ON A HYBRID APPROACH TO SOLVE BINARY-LWE

A DUAL/ENUMERATION HYBRID TECHNIQUE FOR BINARY SECRET LEARNING WITH ERROR AND APPLICATION TO SECURITY ESTIMATES OF TFHE AND FHEW

THOMAS ESPITAU[1], ANTOINE JOUX[2], AND NATALIA KHARCHENKO[3]

ABSTRACT. In this paper, we investigate the security of binary secret Learning With Error (LWE). To do so, we improve the classical dual lattice attack. More precisely, we use the dual attack on a projected sublattice, which allows to generate instances of the LWE problem with a slightly bigger noise that correspond to a fraction of the secret key. Then, we search for the fraction of the secret key by computing the corresponding noise for each candidate using the constructed LWE samples. As secrets are binary vectors, we can perform the search step very efficiently by exploiting the recursive structure of the search space. This approach offers a trade-off between the cost of lattice reduction and the complexity of the search part which allows to speed up the attack. As an application we revisit the security estimates of the Fast Fully Homomorphic Encryption scheme over the Torus (TFHE) which is one of the fastest homomorphic encryption schemes based on the LWE problem. We provide an estimate of the complexity of our method for various parameters under three different cost models for lattice reduction and show that security level of the TFHE scheme should be re-evaluated according to the proposed improvement. Our estimates show that the current security level of the TFHE scheme should be reduced by 10 bits for the parameters proposed in the latest version of the scheme and by 7 bits for the recent update of the parameters that are used in the implementation, available online.

1. INTRODUCTION

The Learning With Errors (LWE) problem was introduced by Regev [Reg05] in 2005. A key advantage of LWE is that it is provably as hard as certain lattice approximation problems in the worst-case [BLP+13]. Since its introduction, the LWE problem has been a rich source of cryptographic constructions. The original Regev's work presents a LWE-based public-key encryption

[1] NTT CORPORATION, TOKYO, JAPAN

[2] INSTITUT DE MATHÉMATIQUES DE JUSSIEU–PARIS RIVE GAUCHE, CNRS, INRIA, UNIV PARIS DIDEROT, PARIS, FRANCE AND CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY, SAARBRÜCKEN, GERMANY

[3] SORBONNE UNIVERSITÉ, LIP 6, CNRS UMR 7606, PARIS, FRANCE AND SORBONNE UNIVERSITÉ

E-mail addresses: t.espitau@gmail.com.

scheme, but besides public key encryption, this problem can be used to build other primitives such as identity-based encryption [GPV08] or even fully homomorphic encryption [BV11].

Fully homomorphic encryption (FHE) allows to perform arbitrary operations on encrypted data without decrypting it. The first fully homomorphic encryption scheme was introduced by Gentry in 2009 [G$^+$09]. Since that time many FHE schemes were proposed, each offering new improvements (e.g. [FV12, GSW13, BGV14, CS15, DM15]). Security of many existing FHE constructions is based on the hardness of the LWE problem and its variations.

In the original formulation of the LWE problem, the secret vector is sampled uniformly at random from $\mathbb{Z}_q^n$, but more recent LWE-based constructions choose to use a more restricted distribution of the secret key in order to be more efficient. For example, some FHE schemes use binary [DM15, CGGI16], ternary [CLP17], or even ternary sparse secrets [HS15]. There are theoretical results supporting these choices which show that the LWE remains hard even with small secrets [BLP$^+$13]. In practice, however, more restricted distributions of the secret key can lead into more efficient attacks [BG14, SC19, CHHS19].

In this paper, we are interesting in evaluating the practical security of the LWE-based constructions with binary secrets. As an application, we consider the bit-security of the Fast Fully Homomorphic Encryption scheme over the Torus [CGGI16, CGGI17, CGGI20], which is currently one of the fastest FHE schemes. In TFHE, the gate bootstrapping can be performed in time about 10-20 ms. The security of the TFHE scheme relies on the hardness of a Learning with Errors (LWE) problem variant, named Torus-LWE. As a "learning a character" problem, it encompasses both the celebrated LWE and ring-LWE problems.

The security of a cryptosystem, of course, depends on the complexity of the most efficient known attack against it. In particular, to estimate the security of a LWE-based construction, it is important to know which attack is the best for the parameters used in the construction. It can be a difficult issue; indeed, the survey of existing attacks against LWE given in [APS15] shows that no known attack would be the best for all sets of LWE parameters.

In the case of TFHE, in [CGGI17], the authors adapted and used the dual distinguishing lattice attack from [Alb17] to evaluate the security of their scheme. Recently, in [CGGI20, Remark 9], the authors propose an updated set of the parameters for their scheme and estimate the security of the new parameters using the LWE estimator from [ACD$^+$18]. As it turns out, for both sets of the parameters, this leads to an overestimate of the security level.

