Further Improvements of the Estimation of Key Enumeration with Applications to Solving LWE

Alessandro Budroni[1] and Erik Mårtensson[2,3*]

[1]Cryptography Research Center, Technology Innovation Institute,
Abu Dhabi, UAE.
[2]Selmer Center, Department of Informatics, University of Bergen,
Bergen, Norway.
[3]Department of Electrical and Information Technology, Lund University,
Lund, Sweden.

*Corresponding author(s). E-mail(s): erik.martensson@eit.lth.se;
Contributing authors: alessandro.budroni@tii.ae;

Abstract

In post-quantum cryptography, Learning With Errors (LWE) is one of the dominant underlying mathematical problems. The dual attack is one of the main strategies for solving the LWE problem, and it has recently gathered significant attention within the research community. A series of studies initially suggested that it might be more efficient than the other main strategy, the primal attack. Then, a subsequent work by Ducas and Pulles (Crypto'23) raised doubts on the estimated complexity of such an approach. 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 decrease the estimated cost of this procedure and, hence, of the whole attack both classically and quantumly. In addition, we explore different enumeration strategies to achieve some further improvements. Our work is independent from the questions raised by Ducas and Pulles, which do not concern the estimation of the enumeration procedure in the dual attack. As our method of calculating the expected cost of enumeration is fairly general, it might be of independent interest in other areas of cryptanalysis or even in other research areas.

Keywords: Cryptography, Lattice-based cryptography, Learning with Errors, Dual attacks.

1 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].

Cryptanalysis 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 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 [9], or using unnecessarily pessimistic upper limit formulas [10].

Contribution. In this manuscript we give 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. Furthermore, our contribution is general enough that it can easily be applied to any situation where enumeration over a vector sampled from a non-uniform distribution is needed.

Recent Related Work. Since publishing the conference version of this manuscript [11], we have seen two interesting developments.

Firstly, Ducas and Pulles published a paper [12], where they questioned many of the heuristics that recent complexity estimates of the dual attack in [8–10, 13] are based on. The likely conclusion here is that the estimates in these works are too optimistic and that the primal attack regains the status as the most efficient attack on cryptographically relevant LWE-based schemes. A provably working dual attack, without a corresponding time complexity analysis, was very recently introduced in [14] in a first attempt to address these issues. Further studying the details of the issues covered in [12] is important, but outside the scope of this work. Important to notice is that their paper does not affect the estimation of the cost of the enumeration block within the dual attack.

Secondly, Glaser, May and Nowakowski published a paper [15] extending the techniques introduced in the conference version of our paper [11]. Briefly, their idea is to enumerate over only the most likely keys and abort if the secret is not among them. At the cost of reducing the success probability to around 1/2, they decrease the cost of the enumeration significantly. They did not study the impact of this improvement on the dual attack. In regards to this, we show that their approach can be stretched much further. By making the success probability a lot lower we can reduce the expected time complexity of enumeration even more. We also generalize our cost estimations from the conference version to incorporate aborted enumeration into the dual attack. It turns out that due to the cost of having to re-run lattice reduction, aborted enumeration does not seem to improve the dual attack on LWE.

Organization. The remaining part of the paper is organized as follows. In Section 2, we present notations and necessary background. In Section 3 we introduce our new key enumeration approach, while in Section 4 we apply it to some lattice-based schemes. In Section 5 we study and slightly generalize the idea of aborted enumeration, and study its impact on the dual attack. Finally, in Section 6 we conclude the paper.

2 Preliminaries

2.1 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. $\boldsymbol{v}$, represent column vectors. We represent with v_j the j-th component of $\boldsymbol{v}$. We let $\log(\cdot)$ denote the 2-logarithm. The notation $\|\boldsymbol{v}\|$ denotes the Euclidean norm of $\boldsymbol{v} \in \mathbb{R}^n$ defined as

$$\|\boldsymbol{v}\| = \sqrt{v_1^2 + \cdots + v_n^2}.$$

For a discrete distribution X, its entropy is defined as

$$H(X) := -\mathbb{E}(\log(X)) = -\sum_k p(x_k) \cdot \log(p(x_k)). \tag{1}$$

2.2 Quantum Search Algorithms

Grover's algorithm is a way of efficiently searching for elements in an unstructured set. Let $\mathcal{S}$ be a finite set of N objects of which $t \leq N$ are *targets*. An oracle $\mathcal{O}$ identifies the targets if, for every $s \in \mathcal{S}$, $\mathcal{O}(s) = 1$ if s is a target, $\mathcal{O}(s) = 0$ otherwise. Classically, one needs $O(N/t)$ oracle queries to identify a target. Grover provided a quantum algorithm that identifies a target with only $O(\sqrt{N/t})$ queries to the oracle [16].

Amplitude amplification is a subsequent work that generalizes Grover's search algorithm [17]. Let us informally explain which classical and quantum search problems it allows us to speed up. Given a search algorithm with a success probability of p. The algorithm is either classical or quantum without a need for intermediate measurements. Naively the algorithm needs to be repeated on average $1/p$ times to find a solution. However, with amplitude amplification, this number is reduced to $\mathcal{O}(1/\sqrt{p})$.

2.3 Lattices and Reduction Algorithms

A **lattice** is a discrete additive subgroup of $\mathbb{R}^n$. Let $B = \{\boldsymbol{b}_1, ..., \boldsymbol{b}_m\} \in \mathbb{R}^n$ be a set of linearly independent vectors. We define the lattice generated by B as

$$\mathcal{L}(B) = \mathcal{L}(\boldsymbol{b}_1, ..., \boldsymbol{b}_m) = \left\{ \boldsymbol{v} \in \mathbb{R}^n : \boldsymbol{v} = \sum_{i=1}^{m} \alpha_i \mathbf{b}_i,\ \alpha_i \in \mathbb{Z} \right\}.$$

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

Typically, lattice reduction algorithms such as LLL or BKZ [18–20], 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. [21], 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 $2^{c\beta+o(\beta)}$, where β is the dimension of the lattice and c is a constant equal to 0.292 for classical computers [22]. This constant can be improved quantumly to 0.2653 using Grover's algorithm [23]. It was recently further improved to 0.2570 in [24] and 0.2563 in [25], using increasingly sophisticated quantum methods.

