden
erik.martensson@uib.no,eit.lth.se

Abstract—In post-quantum cryptography (PQC), Learning With Errors (LWE) is one of the dominant underlying mathematical problems. For example, in NIST’s PQC standardization process, the Key Encapsulation Mechanism (KEM) protocol chosen for standardization was Kyber, an LWE-based scheme. Recently, the dual attack surpassed the primal attack in terms of concrete complexity for solving the underlying LWE problem for multiple cryptographic schemes, including Kyber. The dual attack consists of a reduction part and a distinguishing part. When estimating the cost of the distinguishing part, one has to estimate the expected cost of enumerating over a certain number of positions of the secret key. Our contribution consists of giving a polynomial-time approach for calculating the expected complexity of such an enumeration procedure. This allows us to revise the complexity of the dual attack on the LWE-based protocols Kyber, Saber and TFHE. For all these schemes we improve upon the total bit-complexity in both the classical and the quantum setting.

As our method of calculating the expected cost of enumeration is fairly general, it might be of independent interest in other areas of cryptography or even in other research areas.

I. INTRODUCTION

Introduced by Regev in 2005 [1], the Learning With Errors Problem (LWE) is a computational problem that has been used as a building block for several quantum-resistant cryptographic primitives. A consistent number of schemes in each round of NIST’s Post-Quantum Standardization Process [2] base their security on the hardness of LWE. One of them is Kyber, which was chosen as the standard algorithm for encryption. Saber is another LWE-based scheme, which is very similar to Kyber and made it to the third round of the competition. It is also possible to build Fully Homomorphic Encryption (FHE) on LWE. TFHE is such an encryption scheme, based on [3].

Crytopanalysis of LWE is an active area of research that encompasses various techniques, including combinatorial methods like the Blum-Kalai-Wasserman (BKW) algorithm [4], algebraic methods [5], and lattice-reduction-based approaches, such as the primal attack [6] and the (recently) most successful dual attack [7]–[10]. Both BKW and the dual attack, in their most recent variants, include a subroutine consisting of enumerating a vector with entries from a non-uniform distribution. Previous works dealt with this problem either using unexplained models for estimating the cost of enumeration or using unnecessarily pessimistic upper limit formulas [9], [10].

The contribution of this manuscript is to provide a new and more accurate method to estimate the cost of such an enumeration procedure. Our key realization is that the frequencies of the different possible secret coefficient values follow a multinomial distribution, meaning that the number of unique probabilities for different possible keys is only polynomial in the number of positions we enumerate over. This allows us to precisely calculate the expected cost of key enumeration in polynomial time.

We integrate this new method into the complexity estimation of the dual attack and obtain new security estimates for the widely studied lattice-based schemes Kyber, Saber and TFHE, both for the classic and quantum case scenario. Furthermore, our contribution is general enough that it easily can be applied to any situation where enumeration over a vector sampled from a non-uniform distribution is needed.

The remaining part of the paper is organized as follows. In Section II, we present notations and necessary background. In Section III we introduce our new key enumeration approach, while in Section IV we apply it to some lattice-based protocols. Finally, in Section V we give the conclusions.

II. PRELIMINARIES

A. Notation

We denote the set of the integer, rational and real numbers with \( \mathbb{Z}, \mathbb{Q}, \mathbb{R} \) respectively. For a positive integer \( p \), we write \( \mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z} \). Upper case letters, e.g. \( M \), denote matrices, and bold lower case letters, e.g. \( v \), represent column vectors. We represent with \( v_j \) the \( j \)-th component of \( v \). We let \( \log(\cdot) \) denote the 2-logarithm. The notation \( \|v\| \) denotes the Euclidean norm of \( v \in \mathbb{R}^n \) defined as

\[ \|v\| = \sqrt{v_1^2 + \cdots + v_k^2}. \]

For a discrete distribution \( X \), its entropy is defined as

\[ H(X) := -E(\log(X)) = -\sum_k p(x_k) \cdot \log(p(x_k)). \quad (1) \]

B. Quantum Search Algorithms

Grover’s algorithm is a way of efficiently searching for elements in an unstructured set. Let \( S \) be a finite set of \( N \)
objects of which $t \leq N$ are targets. An oracle $O$ identifies
the targets if, for every $s \in S$, $O(s) = 1$ if $s$ is a target,
$O(s) = 0$ otherwise. Classically, one needs $O(N/t)$ queries
to the oracle to identify a target. Grover provided a quantum
algorithm that identifies a target with only $O(\sqrt{N/t})$ queries
to the oracle [11]. Amplitude amplification is a subsequent
work that generalizes Grover’s search algorithm [12].