1.1. **Our contribution.** In this work, we generalize the dual lattice attack which is currently used to evaluate the security of the TFHE scheme. First, we present a complete and detailed analysis of the standard dual lattice attack[1] on LWE from [CGGI20]. Then, we show that applying the dual attack to a projected sublattice and combining it with the search for a fraction of the key can yield a more efficient attack.

More precisely, our attack starts by applying lattice reduction to a projected sublattice in the same way it is applied to the whole lattice in the dual attack with lazy modulus switching. This way we generate LWE instances with bigger noise but in smaller dimension, corresponding to a fraction of the secret key. Then, the freshly obtained instances are used to recover the remaining fraction of the secret key. For each candidate for this missing fraction, we compute the noise vector corresponding to the LWE instances obtained at the previous step. This allows us to perform a majority voting procedure to detect the most likely candidates. As the TFHE scheme uses binary vectors for keys, this step boils down to computing a product of a matrix composed of the LWE samples with the matrix composed of all binary strings of length equal to the dimension of the part of the secret key that we are searching for. We show that this computation can be performed efficiently thanks to the recursive structure of the corresponding search space. The number of bits of the secret key that the attack aims to guess gives an additional parameter for tuning the complexity of the attack. Hence, this hybrid approach offers a trade-off between the quality of lattice reduction in the dual attack part and the time subsequently spent in the exhaustive search part. Together with the efficient computation of the noise for each candidate, the optimal parameters for this trade-off gives an asymptotic improvement of the whole complexity.

We evaluate the complexity of the standard dual attack and of our attack for a wide range of LWE parameters. We estimate the complexities of both attacks under three different models of the lattice reduction. For all the models, our estimates show that our attack outperforms the standard dual attack.

In particular, we estimate the complexity of our attack for the parameters used in the TFHE scheme. The TFHE scheme uses two keys: the switching key and the bootstrapping key. Thus, the security of the scheme is measured by the security of the weakest of the two keys.

We shall point out that the parameters of the scheme have been re-evaluated between the initial publication and the final journal version. The latest version of the paper [CGGI20] contains both the old and the new parameters. The security level for the old parameters (given

[1] We shall remark that this attack is slightly more subtle than the classical dual lattice attack, as it encompasses a continuous relaxation of the *lazy modulus switching* technique of [Alb17].

by [CGGI20, Table 3]) of the scheme is evaluated according to the dual attack, which the authors adapt to the settings of the TFHE scheme in [CGGI20, Section 7]. The new parameters are introduced by [CGGI20, Remark 9] and described in [CGGI20, Table 4]. Their security is evaluated using the LWE estimator from [ACD+18]. Also, recently, the on-line implementation of the scheme was updated and another set of the parameters appeared with the update (see [G+16, v1.1 – updated security parameters release, date : 2020.02.21]). For completeness, we re-evaluated the security of all available sets of the parameters. We describe all the choices in Table 1 and showcase the corresponding estimated security within our attack framework.

	parameters (n, α)	λ from [CGGI20]	λ: our attack (sieving model)
Old param.	switching key $n = 500, \alpha = 2.43 \cdot 10^{-5}$	**159**	**94**
	bootstrapping key $n = 1024, \alpha = 3.73 \cdot 10^{-9}$	198	112
New param.	switching key $n = 612, \alpha = 2^{-15}$	**128**	**118**
	bootstrapping key $n = 1024, \alpha = 2^{-26}$	129	120
Implementation param.	switching key $n = 630, \alpha = 2^{-15}$	**128**	**121**
	bootstrapping key $n = 1024, \alpha = 2^{-25}$	130	125

TABLE 1. Security of the parameters of TFHE scheme from [CGGI20, Table 3, Table 4] and from the public implementation [G+16]. n denotes the dimension, α is the parameter of the modular Gaussian distribution, λ denotes bit-security. The bold numbers denote the overall security of the scheme for a given set of parameters.

1.2. **Related work.** The survey [APS15] outlines three strategies for attacks against LWE: exhaustive search, BKW algorithm [BKW03, ACF+15] and lattice reduction. Lattice attacks against LWE can be separated into three categories depending on the lattice used: distinguishing dual attacks [Alb17], decoding (primal) attacks [LP11, LN13], and solving LWE by reducing it to unique-SVP: the unique Shortest Vector Problem [AFG13].