2.4 Learning With Errors and Gaussian Distributions

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

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

- The *search* Learning With Errors (LWE) problem is to find the secret vector $\boldsymbol{s}$ given a fixed number of samples from $\mathcal{A}_{\mathbf{s},\chi_e}$.
- The *decision* Learning With Errors (LWE) problem is to distinguish between samples drawn from $\mathcal{A}_{\mathbf{s},\chi_e}$ and samples drawn uniformly from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

Consider m LWE samples

$$(\mathbf{a}_1, z_1), (\mathbf{a}_2, z_2), \ldots, (\mathbf{a}_m, z_m) \leftarrow \mathcal{A}_{s,\chi}.$$

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

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

where A is an $m \times n$ matrix with rows $\mathbf{a}_1^T, \mathbf{a}_2^T, \ldots, \mathbf{a}_m^T$, $\mathbf{z} = (z_1, z_2, \ldots, z_m)^T$, and $\mathbf{e}$ is the vector of errors $(e_1, e_2, \ldots, e_m)^T$.

In theory, one usually instantiates χ_s and χ_e as the *discrete Gaussian distribution* on $\mathbb{Z}_q$ with mean 0 and variance σ^2 which is defined as follows. First, consider the discrete distribution over $\mathbb{Z}$, denoted with D_σ, as the probability distribution obtained by assigning a probability proportional to $\exp\{-x^2/(2\sigma^2)\}$ to each $x \in \mathbb{Z}$. Then, define the discrete Gaussian distribution χ over $\mathbb{Z}_q$ by folding D_σ and accumulating the values of the probability mass function over all integers in their corresponding residue class modulo q.

In practice, it is more common to use a centered Binomial distribution $\mathbf{B}_\eta$, which takes values in $[-\eta, \eta]$ or a uniform distribution $\mathcal{U}\{a, b\}$, which takes values in $[a, b]$.

Given an LWE problem instance, there exists a polynomial-time transformation [26, 27] that makes the secret vector follow the same distribution as the error's distribution χ_e.

2.5 Distinguishing Attacks Against LWE

Dual Attack. The first attack on LWE performed on the so-called *dual* lattice was introduced in [7]. While the earlier versions of this attack were efficient only for instances with very small coefficients (e.g. $\mathbf{s} \in \{-1, 0, 1\}^n$), thanks to some recent contributions [8–10, 13], the attack now applies also to secrets with not-so-small coefficients.

Let $(A, \mathbf{b} = A\mathbf{s} + \mathbf{e} \mod q)$ be an $m \times n$ LWE instance, for $m \geq n$ where the secret $\mathbf{s}$ and the error $\mathbf{e}$ have been sampled from a discrete normal distribution with mean zero and standard deviations σ_s and σ_e respectively. Partition the matrix A as $(A_1 \parallel A_2)$ and, in correspondence, the secret $\mathbf{s}$ as $(\mathbf{s}_1 \parallel \mathbf{s}_2)$. Consider the following pair

$$(A_2, \mathbf{b} - A_1\tilde{\mathbf{s}}_1 \mod q). \tag{2}$$

For $\tilde{\mathbf{s}}_1 = \mathbf{s}_1$ we have that

$$\mathbf{b} - A_1\tilde{\mathbf{s}}_1 = A_2\mathbf{s}_2 + \mathbf{e} \mod q$$

and therefore (2) is a new LWE instance with reduced dimension. If $\tilde{\mathbf{s}}_1 \neq \mathbf{s}_1$, then (2) is uniform.

By enumerating over all possible vectors $\tilde{\mathbf{s}}_1$ of $\mathbf{s}_1$, one can distinguish the right guess as follows. Let $\mathcal{R}$ be an algorithm (e.g. BKZ, lattice sieving) that returns pairs $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}^{m \times n}$ such that $\mathbf{y}^T = (\mathbf{y}_1 \parallel \mathbf{y}_2)^T = \mathbf{x}^T A \mod q$, and $\mathbf{x}$ and $\mathbf{y}_2$ are *short*. Then, for $\tilde{\mathbf{s}}_1 = \mathbf{s}_1$, we have that

$$\mathbf{x}^T(\mathbf{b} - A_1\mathbf{s}_1) = \mathbf{x}^T(A_2\mathbf{s}_2 + \mathbf{e}) = \mathbf{y}_2^T\mathbf{s}_2 + \mathbf{x}^T\mathbf{e}. \tag{3}$$

This quantity is distributed approximately according to a discrete normal distribution with mean zero and variance $\|\mathbf{x}\|^2\sigma_s^2 + \|\mathbf{y}_2\|^2\sigma_e^2$. The choice of reduction algorithm $\mathcal{R}$ determines the expected length of the vectors $\mathbf{x}$ and $\mathbf{y}_2$, and therefore, the ability to distinguish (3) from uniformly random.

BKW Algorithm. In its original development, the Blum-Kalai-Wasserman (BKW) algorithm was proposed as a subexponential algorithm for solving the Learning Parity with Noise (LPN) problem [28]. Later, it has been applied to LWE [4], and further developed with new ideas such as Lazy Modulus Switching, Coded BKW, Coded BKW with Sieving and smooth Lazy Modulus Switching [29–34].

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 when compared to lattice reduction techniques. On the other hand, BKW allows tuning parameters in a way that offers a higher control on the magnitude distribution of the resulting reduced vectors.