C. Lattices and Reduction Algorithms

A lattice is a discrete additive subgroup of $\mathbb{R}^n$. Let $B = \{b_1, ..., b_m\} \in \mathbb{R}^n$ be a set of linearly independent vectors.
We define the lattice generated by $B$ as

$$\mathcal{L}(B) = \mathcal{L}(b_1, ..., b_m) = \left\{ v \in \mathbb{R}^n : v = \sum_{i=1}^{m} \alpha_i b_i, \alpha_i \in \mathbb{Z} \right\}.$$ 

Unless differently specified, we will consider full-rank lattices,
and therefore $n = m$.

Typically, lattice reduction algorithms such as LLL or
BKZ [13]–[15], take as input a basis $B$ of the lattice and return
another basis with short and nearly orthogonal vectors.
Lattice sieving consists of a class of algorithms, initiated with
the work of Ajtai et al. [16], to solve the Shortest Vector
Problem (SVP). These are usually used internally by BKZ
as an SVP oracle. They allow us to compute a large number
of short vectors and they have an estimated complexity of
as an $SVP$ oracle. They allow us to compute a large number
of short vectors and they have an estimated complexity of
as an $SVP$ oracle. They allow us to compute a large number
of short vectors and they have an estimated complexity of

$$\rho_{0,\sigma}(a) = \frac{\exp(-\pi a^2/2\sigma^2)}{\sum_{d \in \mathbb{Z}} \exp(-\pi d^2/2\sigma^2)},$$

where $\rho_{0,\sigma}(a)$ is the probability distribution function of the
Gaussian distribution $N(0, \sigma)$ with mean 0 and variance $\sigma^2$.

In practice, it is more common to use a centered Binomial
distribution $\mathcal{B}_n[p]$, which takes values in $[0, p]$ or even

$$\mathcal{B}_n[p]:=\{0,1\}^n \text{ with probability mass function}
\frac{p^n}{n!} x^n (1-x)^{n-x}.$$ 

This constant can be improved quantumly to 0.2653 using Grover’s
algorithm [18]. It was recently further improved to 0.2570 by
using more sophisticated quantum methods [19].

D. Learning With Errors and Gaussian Distributions

Definition 1: Let $n$ be a positive integer, $q$ a prime and
two probability distributions over $\mathbb{Z}$. Fix a secret vector
$s \in \mathbb{Z}^n$ whose entries are sampled according to $\chi_s$.
Denote by $A_{\chi_s}$ and $\sigma_{\chi_s}$ the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by sampling $\alpha \in \mathbb{Z}_q^n$ uniformly at random, sampling an error $e \in \mathbb{Z}$ from $\chi_e$ and returning

$$(a, z) = (\alpha, (\alpha \cdot s) + e \mod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$ 

The search Learning With Errors (LWE) problem is to find the secret vector $s$ given a fixed number of samples from $A_{\chi_s}$.

The decision Learning With Errors (LWE) problem is to distinguish between samples drawn from $A_{\chi_s}$ and
samples drawn uniformly from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

Consider $m$ LWE samples

$$(\alpha_1, z_1), (\alpha_2, z_2), ..., (\alpha_m, z_m) \leftarrow A_{\chi_s}.$$ 

Then, one can represent such an LWE instance in a matrix-vector form as

$$(A, z) = (A, As + e \mod q) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$$

where $A$ is an $m \times n$ matrix with rows $a_1^T, a_2^T, ..., a_m^T$, $z = (z_1, z_2, ..., z_m)$, and $e$ is the vector of errors $(e_1, e_2, ..., e_m)$.

In theory, one usually instantiates $\chi_s$ and $\chi_e$ as the discrete Gaussian distribution $D_{\mathbb{Z}, \sigma}$ defined as the probability distribution
that to each $a \in \mathbb{Z}$ assigns the probability

$$\frac{\exp(-\pi a^2/2\sigma^2)}{\sum_{d \in \mathbb{Z}} \exp(-\pi d^2/2\sigma^2)},$$

where $\rho_{0,\sigma}(a)$ is the probability distribution function of the
Gaussian distribution $N(0, \sigma)$ with mean 0 and variance $\sigma^2$.