The idea of hybrid lattice reduction attack was introduced by Howgrave–Graham in [HG07]. He proposed to combine a meet-in-the-middle attack with lattice reduction to attack NTRUEncrypt. Then, Buchmann et al. adapted Howgrave–Graham's attack to the settings of LWE with binary error [BGPW16] and showed that the hybrid attack outperforms existing algorithms for some sets of parameters. This attack uses the decoding (primal) strategy for the lattice reduction part. Following these two works, Wunderer has provided an improved analysis of the hybrid decoding lattice attack and meet-in-the-middle attack and re-estimated security of several LWE and NTRU based cryptosystems in [Wun16]. Also, very recently, a similar combination of primal lattice attack and meet-in-the-middle attack was applied to LWE with ternary and sparse secret [SC19]. This last reference shows that the hybrid attack can also outperform other attacks in the case of ternary and sparse secrets for parameters typical for FHE schemes.

A combination of dual lattice attack and exhaustive search for a part of the secret key was considered in [Alb17, Section 5]. This work considers a hybrid dual attack in context of sparse secret keys. Also, recently, a similar approach was adapted to the case of ternary and sparse keys in [CHHS19]. The main difference of this work compared to [CHHS19, Alb17] is that the secret is non-sparse and binary in case of TFHE, thus, the search part of the hybrid attack requires a different approach.

Outline. This paper is organized as follows. In Section 2, we provide background. In Section 3, we revisit the dual lattice attack which was originally used to estimate the security level of TFHE. In Section 4, we describe our hybrid dual lattice attack. In Section 5, we compare the complexities of two attacks, revisit the security estimate of the TFHE scheme and provide some experimental evidence supporting our analysis.

2. Background

We use column notation for vectors and denote them using bold lower-case letters (e.g. $\mathbf{x}$). Matrices are denoted using bold upper-case letters (e.g. $\mathbf{A}$). For a vector $\mathbf{x}$, $\mathbf{x}^t$ denotes the transpose of $\mathbf{x}$, i.e., the corresponding row-vector. Base-2 logarithm is denoted as $\log$, natural logarithm is denoted as $\ln$. We denote the set of real numbers modulo 1 as the torus $\mathbb{T}$. For a finite set S, we denote by $\mathcal{U}(S)$ the discrete uniform distribution on S. For any compact set $S \subset \mathbb{R}^n$, the uniform distribution over S is also denoted by $\mathcal{U}(S)$. When S is not specified, $\mathcal{U}$ denotes uniform distribution over $(-0.5; 0.5)$.

2.1. **LWE problem.** Abstractly, all operations of the TFHE scheme are defined on the real torus $\mathbb{T}$ and to estimate the security of the scheme it is convenient to consider a scale-invariant version of LWE problem.

Definition 2.1 (Learning with Errors, [BLP+13, Definition 2.11])**.** Let $n \geqslant 1$, $\mathrm{s} \in \mathbb{Z}^n$, ξ be a distribution over $\mathbb{R}$ and $\mathcal{S}$ be a distribution over $\mathbb{Z}^n$.

We define the *LWE*$_{\mathbf{s},\xi}$ *distribution* as the distribution over $\mathbb{T}^n \times \mathbb{T}$ obtained by sampling $\mathbf{a}$ from $\mathcal{U}(\mathbb{T}^n)$, sampling e from ξ and returning $(\mathbf{a}, \mathbf{a}^t\mathbf{s} + e)$.

Given access to outputs from this distribution, we can consider the two following problems:

- *D*ecision-LWE. Distinguish, given arbitrarily many samples, between $\mathcal{U}(\mathbb{T}^n \times \mathbb{T})$ and LWE$_{\mathbf{s},\xi}$ distribution for a fixed s sampled from $\mathcal{S}$.
- *S*earch-LWE. Given arbitrarily many samples from LWE$_{\mathbf{s},\xi}$ distribution with fixed s $\leftarrow$ $\mathcal{S}$, recover the vector s.

To complete the description of the LWE problem we need to choose the error distribution ξ and the distribution of the secret key $\mathcal{S}$. Following the description of the TFHE scheme, we choose $\mathcal{S}$ to be $\mathcal{U}(\{0,1\}^n)$ and ξ to be a centered continuous Gaussian distribution, i.e. , we consider the LWE problem with binary secret. In [BLP+13], it is shown that this variation of LWE with binary secret remains hard. Finally, we use the notation LWE$_{\mathbf{s},\sigma}$ as a shorthand for LWE$_{\mathbf{s},\xi}$, when ξ is the Gaussian distribution centered at 0 and with standard deviation σ.

