Cryptography and Communications (2023) 15:331–350
https://doi.org/10.1007/s12095-022-00597-0

Modeling and simulating the sample complexity of solving LWE using BKW-style algorithms

Qian Guo[1] · **Erik Mårtensson**[1,2] · **Paul Stankovski Wagner**[1]

Received: 22 November 2021 / Accepted: 7 June 2022 / Published online: 9 August 2022

Abstract
The Learning with Errors (LWE) problem receives much attention in cryptography, mainly due to its fundamental significance in post-quantum cryptography. Among its solving algorithms, the Blum-Kalai-Wasserman (BKW) algorithm, originally proposed for solving the Learning Parity with Noise (LPN) problem, performs well, especially for certain parameter settings with cryptographic importance. The BKW algorithm consists of two phases, the reduction phase and the solving phase. In this work, we study the performance of distinguishers used in the solving phase. We show that the Fast Fourier Transform (FFT) distinguisher from Eurocrypt'15 has the same sample complexity as the optimal distinguisher, when making the same number of hypotheses. We also show via simulation that it performs much better than previous theory predicts and develop a sample complexity model that matches the simulations better. We also introduce an improved, pruned version of the FFT distinguisher. Finally, we indicate, via extensive experiments, that the sample dependency due to both LF2 and sample amplification is limited.

Keywords LWE · BKW · FFT distinguisher · Hypothesis testing

Mathematics Subject Classification (2010) 94A60 · 68P30

1 Introduction

Post-quantum cryptography studies replacements of cryptographic primitives based on the factoring or discrete-log problem, since both can be efficiently solved by a quantum computer [2]. Lattice-based cryptography is its main area. In the NIST Post-Quantum Cryptography Standardization [3], 5 out of 7 finalists and 2 out of 8 alternates are lattice-based.

The *Learning with Errors* (LWE) problem, introduced by Regev [4], is the major problem in lattice-based cryptography. Its average-case hardness can be based on the worst-case hardness

Part of the material in this paper was presented at the 2021 IEEE International Symposium on Information Theory (ISIT 2021), Melbourne, Australia, July 12-20, 2021 [1].

✉ Erik Mårtensson
erik.martensson@uib.no

[1] Department of Electrical and Information Technology, Lund University, Lund, Sweden

[2] Selmer Center, Department of Informatics, University of Bergen, Bergen, Norway

of some standard lattice problems, which is extremely interesting in theoretical crypto. The most famous, of its many cryptographic applications, is the design of Fully Homomorphic Encryption (FHE) schemes. Its binary counterpart, the *Learning Parity with Noise* problem (LPN), also plays an significant role in cryptography (see [5]), especially in light-weight cryptography for very constrained environments such as RFID tags and low-power devices.

The algorithms for solving LWE can be divided into lattice-based, algebraic, and combinatorial methods. The last class of algorithms all inherit from the famous Blum-Kalai-Wasserman (BKW) algorithm [6, 7], and are the most relevant to our study. We refer interested readers to [8] for concrete complexity estimation for solving LWE instances, and to [9, 10] for asymptotic complexity estimations.

The BKW-type algorithms include two phases, the reduction phase and the solving phase. The prior consists of a series of operations, called BKW steps, iteratively reducing the dimension of the problem at the cost of increasing its noise level. At the end of the reduction phase, the original LWE problem is transformed to a new problem with a much smaller dimension. The new problem can be solved efficiently by a procedure called distinguishing in the solving phase.

One of the main challenges in understanding the precise performance of BKW variants on solving the LWE problem comes from the lack of extensive experimental studies, especially on the various distinguishers proposed for the solving phase. Firstly, we have borrowed many heuristics from BKW variants on the LPN problem, but only very roughly or not at all verified them for the LWE problem. Secondly, the tightness of the nice theoretical bound in [11] on the sample complexity of the FFT distinguisher also needs to be experimentally checked. Lastly, a performance comparison of the different known distinguishers is still lacking.

1.1 Related work

The BKW algorithm proposed by Blum et al. [6, 7] is the first sub-exponential algorithm for solving the LPN problem. Its initial distinguisher, an exhaustive search method in the binary field, recovers one bit of the secret by employing majority voting. Later, Levieil and Fouque [12] applied the fast Walsh-Hadamard transform (FWHT) technique to accelerate the distinguishing process and recovered a number of secret bits in one pass. They also proposed some heuristic versions and tested these assumptions by experiments. In [13] Kirchner proposed a secret-noise transform technique to change the secret distribution to be sparse. This technique is an application of the transform technique proposed in [14] for solving LWE. Bernstein and Lange [15] further instantiated an attack on the Ring-LPN problem, a variant of LPN with algebraic ring structures. In [16, 17], Guo, Johansson, and Löndahl proposed a new distinguishing method called subspace hypothesis testing. Though this distinguisher can handle an instance with larger dimension by using covering codes, its inherent nature is still an FWHT distinguisher. Improvements of the BKW algorithm were further studied by Zhang et al. [18] and Bogos-Vaudenay [19]. An elaborate survey with experimental results on the BKW algorithm for solving LPN can be found in [20].

BKW for solving LWE follows a similar research line. Albrecht et al. initiated the study in [21]. In PKC 2014 [22], a new reduction technique called lazy modulus switching was proposed. In both works, the solving phase uses an exhaustive search approach. In [11] Duc et al. introduced the Fast Fourier Transform (FFT) technique in the distinguishing process and bounded the sample complexity theoretically from the Hoeffding inequality. Note that the actual performance regarding the bound is not experimentally verified and the information loss in the FFT distinguisher is unclear. There are new reduction methods in [23–25], and in [23], the authors also proposed a new method with polynomial reconstruction in

the solving phase. This method has the same sample complexity as that of the exhaustive search approach but requires $(q+1)$ FFT operations rather than only one FFT in [11]. The BKW variants with memory constraints were recently studied in [26–28].

1.2 Contributions

In the paper, we compare the performances of the known distinguishers empirically. We investigate the performance of the optimal distinguisher and the FFT distinguisher. We also test the sample dependency when using LF2 or sample amplification. We have the following contributions.

1. We show that the FFT distinguisher and the optimal distinguisher have the same sample complexity, if we make sure that the distinguishers make the same number of hypotheses. Thus, except for very sparse secrets, the FFT distinguisher is always preferable. This also makes the polynomial reconstruction method of [23] obsolete.
2. We indicate through simulation that the formula from [11] for the number of samples needed for distinguishing is off by roughly an order of magnitude. We develop a new sample complexity model, which matches the simulation values well.
3. We introduce a pruned FFT method. By only testing probable hypotheses, we improve the performance of the FFT method from [11] with no computational overhead.
4. We indicate that the sample dependency due to using LF2 or sample amplification is limited.