In practice, it is more common to use a centered Binomial
distribution $\mathcal{B}_n[p]$, which takes values in $[0, p]$ or even

$$\mathcal{B}_n[p]:=\{0,1\}^n \text{ with probability mass function}
\frac{p^n}{n!} x^n (1-x)^{n-x}.$$ 

This constant can be improved quantumly to 0.2653 using Grover’s
algorithm [18]. It was recently further improved to 0.2570 by
using more sophisticated quantum methods [19].
The BKW algorithm can be seen as a variant of the dual attack where the reduction is performed using combinatorial methods instead of lattice reduction. For this reason, techniques and improvements developed for BKW on the distinguishing stage have been successfully applied to the dual attack too. More generally, the BKW algorithm has the disadvantage of requiring an exponential number of samples \((m \gg n)\) to perform reduction. On the other hand, one typically has more control over the outcome (i.e. reduced samples).

### III. Improved Estimation of Key Enumeration

Consider the problem of guessing the random value \(X\) sampled from a discrete probability distribution with mass function \(p_k := P(X = x_k)\). Without loss of generality, we assume it to be non-increasing (i.e. \(p_0 \geq p_1 \geq p_2 \geq \ldots\)). The optimal strategy is obviously to guess that \(X = x_0\), followed by guessing that \(X = x_1\), and so on. The expected number of guesses until the right value is found with this strategy is

\[
G(X) = \sum_i i \cdot p_i.
\]  

(4)

Massey showed in [29] that

\[
G(X) \geq \frac{1}{4} 2^H(X).
\]

He also showed why there is no such formula for upper limiting \(G(X)\) in terms of \(H(X)\).

Now consider a sample of \(n\) values, each one drawn independently from the same distribution with mass function \((p_0, p_1, \ldots, p_{r-1})\). When enumerating all the possible samples of \(s\) on these \(n\) positions, we want to do so in decreasing order of probability until we find the solution. Since the total number of outcomes is equal to \(r^n\), we cannot simply compute the probability of every single outcome, sort all the probabilities and then compute the expectation that way. However, we can use the fact that the frequencies of each possible secret value follow the multinomial distribution [30]. The number of outcomes where \(k_0\) values are equal to \(x_0\), \(k_1\) values are equal to \(x_1\) and so on until \(k_{r-1}\) values are equal to \(x_{r-1}\), \(\sum_{i=0}^{r-1} k_i = n\), is

\[
\binom{n}{k_0, \ldots, k_{r-1}} = \frac{n!}{k_0! k_1! \cdots k_{r-1}!}.
\]

(5)

Notice that all these outcomes have exactly the same probability of

\[
\prod_{i=0}^{r-1} p_i^{k_i}.
\]

(6)

The total number of unique probabilities is only

\[
\mu = \binom{n + r - 1}{n} = \frac{(n+r-1) \cdots (n+1)}{(r-1)!} = \frac{(n+r-1)!}{(r-1)! n!}.
\]

(7)

For a fixed number \(r\) this expression is \(O(n^{r-1})\). Thus, for a sparse distribution the number of unique probabilities is low enough to be computed and sorted efficiently (i.e. in polynomial time w.r.t. \(n\)).

Denote the unique probabilities by \(p'_0, p'_1, \ldots, p'_{\mu-1}\), such that \(p'_0 \geq p'_1 \geq \cdots \geq p'_{\mu-1}\). Let \(f_i\) denote the number of times \(p'_i\) occurs. Also let \(F_i = \sum_{j=0}^{f_i} f_i\). Now we can express the expected number of guesses to make until we find the right one from (4), as

\[
\sum_{i=0}^{\mu-1} p'_i \left( F_i + \sum_{j=1}^{f_i} j \right) = \sum_{i=0}^{\mu-1} p'_i \left( F_i + \frac{f_i(f_i + 1)}{2} \right).
\]

(8)

Since (8) has \(O(n^{r-1})\) terms and each term can be computed efficiently, the whole expression can be computed efficiently for small values of \(r\).

#### A. Quantum setting

Consider again random values sampled from a discrete probability with mass function \((p_0, p_1, p_2, \ldots)\). With a quantum computer, as shown in [31] using amplitude amplification as a tool, the expected number of guesses to find the right value is

\[
G_{qc}(X) = \sum_i \sqrt{i} \cdot p_i.
\]

(9)

Using the Cauchy-Schwartz inequality we have that

\[
G_{qc}(X) = \sum_i \sqrt{i} \cdot p_i \cdot \sqrt{p_i} \leq \left( \sum_i i \cdot p_i \right) \cdot \left( \sum_i p_i \right) = \sqrt{G(X)}.
\]

(10)

Here, our method for computing the estimated cost of the enumeration of (9) still applies, with a minor twist. In this setting (8) changes to

\[
\sum_{i=0}^{\mu-1} p'_i \left( \sum_{j=1}^{f_i} \sqrt{F_i + j} \right).
\]

(11)

We can rewrite \(\sum_{j=1}^{f_i} \sqrt{F_i + j} = \sum_{j=1}^{F_i} j \frac{1}{\sqrt{j}} - \sum_{j=1}^{F_i} \sqrt{j} j\).