2.2. **Lattices.** A *lattice* Λ is a discrete subgroup of $\mathbb{R}^d$. As such, a lattice Λ of rank n can be described as a set of all integer linear combinations of $n \leqslant d$ linearly independent vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_d\} \subset \mathbb{R}^d$:

$$\Lambda = \mathcal{L}(\mathbf{B}) := \mathbb{Z}\mathbf{b}_1 \oplus \cdots \oplus \mathbb{Z}\mathbf{b}_d,$$

called a basis. Bases are not unique, one lattice basis may be transformed into another one by applying an arbitrary unimodular transformation. The *volume of the lattice* $\mathrm{vol}(\Lambda)$ is equal to the square root of the determinant of the Gram matrix $\mathbf{B}^t\mathbf{B}$: $\mathrm{vol}(\Lambda) = \sqrt{\det(\mathbf{B}^t\mathbf{B})}$. For every lattice Λ we denote the length of its shortest non-zero vector as $\lambda_1(\Lambda)$. Minkowski's theorem states that $\lambda_1(\Lambda) \leqslant \sqrt{\gamma_n} \cdot \mathrm{vol}(\Lambda)^{1/n}$ for any d-dimensional lattice Λ, where $\gamma_d < d$ is d-dimensional Hermite's constant. The problem of finding the shortest non-zero lattice vector is called the *Shortest Vector Problem*(SVP). It is known to be NP-hard under randomized reduction [Ajt98].

2.3. **Lattice reduction.** A lattice reduction algorithm is an algorithm which, given as input some basis of the lattice, finds a basis that consists of relatively short and relatively pairwise-orthogonal vectors. The quality of the basis produced by lattice reduction algorithms is often measured by the Hermite factor $\delta = \frac{\|\mathbf{b}_1\|}{\det(\Lambda)^{1/d}}$, where $\mathbf{b}_1$ is the first vector of the output basis. Hermite factors bigger than $\left(\frac{4}{3}\right)^{n/4}$ can be reached in polynomial time using the LLL algorithm [LLL82]. In order to obtain smaller Hermite factors, blockwise lattice reduction algorithms, like BKZ-2.0 [CN11] or S-DBKZ [MW16], can be used. The BKZ algorithm takes as input a basis of dimension d and proceeds by solving SVP on lattices of dimension $\beta < d$ using sieving [BDGL16] or enumeration [GNR10]. The quality of the output of BKZ depends on the blocksize β. In [HPS11] it is shown that after a polynomial number of calls to SVP oracle, the BKZ algorithm with blocksize β produces a basis $\mathbf{B}$ that achieves the following bound:

$$\|\mathbf{b}_1\| \leqslant 2\gamma_\beta^{\frac{d-1}{2(\beta-1)}+\frac{3}{2}} \cdot \mathrm{vol}(\mathbf{B})^{1/d}.$$

However, up to our knowledge, there is no closed formula that tightly connects the quality and complexity of the BKZ algorithm. In this work, we use experimental models proposed in [ACF+15, ACD+18] in order to estimate the running time and quality of the output of lattice reduction. They are based on the following two assumptions on the quality and shape of the output of BKZ. The first assumption states that the BKZ algorithm outputs vectors with balanced coordinates, while the second assumption connects the Hermite factor δ with the chosen blocksize β.

Assumption 1. Given as input, a basis B of a d-dimensional lattice Λ, BKZ outputs a vector of norm close to $\delta^d \cdot \det(\Lambda)^{1/d}$ with balanced coordinates. Each coordinate of this vector follows a distribution that can be approximated by a Gaussian with mean 0 and standard deviation $\delta^d \det(\Lambda)^{1/d}/\sqrt{d}$.

Assumption 2. BKZ with blocksize β achieves Hermite factor

$$\delta = \left(\frac{\beta}{2\pi e}(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}}.$$

This assumption is experimentally verified in [Che13].

BKZ cost models. To estimate the running time of BKZ, we use three different models. The first model is an extrapolation by Albrecht [ACF+15] et al. of the Liu–Nguyen datasets [LN13].

According to that model, the logarithm of the running time of BKZ-2.0 (expressed in bit operations) is a quadratic function of $\log(\delta)^{-1}$:

$$\log(T(\mathrm{BKZ}_\delta)) = \frac{0.009}{\log(\delta)^2} - 27.$$

We further refer to this model as the delta-squared model. The model was used in [CGGI17] to estimate the security of TFHE.