3 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. $p_0 \geq p_1 \geq p_2 \geq \dots$). 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

$$G(X) = \sum_i i \cdot p_i. \tag{4}$$

$G(X)$ is called the guessing entropy of X. Massey showed in [35] 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 $(p_0, \dots, p_{r-1})$. When enumerating all the possible values of $\mathbf{s}$ 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, simply computing the probability of every single outcome, sorting all the probabilities and then computing the expectation directly according to (4), is inefficient. However, we can use the fact that the frequencies of each possible secret value follow the multinomial distribution [36]. 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}, where $\sum_{i=0}^{r-1} k_i = n$, is

$$\binom{n}{k_0, \dots, k_{r-1}} = \frac{n!}{k_0! k_1! \cdots k_{r-1}!}. \tag{5}$$

Notice that all these outcomes have exactly the same probability of

$$\prod_{l=0}^{r-1} p_l^{k_l}. \tag{6}$$

The total number of unique probabilities is only

$$\mu = \binom{n+r-1}{n} = \frac{(n+r-1)\cdots(n+1)}{(r-1)!} = \frac{(n+r-1)!}{(r-1)!n!}. \tag{7}$$

For a fixed number r this expression is $\mathcal{O}(n^{r-1})$. 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 $p'_0, p'_1, \dots, p'_{\mu-1}$, such that $p'_0 \geq p'_1 \geq \dots \geq p'_{\mu-1}$. Let f_i denote the number of times p'_i occurs. Also let $F_i = \sum_{j=0}^{i-1} f_j$. Now we can express the expected number of guesses to make until we find the right one from (4), as

$$\sum_{i=0}^{\mu-1} p'_i \left(F_i + \sum_{j=1}^{f_i} j \right) = \sum_{i=0}^{\mu-1} p'_i \left(F_i + \frac{f_i(f_i+1)}{2} \right). \tag{8}$$

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

3.1 Quantum Setting

Consider again random values sampled from a discrete probability with probability mass function $(p_0, \dots, p_{r-1})$. With a quantum computer, the most obvious approach is to use Grover search over the entire sample space. However, employing Montanaro's algorithm [37] gives better results. On a high level, this consists of performing Grover search over a sequence of sub-intervals of increasing length, until the target value is found. The expected number of guesses using Montanaro's algorithm to find the right key is

$$G_{\mathrm{qc}}(X) = \sum_i \sqrt{i} \cdot p_i. \tag{9}$$

Using the Cauchy-Schwartz inequality we have that

$$\begin{aligned} G_{\mathrm{qc}}(X) &= \sum_i \sqrt{i \cdot p_i} \cdot \sqrt{p_i} \leq \sqrt{\sum_i i \cdot p_i \cdot \sum_i p_i} \\ &= \sqrt{\sum_i i \cdot p_i} = \sqrt{G(X)}. \end{aligned} \tag{10}$$

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

$$\sum_{i=0}^{\mu-1} p'_i \left(\sum_{j=1}^{f_i} \sqrt{F_i + j} \right). \tag{11}$$

We can rewrite $\sum_{j=1}^{f_i} \sqrt{F_i + j} = \sum_{j=1}^{F_i+f_i} \sqrt{j} - \sum_{j=1}^{F_i} \sqrt{j}$. Now, to compute (11) efficiently we only need to have an efficient and precise formula for computing $f(n) = \sum_{i=1}^{n} \sqrt{i}$. For $n \leq 30$ we can pre-compute the expression. For $n > 30$ using the Euler-Maclaurin formula [38], we can derive the function

$$f(n) \approx \zeta(-0.5) + \frac{1}{2} n^{\frac{1}{2}} + \frac{2}{3} n^{\frac{3}{2}} + \frac{1}{24} n^{-\frac{1}{2}} - \frac{1}{1920} n^{-\frac{5}{2}} + \frac{1}{9216} n^{-\frac{9}{2}}, \tag{12}$$

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.

3.2 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. The most obvious example of this is a symmetric distribution, where $P(x_i) = P(-x_i)$, for all x_i.

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.

3.3 Related Work on Guessing Entropy

Guessing entropy has been studied in subsequent works after the initial paper by Massey [35], but generally in different settings and with different focus than ours. In [39] guessing entropy was studied in the context of side-channel attacks on for example AES. Unfortunately our method does not apply in their setting. Also, the authors only give lower limit formulas, whereas we are more interested in either upper limit formulas or precise estimates. Finally, the authors do not study the guessing entropy of quantum algorithms.

Recently, in [40] guessing entropy was extensively studied, with the quantum setting of (9) corresponding to setting $\rho = 0.5$ in Section 5D. However, also in this paper there is no upper limit formulas or methods to calculate the guessing entropy exactly.

4 Application to Lattice-based Schemes

In the Matzov version of the dual attack on LWE, the n positions of the secret $\mathbf{s}$ are divided up into three parts, k_{lat}, k_{fft} and k_{enum}. The attack first performs lattice reduction on k_{lat} positions. In the second phase it enumerates, in decreasing order of probability, all possible secrets on k_{enum} positions. For each such secret it performs an FFT on k_{fft} 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.

$$\mathcal{O}\left(G(\chi^{k_{\text{enum}}}) \cdot (D + p^{k_{\text{fft}}})\right), \tag{13}$$

where D is the number of samples needed to distinguish the secret and $\chi^{k_{\text{enum}}}$ refers to the distribution of k_{enum} values sampled independently from the distribution χ. The fact that the cost is additive in D and $p^{k_{\text{fft}}}$ 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

$$\mathcal{O}\left(\sqrt{D} \cdot p^{k_{\text{fft}}/2} \cdot G_{\text{qc}}(\chi^{k_{\text{enum}}}) \cdot \text{poly}(\log(n))\right). \tag{14}$$

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 $k_{\text{fft}} = 0$.

In Matzov [9], it was assumed that the expected cost of enumerating over k_{enum} positions is $2^{k_{\text{enum}} \cdot H(\chi)}$, without any explanation. In [10], this problem was addressed. They developed an upper limit formula for the expected cost of enumerating over k_{enum} positions sampled from a Discrete Gaussian distribution with a specified standard deviation σ. 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, according to Table 3. In the quantum setting they developed a similar model.

Using the method detailed in Section 3, 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/uniform) 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 1 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 [41]. QN and Q0 correspond to CN and C0, but with the classical lattice sieving of [22] replaced by the quantum lattice sieving of [24]. 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 2 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 3[1]. 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.

Actually make the code available

Recently, another preprint of an improved version of the dual attack of Matzov was published [13]. There they introduce a modified way of enumerating over the secret. Compared to the results from [10] they achieve comparable levels of improvements to

[1] See https://github.com/ErikMaartensson/ImprovedKeyEnumeration for the source code.