Now, to compute (11) efficiently we only need to have an efficient and precise formula for computing \(f(n) = \sum_{i=1}^{n} \zeta(i)\). For \(n \leq 30\) we can pre-compute the expression. For \(n > 30\) using the Euler-Maclaurin formula [32], we can derive the function

\[
f(n) \approx \zeta(-0.5) + \frac{1}{2} n^{1.5} + \frac{2}{3} n^{2.5} + \frac{1}{24} n^{3.5} - \frac{1}{1920} n^{4.5} + \frac{1}{9216} n^{5.5} - \frac{1}{45360} n^{6.5}.
\]

(12)

where \(\zeta(\cdot)\) is the Riemann zeta function, which approximates the sum with a relative error that is smaller than or equal to machine epsilon.
B. Further optimizations

If for two outcomes \(x_1\) and \(x_2\) we have \(P(x_1) = P(x_2)\), then we can merge these terms to speed up the calculation of the enumeration.

Also, more generally, if throughout the enumeration we have two lists of values \([x_1, x_2, \ldots, x_k]\) and \([x'_1, x'_2, \ldots, x'_k]\) and \(P([x_1, x_2, \ldots, x_k]) = P([x'_1, x'_2, \ldots, x'_k])\), then we can also merge these two terms.

IV. APPLICATION TO LATTICE-BASED SCHEMES

In the Matzov version of the dual attack on \(LWE\), the \(n\) positions of the secret \(s\) are divided up into three parts, \(k_{\text{lat}}\), \(k_{\text{fin}}\) and \(k_{\text{enum}}\). The attack first performs lattice reduction on \(k_{\text{lat}}\) positions. In the second phase it enumerates, in decreasing order of probability, all possible secrets on \(k_{\text{enum}}\) positions. For each such secret it performs an FFT on \(k_{\text{fin}}\) positions and checks if it has found the correct solution. Rewriting [9, Theorem 5.1] asymptotically we get the following formula for the cost of the distinguishing part of the dual attack.

\[
O(G(\chi^{k_{\text{enum}}}) \cdot (D + p^{k_{\text{fin}}})) 
\]

where \(D\) is the number of samples needed to distinguish the secret and \(\chi^{k_{\text{enum}}}\) refers to the distribution of \(k_{\text{enum}}\) values sampled independently from the distribution \(\chi\). The fact that the cost is additive in \(D\) and \(p^{k_{\text{fin}}}\) means that it is best to keep these two terms of similar size. Quantumly however, the cost is proportional to the square root of the number of samples needed to distinguish the secret, the cost of enumeration and the cost of performing the FFT quantumly [10, (4)]. More concretely the cost is

\[
O \left( \sqrt{D} \cdot p^{k_{\text{fin}}/2} \cdot G_{\text{qc}}(\chi^{k_{\text{enum}}}) \cdot \text{poly}(\log(n)) \right) 
\]

The drastically reduced cost of distinguishing is the main source of the quantum improvement that [10] achieves compared to [9]. Notice the more than quadratic speed-up of \(G_{\text{qc}}(\chi^{k_{\text{enum}}})\) over \(G(\chi^{k_{\text{enum}}})\), as shown in (10). In practice this speed-up means that it is optimal for the schemes studied in this paper to do enumeration only and let \(k_{\text{fin}} = 0\).

In Matzov [9], it was assumed that the expected cost of enumerating over \(k_{\text{enum}}\) positions is \(2^{k_{\text{enum}}}/H(x)\), without any explanation. In [10], this problem was addressed. They developed an upper limit formula for the expected cost of enumerating over \(k_{\text{enum}}\) positions sampled from a Discrete Gaussian distribution with a specified standard deviation \(\sigma\). When estimating the expected cost of enumerating over the secret of an actual scheme, they simply approximated the secret distribution as a Discrete Gaussian with the same standard deviation, see Table III. In the quantum setting they developed a similar model.