Another cost model [ACD+18] assumes that the running time of BKZ with blocksize β for d-dimensional basis is $T(\mathrm{BKZ}_{\beta,d}) = 8d \cdot T(\mathrm{SVP}_\beta)$, where $T(\mathrm{SVP}_\beta)$ is the running time of an SVP oracle in dimension β. For the SVP oracle, we use the following two widely used models:

$$\begin{aligned} \text{Sieving model:} \qquad & T(\mathrm{SVP}_\beta) \approx 2^{0.292\beta+16.4}, \\ \text{Enumeration model:} \qquad & T(\mathrm{SVP}_\beta) \approx 2^{0.187\beta\log(\beta)-1.019\beta+16.1}. \end{aligned}$$

The sieving algorithm [BDGL16] yields around $\left(\frac{4}{3}\right)^{\frac{n}{2}}$ short vectors while solving SVP on an n-dimensional lattice. Therefore, when using the sieving model, we shall assume that one run of the BKZ routine produces $\left(\frac{4}{3}\right)^{\frac{\beta}{2}}$ short lattice vectors, where β is the chosen blocksize. As such, we shall provide the following heuristic, which generalizes the repartition given in Assumption 1 when the number of output vectors is small with regards to the number of possible vectors of desired length:

Assumption 3. Let $R \ll \delta^{d^2} V_d$ and $R \leqslant (4/3)^{\beta/2}$ where V_d is the volume of the ℓ_2 unit ball in dimension d. Given as input, a basis B of a d-dimensional lattice Λ, BKZ_β with a sieving oracle as SVP oracle outputs a set of R vectors of norm close to $\delta^d \cdot \det(\Lambda)^{1/d}$ with balanced coordinates. Each coordinate of these vector follows a distribution that can be approximated by a Gaussian with mean 0 and standard deviation $\delta^d \det(\Lambda)^{1/d}/\sqrt{d}$.

In practice, for the dimension involved in cryptography and for the parameters yields by our techniques, this assumption can be experimentally verified. In a general setting, one might see it as an slight underestimate of the resulting security parameters.

2.4. **Modular Gaussian distribution.** Let $\sigma > 0$. For all $x \in \mathbb{R}$, the density of the centered Gaussian distribution with standard deviation σ is defined as $\rho_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$. We define the distribution that is obtained by sampling a centered Gaussian distribution of standard deviation σ and reducing it modulo 1 as the *modular Gaussian distribution* of parameter σ and denote it as $\mathcal{G}_\sigma$.

The support of the distribution is $\big(-\frac{1}{2};\frac{1}{2}\big)$. The probability density function is given by the absolutely convergent series:

$$g_\sigma(x) = \sum_{k\in\mathbb{Z}} \rho_\sigma(x+k).$$

For large values of σ, the sum that defines the density of a modular Gaussian can be closely approximated.

Lemma 2.2. As $\sigma \to \infty$, $g_\sigma(x) = 1 + 2e^{-2\pi^2\sigma^2}\cos(2\pi x) + O(e^{-8\pi^2\sigma^2})$.

Proof. The Fourier transform of the Gaussian function $\rho_{\sigma,m}(x) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x+m)^2}{2\sigma^2}}$ is given by $\hat{\rho}_{\sigma,m}(y) = e^{-2\pi^2\sigma^2m^2+2\pi imx}$. Then, using the Poisson summation formula, we obtain:

$$\begin{aligned}g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma}\sum_{k\in\mathbb{Z}} e^{-\frac{(k+x)^2}{2\sigma^2}} &= 1 + 2\sum_{k>0} e^{-2\pi^2\sigma^2k^2}\cos(2\pi kx) = \\ &1 + 2e^{-2\pi^2\sigma^2}\cos(2\pi x) + O(e^{-8\pi^2\sigma^2}).\end{aligned} \tag{1}$$

□

2.5. **Probability background.**

Berry-Esseen inequality. The Berry-Esseen inequality shows how closely the distribution of the sum of independent random variables can be approximated by a Gaussian distribution.

Theorem 2.3. Let $X_1,\dots,X_n$ be independent random variables such that for all $i \in \{1,\dots,n\}$ $\mathbb{E}\{X_i\} = 0$, $\mathbb{E}\{X_i^2\} = \sigma_i^2 > 0$, and $\mathbb{E}\{|X_i|^3\} = \rho_i < \infty$. Denote the normalized sum

$$\frac{\sum\limits_{i=1}^{n} X_i}{\sqrt{\sum\limits_{i=1}^{n}\sigma_i^2}}$$

as S_n. Also denote by F_n the cumulative distribution function of S_n, and by Φ the cumulative distribution function of the standard normal distribution. Then, there exists a constant C_0 such that

$$\sup_{x\in\mathbb{R}} |F_n(x) - \Phi(x)| \leqslant C_0 \frac{\sum\limits_{i=1}^{n}\rho_i}{\Big(\sum\limits_{i=1}^{n}\sigma_i^2\Big)^{3/2}}.$$

We use the Berry-Esseen inequality in order to estimate how closely the distribution that we obtain after the lattice reduction step of the dual attack can be approximated by a discrete Gaussian distribution (see Theorem 3.1). The Berry-Esseen inequality requires a finite third absolute moment of the random variables. In the proof of Theorem 3.1, we need the expression of third absolute moment of a Gaussian distribution. It can be obtained from the following lemma.