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, which is the setting where our contribution has the largest impact.

Given the recent work by Ducas and Pulles [12], the complexity numbers of Tables 1 and 2 should only be viewed as lower limits of the costs of the dual attack. However, we do still believe that they give a good estimate of the impact of our new estimations on the enumeration part of the dual attack. We leave studying the more detailed impact of [12] on the dual attack for future work.

Table 1 Previous state-of-the-art bit complexities for breaking cryptographic schemes using the dual attack.

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

Table 2 Updated state-of-the-art bit complexities for breaking cryptographic schemes using the dual attack.

Scheme	CC	CN	C0	GE19	QN	Q0	Q0[10]	QN[10]
Kyber 512	138.7	133.8	115.0	139.1	123.6	102.4	118.0	98.4
Kyber 768	194.8	190.0	172.9	190.6	174.5	154.5	166.3	148.0
Kyber 1024	260.6	254.5	240.6	251.0	233.4	214.5	223.2	206.2
LightSaber	137.5	132.6	113.3	138.0	122.3	101.0	117.6	97.7
Saber	200.9	195.6	178.5	196.1	179.3	159.2	172.4	153.8
FireSaber	262.9	256.9	242.6	252.8	235.3	216.4	226.2	208.8
TFHE630	115.7	111.3	92.1	118.2	103.9	82.8	95.6	76.8
TFHE1024	120.4	115.6	94.8	122.8	107.7	84.5	101.7	80.4

4.1 Applications to BKW

As discussed in Section 2.5, the techniques introduced in Section 3 apply to the BKW algorithm too. In the setting of [33, 34], the secret coefficients are discrete Gaussian with a relatively large standard deviation, taken from the distributions of the LWE Darmstadt Challenges [42]. 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.

Table 3 The secret distribution and its standard deviation, for each scheme.

Scheme	Distribution	Standard deviation
Kyber512	$\mathbf{B}_3$	$\sqrt{6}/2$
Kyber768	$\mathbf{B}_2$	1
Kyber1024	$\mathbf{B}_2$	1
LightSaber	$\mathbf{B}_5$	$\sqrt{10}/2$
Saber	$\mathbf{B}_4$	$\sqrt{2}$
FireSaber	$\mathbf{B}_3$	$\sqrt{6}/2$
TFHE630	$\mathcal{U}\{0,1\}$	$1/2$
TFHE1024	$\mathcal{U}\{0,1\}$	$1/2$

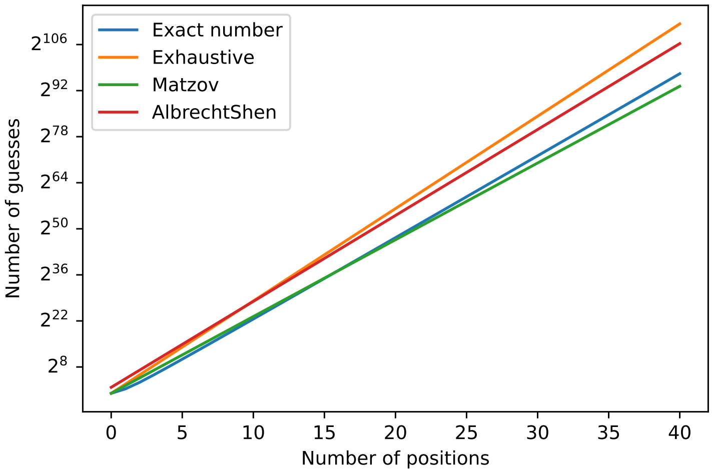

Fig. 1 The expected cost of enumeration in the classic setting for Kyber512/FireSaber.

5 Aborted Enumeration

In [15] the authors studied the expected cost of aborted key enumeration. The idea is to abort the search for the key once we have concluded that none of the most probable keys are equal to the secret key. Let us state their finding slightly more precisely.

The authors enumerate over all n-dimensional keys sampled independently from a non-uniform[2], finite distribution X, according to the procedure described in Section 3. If the secret key is not found after trying all keys with probabilities larger than or equal to $2^{-H(X)n}$, then they abort the search. Let μ' be the index such that $p_{\mu'} \geq 2^{-H(X)n}$ and $p_{\mu'+1} < 2^{-H(X)n}$.

Clearly the maximum number of secret keys to enumerate over is upper limited by $2^{H(X)n}$.[3] The logarithm of this expression is in turn equal to the entropy of the secret key. While the expression is still exponential in n, just like the case for full

[2] For a uniform distribution all keys are equally likely, making it pointless to abort the enumeration early.
[3] The upper limit is reached if and only if X comes from the uniform distribution.

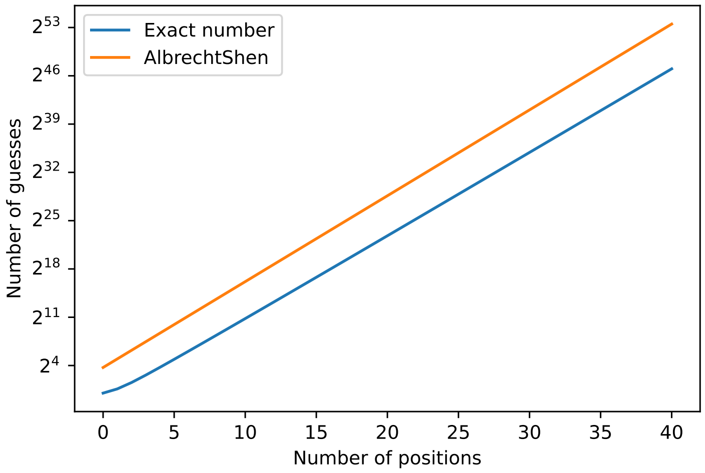

Fig. 2 The expected cost of enumeration in the quantum setting for Kyber512/FireSaber.

enumeration, the coefficient $H(X)$ is smaller than the corresponding coefficient for full enumeration. The authors of [15] show that the success probability of this aborted enumeration procedure is roughly 1/2. Thus, they limit the cost of enumeration in terms of the entropy of the secret[4].

In case enumeration fails to find the secret among the most probable keys, then we have two options.