Using the method detailed in Section III, in both the classical and quantum setting we can calculate the expected cost of enumeration numerically with arbitrarily good precision, to compare against the models of [9], [10]. Since all the schemes use sparse (and symmetric) distributions for the secret, our method is very efficient at computing the expectations.

A classical comparison is illustrated in Figure 1, for the expected cost of enumeration for Kyber512/FireSaber. The exhaustive cost is the obvious upper limit of guessing every possible key. Notice that while the Matzov numbers are a bit too optimistic, they are actually closer to the exact numbers than the Albrecht/Shen model is. Notice that the gaps between the different models increase with the dimension.

Figure 2 covers the quantum setting. Notice that there is a consistent gap between the expected cost according to the Albrecht/Shen model and the exact value, which increases very slowly with the number of dimensions.

Table I shows the state-of-the-art of solving the underlying \(LWE\) problem using the dual attack for the different schemes and models considered in [10]. We briefly summarize the models here. The models CC, CN and C0 are increasingly optimistic models for the cost of the dual attack on classical computers. GE19 refers to the most pessimistic quantum model from [33], QN and Q0 correspond to CN and C0, but with the classical lattice sieving of [17] replaced by the quantum lattice sieving of [19]. Finally, QN [10] and Q0 [10] refer to the works of [10], where quantum speed-ups of the FFT and the enumeration are applied. All the numbers are computed using the script from [10].

Table II shows the updated state-of-the-art. These are achieved by replacing Albrecht’s and Shen’s upper limit formulas for enumeration by the exact values, as described in Section III. For all schemes and all models we show improvements, but the magnitude of the improvements vary. Our largest improvements are for the TFHE schemes, where the secret follows a uniform distribution, meaning that a Discrete Gaussian is a particularly bad approximation.

Very recently, another preprint of an improved version of the dual attack of Matzov was published [34]. There they introduce a modified way of enumerating over the secret. Compared to the results from [10] they achieve comparable levels of improvements to us, in the classical setting. They enumerate over the secret in a different way, meaning that our improved estimate of the cost of enumeration does not apply in their setting. However, they do not provide a quantum version of their improved algorithm, the setting where our contribution is the most impactful.

### TABLE I

<table>
<thead>
<tr>
<th>Scheme</th>
<th>CC</th>
<th>CN</th>
<th>C0</th>
<th>GE19</th>
<th>QN</th>
<th>Q0</th>
<th>Q0 [10]</th>
<th>QN [10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kyber512</td>
<td>139.2</td>
<td>134.4</td>
<td>115.4</td>
<td>139.5</td>
<td>124.4</td>
<td>102.7</td>
<td>119.3</td>
<td>99.7</td>
</tr>
<tr>
<td>Kyber768</td>
<td>196.1</td>
<td>190.6</td>
<td>173.7</td>
<td>191.9</td>
<td>175.3</td>
<td>154.6</td>
<td>168.3</td>
<td>150.0</td>
</tr>
<tr>
<td>Kyber1024</td>
<td>262.4</td>
<td>256.1</td>
<td>241.8</td>
<td>252.0</td>
<td>234.5</td>
<td>215.0</td>
<td>225.6</td>
<td>208.4</td>
</tr>
<tr>
<td>LightSaber</td>
<td>138.5</td>
<td>133.1</td>
<td>113.7</td>
<td>138.4</td>
<td>122.7</td>
<td>101.1</td>
<td>118.9</td>
<td>98.9</td>
</tr>
<tr>
<td>Saber</td>
<td>201.4</td>
<td>195.9</td>
<td>179.2</td>
<td>196.2</td>
<td>179.9</td>
<td>159.4</td>
<td>173.8</td>
<td>155.0</td>
</tr>
<tr>
<td>FireSaber</td>
<td>263.5</td>
<td>258.2</td>
<td>243.8</td>
<td>253.1</td>
<td>235.9</td>
<td>216.7</td>
<td>228.1</td>
<td>210.8</td>
</tr>
<tr>
<td>TFHE630</td>
<td>118.2</td>
<td>113.3</td>
<td>93.0</td>
<td>120.2</td>
<td>105.2</td>
<td>83.0</td>
<td>100.8</td>
<td>80.7</td>
</tr>
<tr>
<td>TFHE1024</td>
<td>122.0</td>
<td>117.2</td>
<td>95.4</td>
<td>123.9</td>
<td>108.5</td>
<td>84.8</td>
<td>105.6</td>
<td>83.2</td>
</tr>
</tbody>
</table>