1.3 Organization

The rest of the paper is organized as follows. Section 2 introduces some necessary background. In Section 3 we cover the basic BKW algorithm. Section 4 goes over distinguishers used for hypothesis testing when solving LWE using BKW and introduces the pruned FFT method. Next, in Section 5 we show why the FFT distinguisher and the optimal distinguisher perform identically for our setting, followed by simulation results in Section 6. In Section 7 we develop a new model for the sample complexity of the FFT distinguisher. Finally, Section 8 concludes the paper.

2 Background

Let us introduce some notation. Bold small letters denote vectors. Let $\langle \cdot, \cdot \rangle$ denote the scalar products of two vectors with the same dimension. By $|x|$ we denote the absolute value of x for a real number $x \in \mathbb{R}$. We also denote by $R(y)$ the real part and $\|y\|$ the absolute value of a complex number $y \in \mathbb{C}$.

2.1 LWE

Let us define the LWE problem.

Definition 1 (LWE) Let n be a positive integer, q an odd prime. Let $\mathbf{s}$ be a uniformly random secret vector in $\mathbb{Z}_q^n$. Assume access to m noisy scalar products between $\mathbf{s}$ and known vectors $\mathbf{a_i}$, i.e.

$$b_i = \langle \mathbf{a_i}, \mathbf{s} \rangle + e_i, \tag{1}$$

for $i = 1,\ldots,m$. The error terms e_i are drawn from a distribution χ. The (search) LWE problem is to find $\mathbf{s}$.

Thus, when solving LWE you have access to a large set of pairs (a_i,b_i) and want to find the corresponding secret vector $\mathbf{s}$. Some versions restrict the number of available samples. If we let $\mathbf{b} = (b_1,b_2,\ldots,b_m)$, $\mathbf{e} = (e_1,e_2,\ldots,e_m)$ and $\mathbf{A} = [\mathbf{a}_1^T, \mathbf{a}_2^T \cdots \mathbf{a}_m^T]$ we can write the problem on matrix form as

$$\mathbf{b} = \mathbf{sA} + \mathbf{e}. \tag{2}$$

2.2 Rounded Gaussian distribution

For the error we use the rounded Gaussian distribution.[1] Let $f(x|0,\sigma^2)$ denote the PDF of the normal ditribution with mean 0 and standard deviation σ, this distribution in turn being denoted as $\mathcal{N}(0,\sigma^2)$. The rounded Gaussian distribution samples from $\mathcal{N}(0,\sigma^2)$, rounds to the nearest integer and wraps to the interval $[-(q-1)/2,(q-1)/2]$. In other words, the probability of choosing a certain error e is equal to

$$\sum_{k=-\infty}^{\infty} \int_{e-1/2+k\cdot q}^{e+1/2+k\cdot q} f(x|0,\sigma^2)dx,$$

for $e \in [-(q-1)/2,(q-1)/2]$. We denote this distribution by $\bar{\Psi}_{\sigma,q}$. We use the well-known heuristic approximation that the sum of two independent distributions X_1 and X_2, drawn from $\bar{\Psi}_{\sigma_1,q}$ and $\bar{\Psi}_{\sigma_2,q}$, is drawn from $\bar{\Psi}_{\sqrt{\sigma_1^2+\sigma_2^2},q}$. We also use the notation $\alpha = \sigma/q$. Finally, we let $\mathcal{U}(a,b)$ denote the discrete uniform distribution taking values from a up to b.

3 BKW

The BKW algorithm was originally invented to solve LPN. It was first used for LWE in [21]. The BKW algorithm consists of two parts, reduction and hypothesis testing.

3.1 Reduction

We divide samples into categories based on b position values in the $\mathbf{a}$ vectors. Two samples should be in the same category if and only if the b position values get canceled when adding or subtracting the $\mathbf{a}$ vectors. Given two samples $([\pm a_0,a_1],b_1)$ and $([\pm a_0,a_2],b_2)$ within the same category. By adding/subtracting the $\mathbf{a}$ vectors we get

[1] Also common is to use the Discrete Gaussian distribution, which is similar.

$$\mathbf{a_{1,2}} = [\underbrace{0\ 0\ \cdots\ 0}_{b \text{ symbols}}\ *\ *\ \cdots\ *\ ].$$

The corresponding b value is $b_{1,2} = b_1 \pm b_2$. Now we have a new sample $(a_{1,2},b_{1,2})$. The corresponding noise variable is $e_{1,2} = e_1 \pm e_2$, with variance $2\sigma^2$, where σ^2 is the variance of the original noise. By calculating a suitable number of new samples for each category we have reduced the dimensionality of the problem by b, but increased the noise variance to $2\sigma^2$. If we repeat the reduction process t times we end up with a dimensionality of $n - tb$, and a noise variance of $2^t \cdot \sigma^2$.

3.1.1 LF1 and LF2

LF1 and LF2 are two implementation tricks originally proposed for solving LPN in [12]. Both can naturally be generalized for solving LWE.

In LF1 we choose one representative per category. We form new samples by combining the other samples with the representative. This way all samples at the hypothesis testing stage are independent of each other. However, the sample size shrinks by $(q^b - 1)/2$ samples per generation, requiring a large initial sample size.

In LF2 we allow combining any pair of samples within a category, creating much more samples. If we form every possible sample, a sample size of $3(q^b - 1)/2$ is enough to keep the sample size constant between steps. The disadvantage of this approach is that the samples are no longer independent, leading to higher noise levels in the hypothesis stage of BKW. It is generally assumed that this effect is quite small. This assumption is well tested for solving the LPN problem [12].

3.1.2 Sample amplification

Some versions of LWE limit the number of samples. We can get more samples using sample amplification. For example, by adding/subtracting triples of samples we can increase the initial sample size m up to a maximum of $4 \cdot \binom{m}{3}$. This does increase the noise by a factor of $\sqrt{3}$. It also leads to an increased dependency between samples in the hypothesis testing phase, similar in principle to LF2.

3.1.3 Secret-noise transformation

There is a transformation of the LWE problem that makes the distribution of the secret vector follow the distribution of the noise [13, 14].

3.1.4 Improved reduction steps

There are many improvements of the plain BKW steps. Lazy modulus switching (LMS) was introduced in [22] and further developed in [24]. In [23] coded-BKW was introduced. Coded-BKW with sieving was introduced in [25] and improved in [10, 29].

Since only the final noise level, not the type of steps, matters for the distinguishers, we only use plain steps in this paper.

3.2 Hypothesis testing

Assume that we have reduced all but k positions to 0, leaving k positions for the hypothesis testing phase. After the reduction phase we have samples on the form

$$b = \sum_{i=1}^{k} a_i \cdot s_i + e \Leftrightarrow b - \sum_{i=1}^{k} a_i \cdot s_i = e, \tag{3}$$

where e is (approximately) rounded Gaussian distributed with a standard deviation of $\sigma_f = 2^{t/2} \cdot \sigma$ and mean 0. Now the problem is to distinguish the correct guess $\mathbf{s} = (s_1, s_2, \ldots, s_k)$ from all the incorrect ones, among all q^k guesses.[2] For each guess $\hat{\mathbf{s}}$ we calculate the corresponding error terms in (3). For the correct guess the observed values of e are rounded Gaussian distributed, while for the wrong guess they are uniformly random. How to distinguish the right guess from all the wrong ones is explained in Section 4.