1. Either we accept that there is a risk of failure.
2. Or we restart the enumeration with a new sample. The details of how this works depends on the context and will be discussed later in this section.

Let us generalize the setting from [15] a bit. Just like in Section 3, we are guessing a random value X sampled from a known probability distribution. Now, we add the option of re-sampling. At any point, we are allowed to discard the current value and sample a new one from the same probability distribution. For now, we assume that the cost of re-sampling is 0, but in certain settings it will be expensive. We will discuss more details below. The expected cost of performing one iteration of enumeration is

$$\sum_{i=0}^{\mu'} p'_i \left(F_i + \frac{f_i(f_i+1)}{2} \right) + \left(1 - \sum_{i=0}^{\mu'} f_i p'_i \right) F_{\mu'+1}. \tag{15}$$

Here, the last term corresponds to the fact that if the secret is not among the most probable keys, which happens with the probability $1 - \sum_{i=0}^{\mu'} f_i p'_i$, then we need to enumerate over all the $F_{\mu'+1}$ most probable keys to find this out. Now, the expected cost of enumeration until we find the secret key is

[4] This does not contradict the original result by Massey [35], saying that we cannot limit the guessing entropy in terms of the entropy of the distribution. Firstly, [15] deals with distributions of special shapes only. Secondly, Massey's original result did not take aborted enumeration into consideration.

$$\frac{\sum_{i=0}^{\mu'} p'_i \left(F_i + \frac{f_i(f_i+1)}{2}\right) + \left(1 - \sum_{i=0}^{\mu'} f_i p'_i\right) F_{\mu'+1}}{\sum_{i=0}^{\mu'} f_i p'_i}. \tag{16}$$

The idea of quantum enumeration can also be improved using aborted enumeration. Here we have two possible algorithms to consider.

Montanaro's Algorithm with Abortion

A first option is an aborted version of Montanaro's algorithm. Here we simply apply Montanaro's algorithm on the most likely keys only. If we fail to find the key, then we re-sample the secret and try again. The expected cost of it is

$$\frac{\sum_{i=0}^{\mu'} p'_i \left(\sum_{j=1}^{f_i} \sqrt{F_i + j}\right) + \left(1 - \sum_{i=0}^{\mu'} f_i p'_i\right) \sqrt{F_{\mu'+1}}}{\sum_{i=0}^{\mu'} f_i p'_i}. \tag{17}$$

Just like in the setting with full enumeration, the difference between the classical formula of (16) and the quantum formula is that we apply square roots to the F_i terms.

Grover's Algorithm with Abortion

In [15], the authors suggested replacing Montanaro's algorithm with abortion, with simply performing Grover's algorithm over the most likely keys. One iteration of this type of enumeration then costs

$$\sqrt{F_{\mu'+1}}. \tag{18}$$

Since Grover does not take the structure of the distribution into consideration, its cost is independent of the probability distribution[5]. The success probability of one iteration of aborted Grover is $\sum_{i=0}^{\mu'} f_i p'_i$. Grover's algorithm does not require any intermediate measurements. Thus, if we can get re-sampling for free, then we get a cost of

$$\frac{\sqrt{F_{\mu'+1}}}{\sqrt{\sum_{i=0}^{\mu'} f_i p'_i}} = \sqrt{\frac{F_{\mu'+1}}{\sum_{i=0}^{\mu'} f_i p'_i}}, \tag{19}$$

for aborted Grover using amplitude amplification [17]. Since Montanaro's algorithm uses intermediate measurements, we cannot get the corresponding speed-up for aborted Montanaro.

5.1 An Illustration of the Cost of Aborted Enumeration

The suggestion of aborting once the success probability per key is less than $2^{-H(X)n}$, leading to a total success probability of around 1/2, is of course arbitrary. It is indeed chosen, by design, to show that aborted enumeration can achieve an expected complexity upper limited by $2^{H(X)n}$.

We can generalize the idea to enumerating over the most likely keys and aborting when the total success probability is equal to whatever success probability p that

[5] Except that $F_{\mu'+1}$ does depend on the probability distribution.

we want. The formulas in (16)-(19) are unchanged, except that μ' is now the largest positive integer such that $\sum_{i=0}^{\mu'} f_i p'_i \leq p$.

In Figure 3 we compare the classical aborted enumeration algorithm against the two aborted quantum algorithms. We enumerate over 30 positions of secrets sampled from a centered Binomial distribution $\mathbf{B}_2$, which corresponds to the secret entries of Kyber768 and Kyber1024. We plot the time complexity against the success probability. The key assumption in this figure is that the cost of re-sampling is 0.

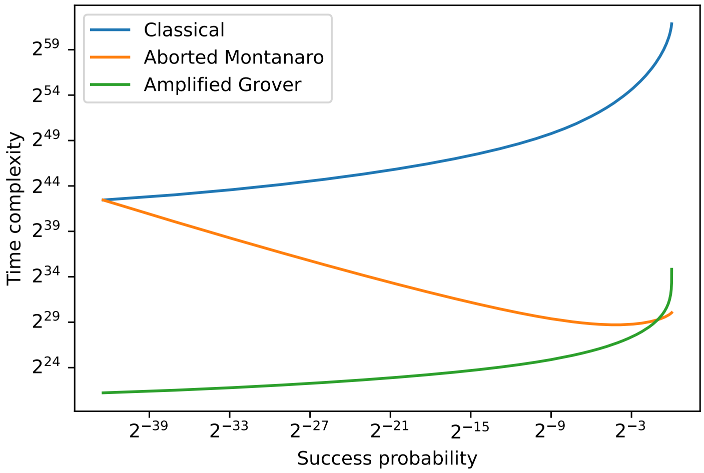

Fig. 3 The expected cost of aborted enumeration.

The key realization of [15] is that by reducing the success probability of aborted classical enumeration to around 1/2, the overall computational cost decreases dramatically. This principle can be extended much further. By stretching the enumeration process all the way to guessing the all-zeros vector and re-sampling in case of failure, we get the lowest possible time complexity. Also notice that the time complexity - for the success probability around 1/2 - already is around $2^{H(X)n}$. Thus, we can clearly go way below the this entropy limit.