\(^2\text{The code for computing these numbers will be made available.}\)
TABLE II
Updated state-of-the-art.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>CC</th>
<th>CN</th>
<th>CO</th>
<th>GE19</th>
<th>QN</th>
<th>Q0</th>
<th>Q0 [10]</th>
<th>QN [10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kyber 512</td>
<td>128.7</td>
<td>133.8</td>
<td>115.0</td>
<td>139.1</td>
<td>125.6</td>
<td>102.4</td>
<td>118.0</td>
<td>98.4</td>
</tr>
<tr>
<td>Kyber 768</td>
<td>194.8</td>
<td>190.0</td>
<td>172.9</td>
<td>190.6</td>
<td>174.5</td>
<td>154.5</td>
<td>166.3</td>
<td>148.0</td>
</tr>
<tr>
<td>Kyber 1024</td>
<td>260.6</td>
<td>254.5</td>
<td>240.6</td>
<td>251.0</td>
<td>233.4</td>
<td>214.5</td>
<td>223.2</td>
<td>206.2</td>
</tr>
<tr>
<td>LightSaber</td>
<td>137.5</td>
<td>132.6</td>
<td>113.3</td>
<td>138.0</td>
<td>122.3</td>
<td>101.0</td>
<td>117.6</td>
<td>97.7</td>
</tr>
<tr>
<td>Saber</td>
<td>200.9</td>
<td>195.6</td>
<td>178.5</td>
<td>196.1</td>
<td>179.3</td>
<td>159.2</td>
<td>172.4</td>
<td>153.8</td>
</tr>
<tr>
<td>FireSaber</td>
<td>262.9</td>
<td>256.9</td>
<td>242.6</td>
<td>252.8</td>
<td>235.3</td>
<td>216.4</td>
<td>226.2</td>
<td>208.8</td>
</tr>
<tr>
<td>TFHE630</td>
<td>115.7</td>
<td>111.3</td>
<td>92.1</td>
<td>118.2</td>
<td>101.0</td>
<td>84.5</td>
<td>101.7</td>
<td>80.4</td>
</tr>
<tr>
<td>TFHE1024</td>
<td>120.4</td>
<td>115.6</td>
<td>94.8</td>
<td>122.8</td>
<td>101.7</td>
<td>80.4</td>
<td>101.7</td>
<td>80.4</td>
</tr>
</tbody>
</table>

TABLE III
The secret distribution and its standard deviation, for each scheme.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Distribution</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kyber512</td>
<td>B_3</td>
<td>√6/2</td>
</tr>
<tr>
<td>Kyber768</td>
<td>B_3</td>
<td>1</td>
</tr>
<tr>
<td>Kyber1024</td>
<td>B_3</td>
<td>√10/2</td>
</tr>
<tr>
<td>LightSaber</td>
<td>B_3</td>
<td>√8/2</td>
</tr>
<tr>
<td>Saber</td>
<td>B_3</td>
<td>√6/2</td>
</tr>
<tr>
<td>FireSaber</td>
<td>U{0, 1}</td>
<td>1/2</td>
</tr>
<tr>
<td>TFHE630</td>
<td>U{0, 1}</td>
<td>1/2</td>
</tr>
<tr>
<td>TFHE1024</td>
<td>U{0, 1}</td>
<td>1/2</td>
</tr>
</tbody>
</table>

A. Applications to BKW

As discussed in Section II-E, the techniques introduced in Section III apply to the BKW algorithm too. In the setting of [27], [28], the secret coefficients are discrete Gaussian with a relatively large standard deviation, taken from the distributions of the LWE Darmstadt Challenges [35]. The authors perform enumeration over all possible secret values within 3 standard deviations for each position. By instead enumerating over the secret coefficients in decreasing order of probability, one would see improvements similar to those of the dual attack.

V. CONCLUSIONS

The method presented in this paper improves over previous estimations for key-enumeration used in the literature. As a direct application, we used it to revise the state-of-the-art complexities for the dual attack against Kyber, Saber and TFHE. Future research directions include the application of this method to other areas in cryptanalysis where an enumeration of a non-uniform vector is required. Furthermore, thanks to its generality, the method might find application also in areas outside the context of cryptography.

Acknowledgments. We thank Qian Guo, Martin Albrecht and Yixin Shen for helpful discussions on the topic. We also thank the anonymous reviewers of ISIT23 for useful comments on the initial submission of this paper. Erik Mårtensson was supported by the project “Kvantesikker Kryptografi” from the National Security Authority of Norway.

REFERENCES