4 Distinguishers

For the hypothesis testing we study the optimal distinguisher, which is an exhaustive search method; and a faster method based on the Fast Fourier Transform.

4.1 Optimal distinguisher

Let $D_{\hat{\mathbf{s}}}$ denote the distribution of the e values for a given guess of the secret vector $\hat{\mathbf{s}}$. As is shown in [30, Prop. 1] to optimally distinguish the hypothesis $D_{\hat{\mathbf{s}}} = \mathcal{U}(0, q-1)$ against $D_{\hat{\mathbf{s}}} = \bar{\Psi}_{\sigma_f, q}$ we calculate the log-likelihood ratio

$$\sum_{e=0}^{q-1} N(e) \log \frac{\Pr_{\bar{\Psi}_{\sigma_f,q}}(e)}{\Pr_{\mathcal{U}(0,q-1)}(e)} = \sum_{e=0}^{q-1} N(e) \log \left(q \cdot \Pr_{\bar{\Psi}_{\sigma_f,q}}(e) \right), \tag{4}$$

where $N(e)$ denotes the number of times e occurs for the guess $\hat{\mathbf{s}}$, σ_f denotes the standard deviation of the samples after the reduction phase and $\Pr_D(e)$ denotes the probability of drawing e from the distribution D. We choose the value $\hat{\mathbf{s}}$ that maximizes (4). The time complexity of this distinguisher is

$$\mathcal{O}(m \cdot q^k), \tag{5}$$

if we try all possible hypotheses. After performing the secret-noise transformation of Section 3.1.3 we can limit ourselves to assuming that the k values in $\mathbf{s}$ have an absolute value of at most d, reducing the complexity to

$$\mathcal{O}(m \cdot (2d+1)^k). \tag{6}$$

[2] After the secret-noise transforming most of these hypotheses are almost guaranteed to be incorrect, simplifying the hypothesis testing a bit.

By only testing the likely hypotheses we have a lower risk of choosing an incorrect one.[3] This trick of limiting the number of hypotheses can of course also be applied to the FFT method of Section 4.2, which we do in Section 4.4.

4.2 Fast Fourier Transform method

For LWE, the idea of using a transform to speed up the distinguishing was introduced in [11]. Consider the function

$$f(\mathbf{x}) = \sum_{j=1}^{m} \mathbb{1}_{\mathbf{a}_j=\mathbf{x}} \theta_q^{b_j}, \tag{7}$$

where $\mathbf{x} \in \mathbb{Z}_q^k$, $\mathbb{1}_{\mathbf{a}_j=\mathbf{x}}$ is equal to 1 if and only if $\mathbf{x} = \mathbf{a}_j$ and 0 otherwise, and θ_q denotes the q-th root of unity. The idea of the FFT distinguisher is to calculate the FFT of f, that is

$$\hat{f}(\boldsymbol{\alpha}) = \sum_{\mathbf{x}\in\mathbb{Z}_q^k} f(\mathbf{x}) \theta_q^{-\langle \mathbf{x},\boldsymbol{\alpha}\rangle} = \sum_{j=1}^{m} \theta_q^{-(\langle \mathbf{a}_j,\boldsymbol{\alpha}\rangle - b_j)}. \tag{8}$$

Given enough samples compared to the noise level, the correct guess $\boldsymbol{\alpha} = \mathbf{s}$ maximizes $\Re(\hat{f}(\boldsymbol{\alpha}))$ in (8).

The time complexity of the FFT distinguisher is

$$\mathcal{O}(m + k \cdot q^k \cdot \log(q)). \tag{9}$$

In general this complexity is much lower than the one in (5). However, it does depend on the sparsity of the secret $\mathbf{s}$. For a binary $\mathbf{s}$, the exhaustive methods are better.

From [11, Thm. 16] we have the following (upper limit) formula for the sample complexity of the FFT distinguisher

$$8 \cdot \ln\left(\frac{q^k}{\epsilon}\right)\left(\frac{q}{\pi}\sin\left(\frac{\pi}{q}\right)e^{-2\pi^2\sigma^2/q^2}\right)^{-2^{r+1}}, \tag{10}$$

where ϵ is the probability of guessing $\mathbf{s}$ incorrectly. Notice that the expression is slightly modified to fit our notation and that a minor error in the formula is corrected.[4]

4.3 Polynomial reconstruction method

In [23], a method combining exhaustive search and the FFT was introduced. It achieves optimal distinguishing information theoretically, while being more efficient than the optimal distinguisher. However, its complexity is roughly a factor q higher than the complexity of the FFT distinguisher.

[3] As long as the correct one is among our hypotheses.

[4] Using our notation k should be within the logarithm and not as a factor in front of it like in [11].

4.4 Pruned FFT distinguisher

Also when using an FFT distinguisher we can limit the number of hypotheses. We only need a small subset of the output values of the FFT distinguisher in (8), so we can speed-up the calculations using a pruned FFT. In general, if we only need K out of all N output values, the time complexity for calculating the FFT improves from $\mathcal{O}(N\log(N))$ to $\mathcal{O}(N\log(K))$ [31]. Limiting the magnitude when guessing the last k positions of $\mathbf{s}$ to d, this changes the time complexity from (9) to

$$\mathcal{O}(m + k \cdot q^k \cdot \log(2d+1)). \tag{11}$$

More importantly this method reduces the sample complexity. In the formula for sample complexity (10), the numerator q^k corresponds to the number of values that $\mathbf{s}$ can take on the last k positions. Re-doing the proofs of [11, Thm. 16], limiting the magnitude of the guess in each position to d, we get

$$8 \cdot \ln\left(\frac{(2d+1)^k}{\epsilon}\right)\left(\frac{q}{\pi}\sin\left(\frac{\pi}{q}\right)e^{-2\pi^2\sigma^2/q^2}\right)^{-2^{t+1}}. \tag{12}$$

This reduced sample complexity comes at no extra cost.