For amplified Grover, we get the same pattern as for classical enumeration, except that the absolute complexities are much lower. Looking at (19), we see that the speed-up compared to classical enumeration is at best a square root, since the cost corresponds to the square root of enumerating over all the most likely keys without taking advantage of the structure of the probability distribution.

We see that aborted Montanaro is best for the highest success probabilities, but it quickly starts to perform worse the lower the success probability is. The reason - looking at (17) - is that aborted Montanaro cannot be improved with amplitude amplification. This means that we do not get a square root speed-up in the denominator. When we only enumerate over the single most likely value (the all zeros vector), then aborted Montanaro breaks down to performing Grover's algorithm over a single position and re-sampling in case of failure. This is of course equivalent to classical aborted enumeration over the zeros only vector.

5.2 Some Settings with Aborted Enumeration

So far, this section has assumed that re-sampling can be done as many times as you want and at no cost. Whether this is reasonable depends on the context.

In the context of cracking passwords, this is typically reasonable. Given a large set of users and the task of cracking the password an arbitrary user, re-sampling corresponds to start guessing another user's password. The task is achieved much easier by trying a small number of very common passwords for each user, rather than by brute-forcing for a single user's password.

In the context of using lattice enumeration[6] as an SVP oracle, pruning of the search tree is applied to speed up the enumeration. Pruning here corresponds to aborting. While pruning creates a risk that the enumeration fails, it compensates by lowering the enumeration cost. Taking advantage of the low cost of re-sampling in this setting, it was showed in [43] that by doing extreme pruning - even though each iteration of enumeration had a very low success probability - the reduced cost was so drastic that a significant improvement in performance was achieved.

5.3 Implications of Aborted Key Enumeration on Dual Attacks

For us, the most interesting setting is aborted enumeration as a subroutine for dual attacks on LWE. Notice that if going from full enumeration to aborted enumeration, in case the enumeration fails, then we need to re-sample somehow. This can be achieved by performing the enumeration part of the dual attack on another subset of the secret key entries.

As enumeration is only performed on a small subset of the entire key, this approach allows us to re-sample quite a few times, but there is of course a clear limit. Pushing aborted enumeration as far as in Figure 3 and only guessing that the secret is the all zeros vector fails miserably in this context for two reasons.

1. We can only re-sample a very limited number of times.
2. The cost of re-sampling is way too high, due to having to perform lattice reduction again for each failed enumeration.

The dual attack with full enumeration has a cost of

$$T_{\text{red}} + T_{\text{guess}},$$

where T_{red} is the cost of lattice reduction and T_{guess} is the cost of the guessing procedure. Now, if we do aborted enumeration, then this expression changes to

$$\frac{T'_{\text{red}} + T'_{\text{guess}}}{p},$$

[6] Notice that the word enumeration has a different meaning in this paragraph compared to the rest of the paper.

where T'_{red} is the cost of lattice reduction, T'_{guess} is the cost of the guessing procedure and p is the success probability of the enumeration. Here, on one hand, the cheaper cost of enumeration means that the algorithm will enumerate over slightly more positions and do lattice reduction on slightly fewer positions, mean that $T'_{\text{red}} < T_{\text{red}}$. On the other hand, since the success probability $p < 1$, to find the secret means that we might need to re-run lattice reduction. Exactly how these two changes affects the overall cost is non-trivial. We use a slightly modified version of the script by [10] to optimize the cost of the dual attack when using aborted enumeration to find the more precise estimate of this effect[7].

See Table 4 for complexity numbers for the dual attack with aborted enumeration, with a success probability of 50 %. Here we leave out the TFHE schemes, as these have secret entries sampled from a uniform distribution, making aborted enumeration pointless.

Note that due to the recent work by Ducas and Pulles [12], just like in Tables 1 and 2, the complexity numbers in Table 4 should be seen as optimistic lower limits. However, the difference between Table 4 and table 2 should still give a good comparison between the full and aborted enumeration subroutine within the dual attack.

Comparing Table 4 to Table 2, for some schemes and settings, the bit complexity is marginally better, while for other schemes and settings it is marginally worse. However, in all cases the difference is very modest. We tried using other success probabilities, also with results very marginally different from using full enumeration.

Lattice reduction on a certain number of positions is much cheaper than enumeration on the same number of positions (we do both only because the costs of lattice reduction and guessing are additive). Enumerating a few more positions means that we get to do lattice reduction on a few less positions. The problem with trying to reduce the guessing cost by lowering the success probability of aborted enumeration is that the cost of the risk of having to re-run lattice reduction roughly neutralizes the gain.

Classically, the problem is that we can only afford enumerating over a fairly small number of positions. The gains of being able to enumerate over a few more positions get canceled out by having to re-run lattice reduction.

Quantumly, full enumeration using Montanaro's algorithm is so cheap that it is optimal to skip the FFT part and focus on enumeration only. The cost of quantum enumeration is less than the square root of the cost of classical enumeration, as shown in (10). The problem is that when doing aborted enumeration, the factor $1/p$ means that aborted Montanaro benefits only modestly from a reduced success probability. At a fairly high success probability, aborted Montanaro even increases in time complexity when further lowering the success probability, as illustrated in Figure 3. Aborted Grover also does not work, as it performs worse than aborted Montanaro for the success probabilities relevant for the dual attack.

5.3.1 Limiting the Number of Hypotheses

A potential improvement of using aborted enumeration - not covered in the estimation of Table 4 - is the benefit of using fewer hypotheses. The lower the success probability we choose, the fewer hypotheses we make. Now let us assume that the secret, with

[7] See https://github.com/ErikMaartensson/ImprovedKeyEnumeration.

Table 4 Bit complexities of breaking cryptographic schemes using the dual attack with aborted enumeration.