Lemma 2.4. Let $\sigma > 0$. Let X be a random variable of a Gaussian distribution with mean 0 and standard deviation σ^2. Then, $\mathbb{E}\{|X|^3\} = 2\sqrt{\frac{2}{\pi}}\sigma^3$.

Proof. Classically we have:

$$\mathbb{E}\{|X|^3\} = 2 \cdot \frac{1}{\sqrt{2\pi\sigma}} \int\limits_0^\infty x^3 e^{-\frac{x^2}{2\sigma^2}}\, dx = 2\sqrt{\frac{2}{\pi}}\sigma^3.$$

□

Hoeffding's inequality. Hoeffding's inequality gives an exponentially decreasing upper bound on the probability that the sum of bounded independent random variables deviates from its expectation by a certain amount.

Theorem 2.5. Let $X_1, \ldots, X_N$ be independent random variables such that $a_i \leqslant X_i \leqslant b_i$ for all $i \in \{1, \ldots, N\}$. Denote the average $\frac{1}{N}\sum\limits_{i=1}^{N} X_i$ as $\bar{X}$. Then, for $t > 0$, we have

$$\mathbb{P}\{\bar{X} - \mathbb{E}\{\bar{X}\} \geqslant t\} \leqslant \exp\left(-\frac{2N^2t^2}{\sum\limits_{i=1}^{n}(b_i - a_i)^2}\right), \tag{2}$$

$$\mathbb{P}\{\bar{X} - \mathbb{E}\{\bar{X}\} \leqslant -t\} \leqslant \exp\left(-\frac{2N^2t^2}{\sum\limits_{i=1}^{n}(b_i - a_i)^2}\right). \tag{3}$$

In this paper, we use Hoeffding's inequality to construct a distinguisher for the uniform and the modular Gaussian distributions (see Section 3.2).

3. Dual distinguishing attack against LWE.

In this section, we revisit the distinguishing dual attack against LWE described in [CGGI20]), providing complete proofs and introducing finer tools as a novel distinguisher for the uniform distribution and the modular Gaussian.

Setting. Let s $\in \{0,1\}^n$ be a secret vector and let $\alpha > 0$ be a fixed constant. The attack takes as input m samples $(\mathbf{a}_1, b_1), \ldots, (\mathbf{a}_m, b_m) \in \mathbb{T}^{n+1} \times \mathbb{T}$ which are either all from LWE$_{\mathbf{s},\alpha}$ distribution or all from $\mathcal{U}(\mathbb{T}^n \times \mathbb{T})$, and guesses the input distribution.

We can write input samples in a matrix form:

$$\mathbf{A} := (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in \mathbb{T}^{n\times m}, \quad \mathbf{b} = (b_1, \ldots, b_m)^t \in \mathbb{T}^m,$$

if input samples are from the $\mathrm{LWE}_{\mathbf{s},\alpha}$ distribution:

$$\mathbf{b} = \mathbf{A}^t\mathbf{s} + \mathbf{e} \mod 1.$$

Distinguisher reduction using a small trapdoor. In order to distinguish between the two distributions, the attack searches for a short vector $\mathbf{v} = (v_1, \ldots, v_m)^t \in \mathbb{Z}^m$ such that the linear combination of the left parts of the inputs samples defined by $\mathbf{v}$, i.e.:

$$\mathbf{x} := \sum_{i=1}^{m} v_i\mathbf{a}_i = \mathbf{A}\mathbf{v} \mod 1$$

is also a short vector. If the input was from the LWE distribution, then the corresponding linear combination of the right parts of the input samples is also small as a sum of two relatively small numbers:

$$\mathbf{v}^t\mathbf{b} = \mathbf{v}^t(\mathbf{A}^t\mathbf{s} + \mathbf{e}) = \mathbf{x}^t\mathbf{s} + \mathbf{v}^t\mathbf{e} \mod 1. \tag{4}$$

On the other hand, if the input is uniformly distributed, then independently of the choice of the non-zero vector $\mathbf{v}$, the product $\mathbf{v}\cdot\mathbf{b} \mod 1$ has uniform distribution on $(-1/2; 1/2)$. Recovering a suitable $\mathbf{v}$ thus turns the decisional-LWE problem into an easier problem of distinguishing two distributions on $\mathbb{T}$.