5 Equal performance of optimal and FFT distinguishers

When starting to run simulations, we noticed that the FFT distinguisher and the optimal distinguisher performed identically, in terms of number of samples to correctly guess the secret. We give a brief explanation of this phenomenon.[5]

Consider a sample on the form (3). By making a guess $\hat{\mathbf{s}}$ we calculate the corresponding error term $\hat{e}$. The Fourier transform of the FFT distinguisher in (8) can now be written as

$$\sum_{j=1}^{m} \theta_q^{\hat{e}_j}. \tag{13}$$

The real part (13) is equal to

$$\sum_{j=1}^{m} \cos\left(2\pi\hat{e}_j/q\right). \tag{14}$$

The FFT distinguisher picks the guess that maximizes (14). Now, let us rewrite (4) for the optimal distinguisher as

$$\sum_{j=1}^{m} \log\left(q \cdot \Pr_{\bar{\Psi}_{\sigma_f,q}}(\hat{e}_j)\right). \tag{15}$$

It turns out that with increasing noise level, the terms in (15) can be approximated as cosine functions with a period of q, as illustrated in Fig. 1. The terms correspond to $q = 1601$, starting with rounded Gaussian noise with $\alpha = 0.005$, $\sigma = \alpha \cdot q = 8.005$ and taking

[5] We do, of course, not claim that this is true in general for distinguishing distributions outside of our context of solving LWE using BKW.

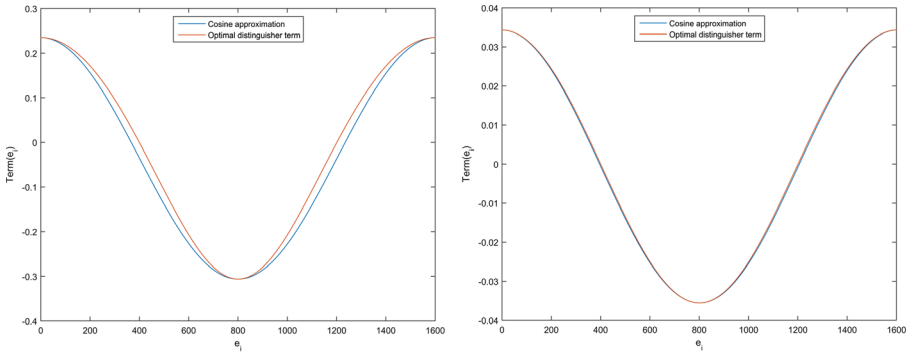

Fig. 1 Approximating the terms in (15) as cosine functions. (**a**) Taking 12 plain BKW steps (**b**) Taking 13 plain BKW steps

12 or 13 steps of plain BKW respectively. Notice that the approximation gets drastically better with increasing noise level.[6] The 13 step picture corresponds to the setting used in most of the experiments in Section 6. For a large-scale problem, the noise level would of course be much larger, resulting in an even better cosine approximation.

Since both distinguishers pick the $\hat{\mathbf{s}}$ that minimizes a sum of cosine functions with the same period, they will pick the same $\hat{\mathbf{s}}$, hence they will perform identically.

There are two immediate effects of this finding.

- The polynomial reconstruction method is obsolete.
- Unless the secret is very sparse, the FFT distinguisher is strictly better than the optimal distinguisher, since it is computationally cheaper.

Hence we limit our investigation to the FFT distinguisher from Section 6. We do not make any claims about the equivalance between the sample complexity of the two distinguishers outside of our context of solving LWE using BKW, when having large rounded (or Discrete) Gaussian noise.[7]

6 Simulations and results

This section covers the simulations we ran, using the FBBL library [32] from [33], and the results they yielded. For all figures, each point corresponds to running plain BKW plus distinguishing at least 30 times. For most points we ran slightly more iterations. See Appendix for details on the number of iterations for all the points. We chose our parameters inspired by the Darmstadt LWE Challenge [34].

The challenges are a set of (search) LWE instances used to compare LWE solving methods. Each instance consists of the dimension n, the modulus $q \approx n^2$, the relative error size

[6] Also notice that the approximation is not necessarily the best cosine approximation. It is simple the approximation that matches the largest and the smallest value of the curve.

[7] Although it could be interesting to investigate.

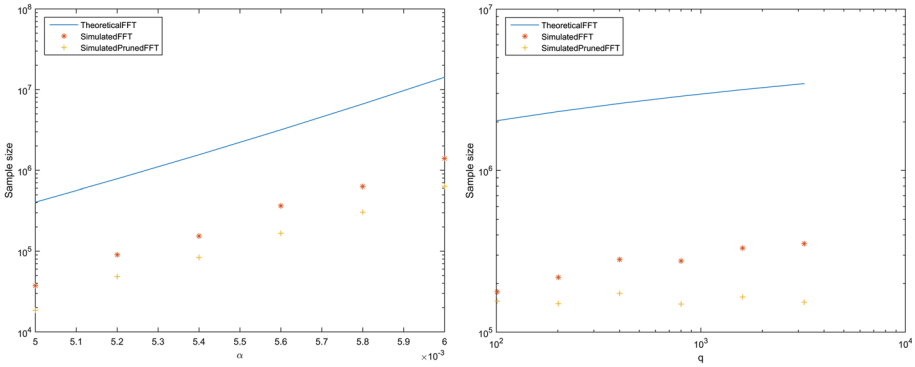

Fig. 2 Theoretical values vs. simulated values

α and $m \approx n^2$ equations of the form (1). Our simulations mostly use parameters inspired by the LWE challenges. We mostly let $q = 1601$ (corresponding to $n = 40$) and vary α to get problem instances that require a suitable number of samples for simulating hypothesis testing. The records for the LWE challenges are set using lattice sieving [35].

6.1 Varying noise level

In the upper part of Fig. 2 we compare the theoretical sample complexity from (10) with simulation results from an implementation of the FFT distinguisher of [11] and our pruned FFT distinguisher. The latter distinguisher guesses values of absolute value up to 3σ, rounded upwards. The simulated points are the median values of our simulations and the theoretical values correspond to setting $\epsilon = 0.5$ in (10). We use $q = 1601$, $n = 28$, we take $t = 13$ steps of plain BKW, reducing 2 positions per step. Finally we guess the last 2 positions and measure the minimum number of samples to correctly guess the secret. We vary α between 0.005 and 0.006. We use LF1 to guarantee that the samples are independent.

We notice that there is a gap of roughly a factor 10 between theory and simulation. More exactly, the gap is a factor [10.8277, 8.6816, 10.1037, 8.6776, 10.5218, 10.1564] for the six points, counting in increasing order of noise level.

We also see a gap between the FFT distinguisher and pruned FFT distinguisher. We can estimate the gap by comparing (12) and (10). Counting in increasing level of noise by theory we expect the gap to be [1.8056, 1.8056, 1.7895, 1.7743, 1.7598, 1.7461] for the 6 data points. The numbers from the simulation were [2.0244, 1.8610, 1.8433, 2.1905, 2.0665, 2.2060], matching theory well.

6.2 Varying *q*

In the lower part of Fig. 2 we show how the number of samples needed for distinguishing varies with q. For q we use the values [101, 201, 401, 801, 1601, 3201], for α we use the values [0.0896, 0.0448, 0.0224, 0.0112, 0.0056, 0.0028] and the number of steps were [5, 7, 9, 11, 13, 15]. Thereby the final noise level and the original $\mathbf{s}$ vectors have almost the same distribution, making the q values the only varying factor. We use LF1 to guarantee that the samples are independent.