Scheme	CC	CN	C0	GE19	QN	Q0	Q0[10]	QN[10]
Kyber 512	139.2	134.5	115.8	139.8	124.4	103.4	118.2	99.0
Kyber 768	195.5	190.1	173.7	190.9	175.4	155.3	166.3	148.2
Kyber 1024	260.6	255.3	240.6	250.6	234.2	215.4	222.7	205.9
LightSaber	138.0	133.0	114.3	138.6	123.2	102.0	117.7	98.0
Saber	201.0	196.0	179.4	196.5	180.2	160.0	172.4	153.8
FireSaber	262.8	257.6	243.1	252.8	236.3	216.8	225.9	208.5

respect to the positions we apply enumeration on, is one of the most likely ones (in other words, we do not miss it due to aborting early). Then the correct hypothesis is competing against a smaller set of incorrect hypotheses, which makes choosing the right one more likely. This idea was studied in a very similar setting for BKW in [44, 45]. Since the distinguishing problem for BKW and the dual attack is the same, these works should apply for the dual attacks too. This could lessen the impact of the problems introduced in [12].

The idea of limiting the number of hypotheses can also be applied to the positions on which we apply the FFT distinguisher. If the distinguisher suggests that the correct guess is a highly unlikely combination of secret key entries, then we discard this guess, assuming that an incorrect guess managed to perform the best by pure chance.

The improvement from lowering the number of samples needed for the guessing phase can be pushed even further. Since we can rank the samples resulting from lattice reduction (based on the Euclidean norm of the reduced positions), by only choosing the best samples our distinguisher will do an even better job. However, as the number of samples needed for guessing is roughly proportional to the logarithm of the number of hypotheses we make, we can expect that the total impact of limiting the number of hypotheses to be noticeable but not groundbreaking.

5.3.2 Re-sampling for Free in Dual Attacks

When the dual attack setting consists of applying the FFT on more positions than the ones to be enumerated (which is typically the case in the classical setting, but not the quantum one), then we can re-sample for free at least once. To re-sample we simply enumerate over (parts of) the positions where we applied the FFT and move (parts of) the FFT to the positions we used to enumerate over.

Unfortunately, this idea of swapping which positions we apply enumeration vs. FFT on is incompatible with the idea of limiting the number of hypotheses on the positions where we apply the FFT. We leave figuring out which idea leads to the larger improvement for future study.

6 Conclusions

The method presented in this paper improves upon 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. While the recent work by Ducas and Pulles [12] implies that these estimates are too optimistic, our enumeration strategy and estimation does still improve upon the dual attack on LWE.

The recent work of aborted key enumeration [15] - while leading to interesting results in the context of pure key enumeration - unfortunately does not seem to improve the dual attack on LWE that much. However, the reduced number of hypotheses needed when using aborted enumeration can lead to some improvement though, as discussed in Section 5.3.1.

Future research directions include the application of this method - whether using full or aborted enumeration - on other areas in cryptanalysis where enumeration of a vector with non-uniform values 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

[1] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing. STOC '05, pp. 84–93. Association for Computing Machinery, New York, NY, USA (2005). https://doi.org/10.1145/1060590.1060603 . https://doi.org/10.1145/1060590.1060603

[2] NIST: Post-Quantum Cryptography Standardization. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization

[3] Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016, pp. 3–33. Springer, Berlin, Heidelberg (2016)

[4] Albrecht, M.R., Cid, C., Faugère, J., Fitzpatrick, R., Perret, L.: On the complexity of the BKW algorithm on LWE. Des. Codes Cryptogr. **74**(2), 325–354 (2015)

[5] Arora, S., Ge, R.: New algorithms for learning in presence of errors. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) Automata, Languages and Programming, pp. 403–415. Springer, Berlin, Heidelberg (2011)

[6] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - A new hope. In: Holz, T., Savage, S. (eds.) USENIX Security 2016: 25th USENIX Security Symposium, pp. 327–343. USENIX Association, Austin, TX, USA (2016)

[7] Micciancio, D., Regev, O.: In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Lattice-based Cryptography, pp. 147–191. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_5 . https://doi.org/10.1007/978-3-540-88702-7_5

[8] Guo, Q., Johansson, T.: Faster dual lattice attacks for solving LWE with applications to CRYSTALS. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2021, Part IV. Lecture Notes in Computer Science, vol. 13093, pp. 33–62. Springer, Singapore (2021). https://doi.org/10.1007/978-3-030-92068-5_2

[9] MATZOV: Report on the Security of LWE: Improved Dual Lattice Attack. Zenodo (2022). https://doi.org/10.5281/zenodo.6412487 . https://doi.org/10.5281/zenodo.6412487

[10] Albrecht, M.R., Shen, Y.: Quantum Augmented Dual Attack. Cryptology ePrint Archive, Paper 2022/656. https://eprint.iacr.org/2022/656 (2022). https://eprint.iacr.org/2022/656

[11] Budroni, A., Mårtensson, E.: Improved estimation of key enumeration with applications to solving lwe. In: 2023 IEEE International Symposium on Information Theory (ISIT), pp. 495–500 (2023). https://doi.org/10.1109/ISIT54713.2023.10206474

[12] Ducas, L., Pulles, L.N.: Does the dual-sieve attack on learning with errors even work? In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023, pp. 37–69. Springer, Cham (2023)

[13] Carrier, K., Shen, Y., Tillich, J.-P.: Faster Dual Lattice Attacks by Using Coding Theory. Cryptology ePrint Archive, Paper 2022/1750. https://eprint.iacr.org/2022/1750 (2022)

[14] Pouly, A., Shen, Y.: Provable Dual Attacks on Learning with Errors. Cryptology ePrint Archive, Paper 2023/1508. https://eprint.iacr.org/2023/1508 (2023). https://eprint.iacr.org/2023/1508

[15] Glaser, T., May, A., Nowakowski, J.: Entropy Suffices for Key Guessing. Cryptology ePrint Archive, Paper 2023/797. https://eprint.iacr.org/2023/797 (2023). https://eprint.iacr.org/2023/797

[16] Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. STOC '96, pp. 212–219. Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237814.237866 . https://doi.org/10.1145/237814.237866

[17] Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemporary Mathematics **305**, 53–74 (2002)

[18] Lenstra, A.K., Lenstra, H.W., Lovasz, L.: Factoring Polynomials with Rational Coefficients. Mathematische Annalen **261**, 515–534 (1982)