This remaining of section is organized in the following way. First, in Section 3.1 we describe how such a suitable vector $\mathbf{v}$ can be discovered by lattice reduction and analyze the distribution of $\mathbf{v}^t\mathbf{b}$. Then, in Section 3.2, we estimate the complexity of distinguishing two distributions on $\mathbb{T}$ that we obtain after this first part. Eventually Section 3.3 estimates the time complexity of the whole attack.

3.1. **Trapdoor construction by lattice reduction.**

Finding a vector $\mathbf{v}$ such that both parts of the sum (4) are small when the input has LWE distribution is equivalent to finding a short vector in the following $(m+n)$-dimensional lattice:

$$\mathcal{L}(\mathbf{A}) = \left\{ \begin{pmatrix} \mathbf{A}\mathbf{v} \mod 1 \\ \mathbf{v} \end{pmatrix} \in \mathbb{R}^{m+n} \,\middle|\, \forall\, \mathbf{v} \in \mathbb{Z}^m \right\}.$$

The lattice $\mathcal{L}(\mathbf{A})$ can be generated by the columns of the following matrix:

$$\mathbf{B} = \begin{pmatrix} \mathbf{I}_n & \mathbf{A} \\ \mathbf{0}^{m\times n} & \mathbf{I}_m \end{pmatrix} \in \mathbb{R}^{(m+n)\times(m+n)}$$

A short vector in $\mathcal{L}(\mathbf{A})$ can be found by applying a lattice reduction algorithm to the basis $\mathbf{B}$. Using Assumption 1, we expect that the lattice reduction process produces a vector $\mathbf{w} = (\mathbf{x}||\mathbf{v})^t \in \mathbb{Z}^{n+m}$ with equidistributed coordinates. Our goal is to minimize the product $\mathbf{v}^t\mathbf{b} = \mathbf{x}^t\mathbf{s} + \mathbf{v}^t\mathbf{e}$. The vectors $\mathbf{e}$ and $\mathbf{s}$ come from different distributions and have different expected norms. For the TFHE scheme, the variance of $\mathbf{e}$ is much smaller than the variance of $\mathbf{s}$. To take this imbalance into account, one introduces an additional rescaling parameter $q \in \mathbb{R}_{>0}$. The first n rows of the matrix $\mathbf{B}$ are multiplied by q, the last m rows are multiplied by $q^{-n/m}$. Obviously, this transformation doesn't change the determinant of the matrix. A basis $\mathbf{B}_q$ of the transformed lattice is given by

$$\mathbf{B}_q = \begin{pmatrix} q\mathbf{I}_n & q\mathbf{A} \\ \mathbf{0}^{m\times n} & q^{-n/m}\mathbf{I}_m \end{pmatrix} \in \mathbb{R}^{(m+n)\times(m+n)}.$$

We apply a lattice reduction algorithm to $\mathbf{B}_q$. Denote the first vector of the reduced basis as $\mathbf{w}_q$. By taking last m coordinates of $\mathbf{w}_q$ and multiplying them by $q^{n/m}$ we recover the desired vector $\mathbf{v}$. This technique can be thought as a continuous relaxation of the modulus switching technique. That part of the attack is summarized in Algorithm 1.

Algorithm 1: Transform m LWE samples to one sample from modular Gaussian distribution

```
input  : $\mathbf{A} \in \mathbb{T}^{n\times m}$, $\mathbf{b} \in \mathbb{T}^m$, $S > 0$, $\alpha > 0$, $\delta \in (1; 1.1)$
output: $x \in \mathbb{T}$
computeV($\mathbf{A}$, $S$, $\alpha$, $\delta$):
    $q := \left(\frac{S}{\alpha}\right)^{\frac{m}{n+m}}$
    $\mathbf{B}_q := \begin{pmatrix} q\mathbf{I}_n & q\mathbf{A} \\ \mathbf{0}^{m\times n} & q^{-n/m}\mathbf{I}_m \end{pmatrix} \in \mathbb{R}^{(m+n)\times(m+n)}$
    $\mathbf{w}_q \leftarrow \mathrm{BKZ}_\delta(\mathbf{B}_q)$
    $\mathbf{v} := q^{n/m} \cdot (w_{q_{n+1}}, \dots w_{q_{n+m}})^t$
    return $(\mathbf{v})$
LWEtoModGaussian($\mathbf{A}$, $\mathbf{b}$, $S$, $\alpha$, $\delta$):
    $\mathbf{v} \leftarrow \mathrm{COMPUTEV}(\mathbf{A}, S, \alpha, \delta)$
    return $\mathbf{v}^t\mathbf{b} \mod 1$
```

The following lemma describes the distribution of the output of Algorithm 1 under Assumption 1 that BKZ outputs vectors with balanced coordinates.