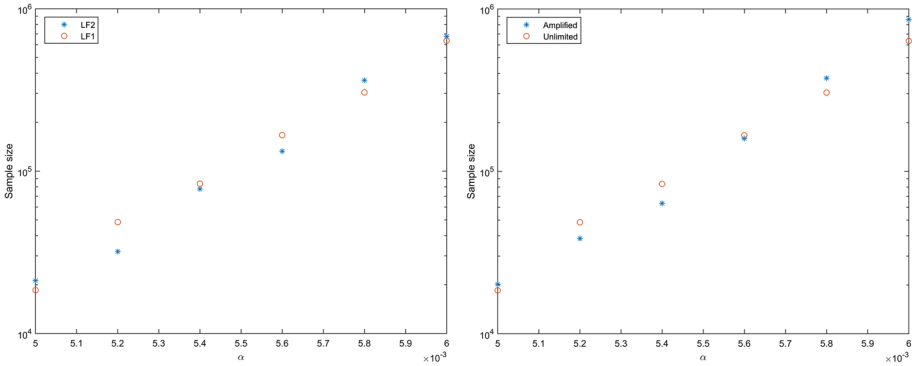

Fig. 3 Testing the effect of sample dependence

Notice that the number of samples needed to guess the secret is roughly an order of magnitude lower than theory predicts, counting in increasing order of q, the gain is a factor [11.4537, 10.6112, 9.2315, 10.4473, 9.5561, 9.7822] for the six points.

Also notice that the pruned version is an improvement, that increases with q. This is because the total number of hypotheses divided by the number of hypotheses we make increases with q. By comparing (12) and (10), we expect the improvement to be a factor [1.1303, 1.2871, 1.4563, 1.6152, 1.7743, 1.9334]. This is pretty close to the factors 1.1435, 1.4551, 1.6215, 1.8507, 2.0121, 2.3045] from simulation.

6.3 LF1 vs LF2

We investigate the increased number of samples needed due to dependencies, when using LF2. For LF2, depending on the number of samples needed for guessing, we used either the minimum number of samples to produce a new generation of the same size or a sample size roughly equal to the size needed for guessing at the end. To test the limit of LF2 we made sure to produce every possible sample from each category. See the upper part of Fig. 3 for details. The setting is the same as in Section 6.1. We only use the pruned FFT distinguisher. Notice that the performance is almost exactly the same in both the LF1 and the LF2 cases, as is generally assumed [12].

6.4 Sample amplification

The lower part of Fig. 3 shows the increased number of samples needed, due to sample amplification. We use $q = 1601$ and 1600 initial samples. We form new samples by combining triples of samples to get a large enough sample size. We vary the noise level between $\alpha = 0.005/\sqrt{3}$ and $\alpha = 0.006/\sqrt{3}$. We take 13 steps of plain BKW, reducing 2 positions per step. Finally we guess the last 2 positions and measure the minimum number of samples needed to guess correctly. We use LF1 and we compare the results against starting with as many samples as we want and noise levels between $\alpha = 0.005$ and $\alpha = 0.006$, both tricks to isolate the dependency due to sample amplification. We only use the pruned FFT distinguisher. The difference between the points is small, implying that the dependency due to sample amplification is limited.

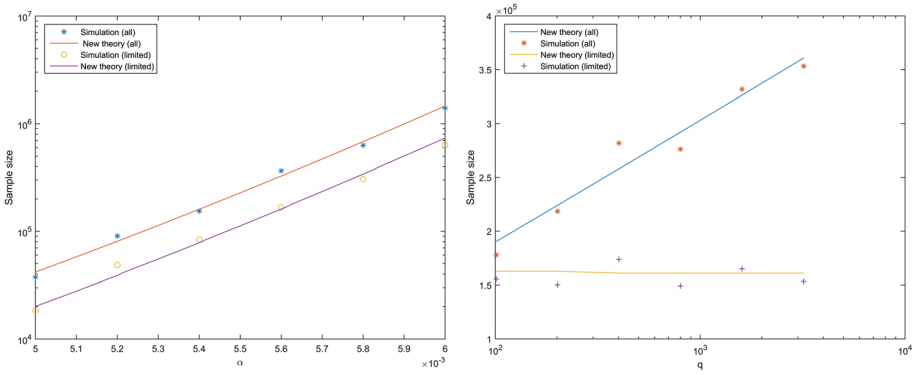

Fig. 4 New theoretical values vs. simulated values

7 Modeling the sample complexity

In this section we will develop a new model of the sample complexity of the FFT distinguisher.

Consider the real part of (13), for an incorrect guess. The sum is sampled from

$$\sum_{j=1}^{m} \cos\left(2\pi U_j/q\right), \tag{16}$$

where $U_j \sim \mathcal{U}(0, q-1)$. The expected value of (16) is 0. Let us denote the variance of each term of (16) by σ_U^2. For the correct guess the real part of (13) is equal to

$$\sum_{j=1}^{m} \cos\left(2\pi E_j/q\right), \tag{17}$$

where E_j is the sum of 2^t independent terms $e_j \sim \bar{\Psi}_{\sigma,q}$. Numerically we can calculate the mean and variance of each term of (17) with arbitrary precision. Denote these by μ_E and σ_E^2.

We can approximate the sum (16) as $X_i \sim \mathcal{N}(0, m \cdot \sigma_U^2)$ and (17) as $Y \sim \mathcal{N}(m \cdot \mu_E, m \cdot \sigma_E^2)$. If we make a total of h hypotheses we choose the correct hypothesis if

$$Y > \max(X_1, \ldots, X_h). \tag{18}$$

For a fixed value of y we have

$$\begin{aligned} P(\max(X_1, \ldots, X_h) < y) = \\ P(X_1 < y, \ldots, X_h < y) = \\ \Pi_{i=1}^{h} P(X_i < y) = \\ \phi\left(\frac{y}{\sqrt{m}\cdot\sigma_U}\right)^h. \end{aligned}$$