[19] Schnorr, C.P.: A hierarchy of polynomial time lattice basis reduction algorithms. Theor. Comput. Sci. **53**(2), 201–224 (1987)

[20] Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 1–20. Springer, Seoul, South Korea (2011). https://doi.org/10.1007/978-3-642-25385-0_1

[21] Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing. STOC '01, pp. 601–610. Association for Computing Machinery, New York, NY, USA (2001). https://doi.org/10.1145/380752.380857 . https://doi.org/10.1145/380752.380857

[22] Becker, A., Ducas, L., Gama, N., Laarhoven, T.: New directions in nearest neighbor searching with applications to lattice sieving. In: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '16, pp. 10–24. Society for Industrial and Applied Mathematics, USA (2016)

[23] Laarhoven, T., Mosca, M., Pol, J.: Finding shortest lattice vectors faster using quantum search. Designs, Codes, and Cryptography **77**, 375–400 (2015)

[24] Chailloux, A., Loyer, J.: Lattice sieving via quantum random walks. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2021, pp. 63–91. Springer, Cham (2021)

[25] Bonnetain, X., Chailloux, A., Schrottenloher, A., Shen, Y.: Finding many collisions via reusable quantum walks. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, pp. 221–251. Springer, Cham (2023)

[26] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) Advances in Cryptology – CRYPTO 2009. Lecture Notes in Computer Science, vol. 5677, pp. 595–618. Springer, Santa Barbara, CA, USA (2009). https://doi.org/10.1007/978-3-642-03356-8_35

[27] Kirchner, P.: Improved Generalized Birthday Attack. Cryptology ePrint Archive, Report 2011/377. https://eprint.iacr.org/2011/377 (2011)

[28] Blum, A., Kalai, A.T., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. In: Symposium on the Theory of Computing (2000)

[29] Albrecht, M.R., Faugère, J.-C., Fitzpatrick, R., Perret, L.: Lazy modulus switching for the BKW algorithm on LWE. In: Krawczyk, H. (ed.) PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography. Lecture Notes in Computer Science, vol. 8383, pp. 429–445. Springer, Buenos Aires, Argentina (2014). https://doi.org/10.1007/978-3-642-54631-0_25

[30] Guo, Q., Johansson, T., Stankovski, P.: Coded-BKW: Solving LWE using lattice codes. In: Gennaro, R., Robshaw, M.J.B. (eds.) Advances in Cryptology – CRYPTO 2015, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 23–42. Springer, Santa Barbara, CA, USA (2015). https://doi.org/10.1007/978-3-662-47989-6_2

[31] Kirchner, P., Fouque, P.-A.: An improved BKW algorithm for LWE with applications to cryptography and lattices. In: Gennaro, R., Robshaw, M.J.B. (eds.) Advances in Cryptology – CRYPTO 2015, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 43–62. Springer, Santa Barbara, CA, USA (2015). https://doi.org/10.1007/978-3-662-47989-6_3

[32] Guo, Q., Johansson, T., Mårtensson, E., Stankovski, P.: Coded-BKW with sieving. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017, Part I. Lecture Notes in Computer Science, vol. 10624, pp. 323–346. Springer, Hong Kong, China (2017). https://doi.org/10.1007/978-3-319-70694-8_12

[33] Budroni, A., Guo, Q., Johansson, T., Mårtensson, E., Wagner, P.S.: Making the bkw algorithm practical for lwe. In: Bhargavan, K., Oswald, E., Prabhakaran, M. (eds.) Progress in Cryptology – INDOCRYPT 2020, pp. 417–439. Springer, Cham (2020)

[34] Budroni, A., Guo, Q., Johansson, T., Mårtensson, E., Wagner, P.S.: Improvements on making bkw practical for solving lwe. Cryptography **5**(4) (2021) https://doi.org/10.3390/cryptography5040031

[35] Massey, J.L.: Guessing and entropy. In: Proceedings of 1994 IEEE International Symposium on Information Theory, p. 204 (1994). https://doi.org/10.1109/ISIT.1994.394764

[36] Wikipedia contributors: Multinomial distribution — Wikipedia, The Free Encyclopedia. accessed 2023-08-09 (2023). https://en.wikipedia.org/wiki/Multinomial_distribution

[37] Montanaro, A.: Quantum search with advice. In: Dam, W., Kendon, V.M., Severini, S. (eds.) Theory of Quantum Computation, Communication, and Cryptography, pp. 77–93. Springer, Berlin, Heidelberg (2011)

[38] Wikipedia contributors: Euler–Maclaurin formula — Wikipedia, The Free Encyclopedia. accessed 2023-01-10 (2023). https://en.wikipedia.org/wiki/Euler-Maclaurin_formula

[39] Tănăsescu, A., Choudary, M.O., Rioul, O., Popescu, P.G.: Tight and scalable side-channel attack evaluations through asymptotically optimal massey-like inequalities on guessing entropy. Cryptography **23**(11) (2021)

[40] Rioul, O.: Variations on a theme by massey. IEEE Transactions on Information Theory **68**(5), 2813–2828 (2022) https://doi.org/10.1109/TIT.2022.3141264

[41] Gidney, C., Ekerå, M.: How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum **5**, 433 (2021) https://doi.org/10.22331/q-2021-04-15-433

[42] TU Darmstadt Learning with Errors Challenge. https://www.latticechallenge.org/lwe_challenge/challenge.php. Accessed: 2023-01-24

[43] Gama, N., Nguyen, P.Q., Regev, O.: Lattice enumeration using extreme pruning. In: Gilbert, H. (ed.) Advances in Cryptology – EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 257–278. Springer, French Riviera (2010). https://doi.org/10.1007/978-3-642-13190-5_13

[44] Guo, Q., Mårtensson, E., Wagner, P.S.: On the sample complexity of solving lwe using bkw-style algorithms. In: 2021 IEEE International Symposium on Information Theory (ISIT), pp. 2405–2410 (2021). https://doi.org/10.1109/ISIT45174.2021.9518190

[45] Guo, Q., Mårtensson, E., Stankovski Wagner, P.: Modeling and simulating the sample complexity of solving lwe using BKW-style algorithms. Cryptography and Communications **15**, 331–350 (2023)