Thus the probability of choosing the correct hypothesis is equal to

$$\int_{-\infty}^{\infty} P(\max(X_1, \ldots, X_h) < y) f_Y(y) dy = \tag{19}$$

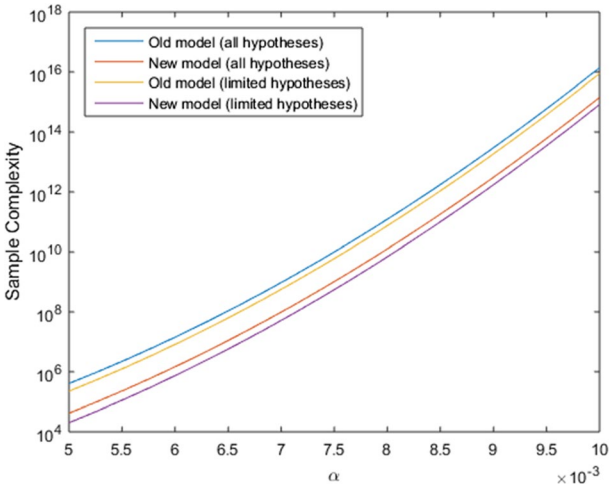

Fig. 5 New theoretical values vs. old theoretical values

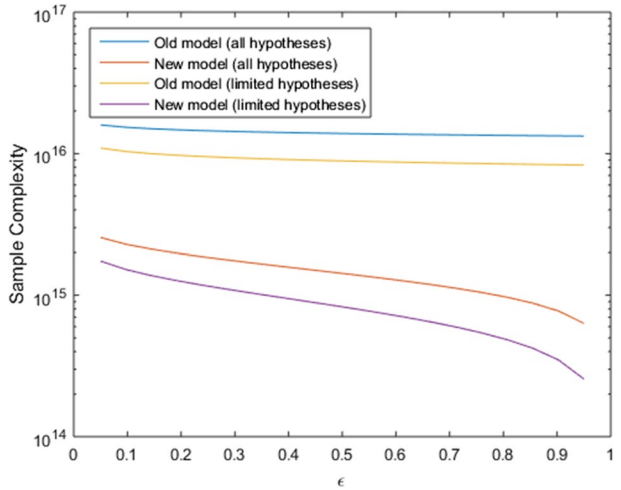

Fig. 6 New vs. old theoretical values when varying ϵ

$$\int_{-\infty}^{\infty} \phi\left(\frac{y}{\sqrt{m}\cdot\sigma_U}\right)^h \frac{1}{\sigma_E\sqrt{2\pi m}} e^{-\frac{1}{2}\left(\frac{y-m\cdot\mu_E}{\sqrt{m}\sigma_E}\right)^2} dy. \tag{20}$$

If we fix the failure probability to $\epsilon > 0$, then we can numerically find the value m that makes (20) equal to $1 - \epsilon$. Fixing $\epsilon = 0.5$, we can calculate the new theoretical sample complexities of the settings for Fig. 2 and compare these against the simulated ones, see Fig. 4 for details. Notice how the theoretical model matches the simulations very well in both settings.

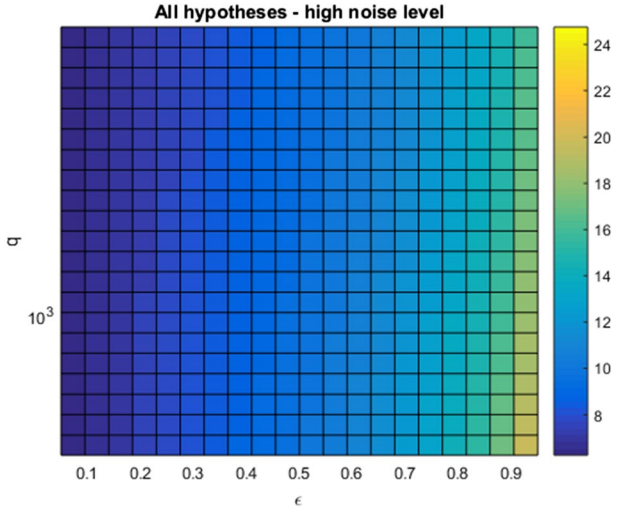

Fig. 7 New theoretical values vs. old theoretical values

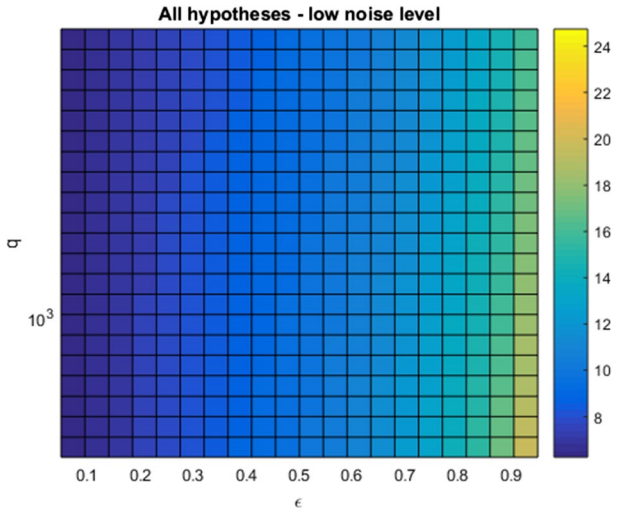

Fig. 8 New theoretical values vs. old theoretical values

7.1 Comparing the new model to previous theory

Here we discuss what predictions the new model makes, compared to the old model, for parameter settings beyond what we can simulate. First we use the same setting as in Fig. 2, but let α increase a bit further. See Fig. 5. We see that the predictions differ by a factor roughly 10 when testing all hypotheses and roughly 11 when limiting the number of hypotheses, almost independently of α. This is the behavior we observed in simulation too.

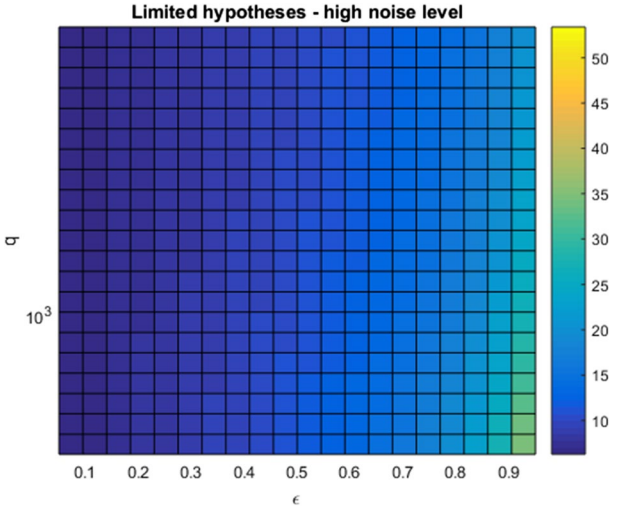

Fig. 9 New theoretical values vs. old theoretical values

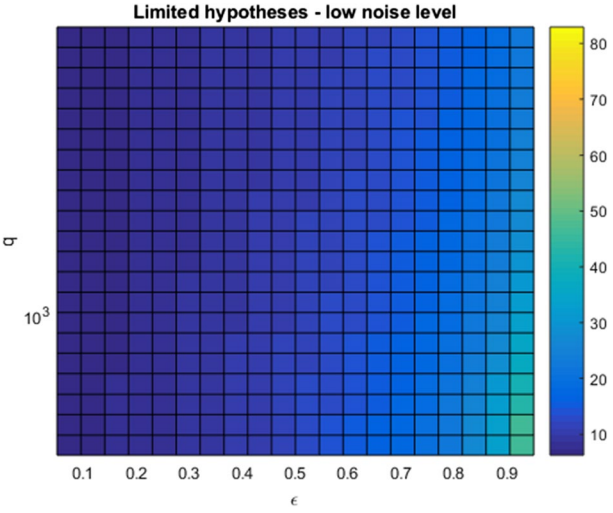

Fig. 10 New theoretical values vs. old theoretical values

Next, let us look at what happens when we vary ϵ. Fix $q = 1601$, $\alpha = 0.01$ and vary ϵ between 0.05 and 0.95. See Fig. 6. Notice for both settings that the gap between the old and the new theory increases with ϵ. The "constant" gap of a factor around 10 that we observed in simulations is what happens to be the gap for $\epsilon = 0.5$.

Next we look at what happens when we vary both q and ϵ. We vary ϵ between 0.05 and 0.95 and vary q between 401 and 6401.

We plot the estimated sample complexity of the old model divided by the estimated sample complexity of the new model in Figs. 7, 8, 9, and 10. In Figs. 7 and 8 we

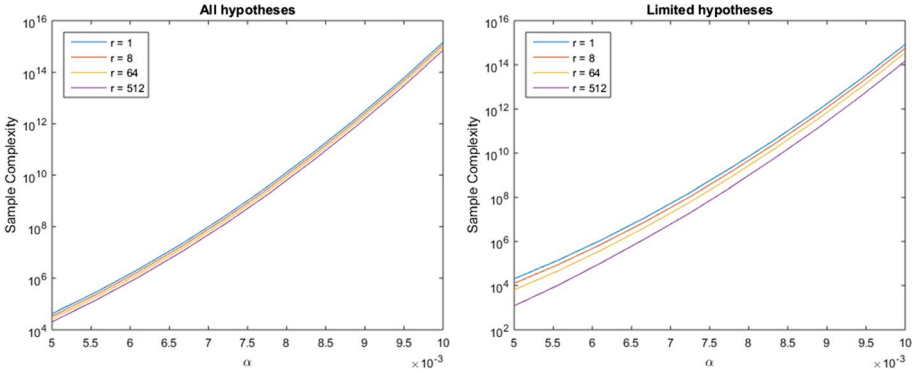

Fig. 11 Sample complexity using various values of r

estimate the sample complexity when testing all hypotheses, while in Figs. 9 and 10 we limit the number of hypotheses. In Figs. 7 and 9 we start at a high noise level using $\alpha = 0.01$, while in Figs. 8 and 10 we start at a low noise level using $\alpha = 0.005$. In all settings we see that the gap increases with ϵ. We also see for large values of ϵ that the gap is larger for the smaller values of q, especially when using a limited number of samples.

7.2 Probability of the correct guess being top *r*

We get a slightly different model if we only require that the correct hypothesis is any of the r top candidates. Sort and re-label the incorrect guesses in increasing order;

$$X_{(1)} < X_{(2)} < \cdots < X_{(h)}.$$

With this notation the k th smallest value has the CDF

$$F_{X_{(k)}}(x) = \sum_{j=k}^{h} \binom{h}{j} F_X(x)^j (1 - F_X(x))^{h-j},$$

see [36] for details. The r th largest values has the CDF

$$F_{X_{(h-r+1)}}(x) = \sum_{j=h-r+1}^{h} \binom{h}{j} F_X(x)^j (1 - F_X(x))^{h-j}.$$

When calculating this sum, it is useful to know that it is the complement probability of the CDF of a $Bin(h,F_X(x))$ distribution, evaluated at $h - r$.

Next, the probability of Y being larger than the r th largest value, in other words $P(Y > X_{(r)})$, is equal to

$$\int_{-\infty}^{\infty} F_{X_{(h-r+1)}}(y) f_Y(y) dy. \tag{21}$$

In Fig. 11 we estimate the sample complexity in the same setting as Fig. 5, for various values of r, testing all and limited number of hypotheses in the upper and the lower half of the figure respectively.

In both settings we can clearly reduce the sample complexity. It comes at a cost though. For each of the r hypotheses we need to backtrack one step and check whether the hypothesis is correct or not. When doing the backtracking for each of the r hypotheses, since the noise level after the previous reduction step is so low, the cost of testing the hypothesis gets reduced. We leave studying the details of this approach to future research.

8 Conclusions

We have shown that the FFT distinguisher and the optimal distinguisher have the same sample complexity for solving LWE using BKW. We have also showed that it performs roughly an order of magnitude better than the upper limit formula from [11, Thm. 16]. We have developed a new sample complexity model, which matches our simulatated complexities well. It also helps explain the gap between our simulated sample complexity originally reported in [1] and previous theory from [11]. Our pruned version of the FFT method improves the sample complexity of the FFT solver, at no cost. Finally, we have indicated that the sample dependency due to both LF2 and sample amplification is limited.

Appendix: : number of iterations in the simulations

The following is a collection of lists of the number of iterations used for each point to get the estimations of the median values in Figs. 3. For each figure and curve we list the number iterations from left to right in, in other words in increasing level of noise level α or modulus q.

Figure 2 - varying α

Simulated FFT	31	51	52	59	50	52
Simulated Pruned FFT	33	41	56	35	30	49

Figure 2 - varying q

Simulated FFT	100	100	95	80	67	82
Simulated Pruned FFT	100	100	95	80	67	82

Figure 3 - LF1 vs. LF2

LF1	33	41	56	35	30	49
LF2	43	46	69	37	69	50

Figure 3 - unlimited vs. sample amplification

Unlimited Samples	33	41	56	35	30	49
Sample Amplification	37	59	38	45	47	40

Acknowledgments This work was supported in part by the Swedish Research Council (Grant No. 2019-04166) and the Swedish Foundation for Strategic Research (Grant No. RIT17-0005 and strategic mobility grant No. SM17-0062). This work was also partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Finally, this work was partially supported by the project "Kvantesikker Kryptografi" from the National Security Authority of Norway.

Funding Open access funding provided by University of Bergen (incl Haukeland University Hospital)

References

1. Guo, Q., Mårtensson, E., Stankovski Wagner, P: On the sample complexity of solving LWE using BKW-style algorithms. In: 2021 IEEE International Symposium on Information Theory (ISIT) (2021)
2. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science, pp 124–134. IEEE Computer Society Press, Santa Fe (1994)
3. NIST Post-Quantum Cryptography Standardization, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization, accessed: 2019-09-24
4. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th Annual ACM Symposium on Theory of Computing, pp 84–93. ACM Press, Baltimore (2005)
5. Blum, A., Furst, M. L., Kearns, M. J., Lipton, R. J.: Cryptographic primitives based on hard learning problems. In: Stinson, D.R. (ed.) Advances in Cryptology – CRYPTO'93, ser. Lecture Notes in Computer Science, vol. 773, pp 278–291. Springer, Santa Barbara (1994)
6. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. In: 32nd Annual ACM Symposium on Theory of Computing, pp 435–440. ACM Press, Portland (2000)
7. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM **50**(4), 506–519 (2003). [Online]. Available: https://doi.org/10.1145/792538.792543
8. Albrecht, M. R., Player, R., Scott, S: On the concrete hardness of learning with errors. J. Mathematical Cryptology **9**(3), 169–203 (2015)
9. Herold, G., Kirshanova, E., May, A.: On the asymptotic complexity of solving LWE. Des. Codes Cryptogr. **86**(1), 55–83 (2018). [Online]. Available: https://doi.org/10.1007/s10623-016-0326-0
10. Guo, Q., Johansson, T., Mårtensson, E., Stankovski Wagner, P.: On the asymptotics of solving the LWE problem using coded-bkw with sieving. IEEE Trans. Information Theory **65**(8), 5243–5259 (2019). [Online]. Available: https://doi.org/10.1109/TIT.2019.2906233

11. Duc, A., Tramèr, F., Vaudenay, S.: Better algorithms for LWE and LWR. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology – EUROCRYPT 2015, Part I, ser. Lecture Notes in Computer Science, vol. 9056, pp 173–202. Springer, Sofia (2015)
12. Levieil, É., Fouque, P.-A.: An improved LPN algorithm. In: Prisco, R.D., Yung, M. (eds.) SCN 06: 5th International Conference on Security in Communication Networks, ser. Lecture Notes in Computer Science, vol. 4116, pp 348–359. Springer, Maiori (2006)
13. Kirchner, P.: Improved generalized birthday attack, Cryptology ePrint Archive, Report 2011/377 (2011) http://eprint.iacr.org/2011/377
14. 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, ser. Lecture Notes in Computer Science, vol. 5677, pp 595–618. Springer, Santa Barbara (2009)
15. Bernstein, D.J., Lange, T.: Never trust a bunny, Cryptology ePrint Archive, Report 2012/355 (2012) http://eprint.iacr.org/2012/355
16. Guo, Q., Johansson, T., Löndahl, C.: Solving LPN using covering codes. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology – ASIACRYPT 2014, Part I, ser. Lecture Notes in Computer Science, vol. 8873, pp 1–20. Springer, Kaoshiung (2014)
17. Guo, Q., Johansson, T., Löndahl, C.: Solving LPN using covering codes. J. Cryptology **33**(1), 1–33 (2020). [Online]. Available: https://doi.org/10.1007/s00145-019-09338-8
18. Zhang, B., Jiao, L., Wang, M.: Faster algorithms for solving LPN. In: Fischlin, M., Coron, J.-S. (eds.) Advances in Cryptology – EUROCRYPT 2016, Part I, ser. Lecture Notes in Computer Science, vol. 9665, pp 168–195. Springer, Vienna (2016)
19. Bogos, S., Vaudenay, S.: Optimization of LPN solving algorithms. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016, Part I, ser. Lecture Notes in Computer Science, vol. 10031, pp 703–728. Springer, Hanoi (2016)
20. Bogos, S., Tramèr, F., Vaudenay, S.: On solving L P N using B K W and variants - implementation and analysis. Cryptogr Commun **8**(3), 331–369 (2016). [Online]. Available: https://doi.org/10.1007/s12095-015-0149-2
21. Albrecht, M. R., Cid, C., Faugère, J. -C., Fitzpatrick, R., Perret, L: On the complexity of the BKW algorithm on LWE. Des Codes Cryptogr **74**(2), 325–354 (2015)
22. 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, ser. Lecture Notes in Computer Science, vol. 8383, pp 429–445. Springer, Buenos Aires (2014)
23. 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, ser. Lecture Notes in Computer Science, vol. 9215, pp 23–42. Springer, Santa Barbara (2015)
24. 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, ser. Lecture Notes in Computer Science, vol. 9215, pp 43–62. Springer, Santa Barbara (2015)
25. Guo, Q., Johansson, T., Mårtensson, E., Stankovski, P.: Coded-BKW with sieving. In: Advances in Cryptology – ASIACRYPT 2017, Part I, ser. Lecture Notes in Computer Science. In: Takagi, T., Peyrin, T. (eds.) , vol. 10624, pp 323–346. Springer, Hong Kong (2017)
26. Esser, A., Kübler, R., May, A.: LPN decoded. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology – CRYPTO 2017, Part II, ser. Lecture Notes in Computer Science, vol. 10402, pp 486–514. Springer, Santa Barbara (2017)
27. Esser, A., Heuer, F., Kübler, R., May, A., Sohler, C.: Dissection-BKW. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018, Part II, ser. Lecture Notes in Computer Science, vol. 10992, pp 638–666. Springer, Santa Barbara (2018)
28. Delaplace, C., Esser, A., May, A.: Improved low-memory subset sum and LPN algorithms via multiple collisions. In: Albrecht, M. (ed.) 17th IMA International Conference on Cryptography and Coding, ser. Lecture Notes in Computer Science, vol. 11929, pp 178–199. Springer, Oxford (2019)
29. Mårtensson, E.: The asymptotic complexity of coded-bkw with sieving using increasing reduction factors. In: IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7-12, 2019. [Online]. Available: https://doi.org/10.1109/ISIT.2019.8849218, pp 2579–2583. IEEE (2019)
30. Baignères, T., Junod, P., Vaudenay, S.: How far can we go beyond linear cryptanalysis? In: Lee, P.J. (ed.) Advances in Cryptology – ASIACRYPT 2004, ser. Lecture Notes in Computer Science, vol. 3329, pp 432–450. Springer, Jeju Island (2004)
31. Sorensen, H. V., Burrus, C. S.: Efficient computation of the dft with only a subset of input or output points. IEEE Trans. Signal Process. **41**(3), 1184–1200 (1993)

32. Budroni, A., Mårtensson, E., Stankovski Wagner, P.: FBBL - file-Based BKW for LWE https://github.com/{{FBBL}}/fbbl (2020)
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 417–39. Springer International Publishing, Cham (2020)
34. TU Darmstadt Learning with Errors Challenge, https://www.latticechallenge.org/lwe_challenge/challenge.php, accessed: 2020-09-30
35. Albrecht, M. R., Ducas, L., Herold, G., Kirshanova, E., Postlethwaite, E. W., Stevens, M.: The general sieve kernel and new records in lattice reduction. In: Ishai, Y. , Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019, Part II, ser. Lecture Notes in Computer Science, vol. 11477, pp 717–746. Springer, Darmstadt (2019)
36. Wikipedia contributors: Cumulative distribution function of order statistics — Wikipedia, the free encyclopedia, (2021) [Online; accessed 2021-09-29]. [Online]. Available: https://en.wikipedia.org/wiki/Orderstatistic#Cumulative_distribution_function_of_order_statistics