Lemma 3.1 (see [CGGI20, Section 7])**.** Let $\alpha > 0$ and $S \in (0; 1)$ be fixed constants, $n \in \mathbb{Z}_{>0}$. Let s $\in \{0, 1\}^n$ be a binary vector such that all bits of s are sampled independently from the Bernoulli distribution with parameter S^2: [2] for all $i \in \{1, \ldots, n\}$: $\mathbb{P}\{s_i = 1\} = S^2$, $\mathbb{P}\{s_i = 0\} = 1 - S^2$. Suppose that Assumption 1 holds and let $\delta > 0$ be the quality of the output of the BKZ algorithm. Then, given as input $m = \sqrt{n \cdot \frac{\ln(S/\alpha)}{\ln(\delta)}} - n$ samples from the $\mathrm{LWE}_{\mathbf{s},\alpha}$ distribution, Algorithm 1 outputs a random variable x with distribution that can be approximated by a Gaussian distribution with mean 0 and standard deviation σ

$$\sigma = \alpha \cdot \exp\left(2\sqrt{n \ln(S/\alpha) \ln(\delta)}\right).$$

Denote as F_x the cumulative distribution function of x and denote as Φ_σ the cumulative distribution function of the Gaussian distribution with mean 0 and standard deviation σ. Then, the distance between the two distributions can be bounded:

$$\sup_{t \in \mathbb{R}} |F_x(t) - \Phi_\sigma(t)| = O\Big(\frac{1}{\sqrt{S^2(m+n)}}\Big),$$

as $n \to \infty$.

Theorem 3.1 can be proved using the Berry-Esseen theorem. We give a proof in Appendix A for completeness.

3.2. **Exponential kernel distinguisher for the uniform and the modular Gaussian distributions.** We now describe a novel distinguisher for the uniform and the modular Gaussian distributions. Formally, we construct a procedure which takes as input N samples which are all sampled independently from one of the two distributions and guesses this distribution.

The crux of our method relies on the use of an empirical estimator of the Levy transform of the distributions, to essentially cancel the effect of the modulus 1 on the Gaussian. Namely, from the N samples $X_1, \ldots, X_N$, we construct the estimator $\bar{Y} = \frac{1}{N} \cdot \sum_{i=1}^{N} \mathrm{e}^{2\pi i X_i}$. As N is growing to infinity, this estimator converges to the Levy transform at 0 of the underlying distribution, that is to say:

- to 0 for the uniform distribution
- to $e^{-2\pi^2\sigma^2}$ for the modular Gaussian.

Hence, in order to distinguish the distribution used to draw the samples, we now only need to determine whether the empirical estimator $\bar{Y}$ is closer to 0 or to $e^{-2\pi^2\sigma^2}$.

[2] In TFHE, the parameter of the key distribution S is equal $\frac{1}{\sqrt{2}}$.

Remark 3.2. The optimal value for the corresponding threshold can be obtained as a log-likelihood estimator. However, this optimization is not giving a close formula. It appears that the gains obtained from a numerical optimization of this value are negligible compared to taking the natural threshold of $1/2e^{-2\pi^2\sigma^2}$.

Algorithm 2: Distinguish $\mathcal{U}$ and $\mathcal{G}_\sigma$

```
input  : $X_1, \dots, X_N \in \big(-\frac{1}{2}; \frac{1}{2}\big)$, $\sigma > 0$, sampled independently from $\mathcal{U}$ or $\mathcal{G}_\sigma$
output: A guess: $G$ if the samples are drawn under $\mathcal{G}_\sigma$ or $U$ otherwise

DistinguishGU($X_1, \dots, X_N, \sigma$):
    $\bar{Y} = \frac{1}{N} \cdot \sum_{i=1}^{N} \exp(2\pi i X_i)$
    if ($\bar{Y} \leqslant \frac{1}{2} \cdot e^{-2\pi^2\sigma^2}$) then
        return $U$
    else
        return $G$
```

Lemma 3.3. Let $\sigma > 0$ be a fixed constant. Assume that Algorithm 2 is given as input N points that are sampled independently from the uniform distribution $\mathcal{U}$ or from the modular Gaussian distribution $\mathcal{G}_\sigma$. Then, Algorithm 2 guesses the distribution of the input points correctly with probability at least

$$p_\sigma = 1 - \exp\Big(-\frac{e^{-4\pi^2\sigma^2}}{8} \cdot N\Big). \tag{5}$$

The time complexity of the algorithm is polynomial in the size of the input.

Proof. For all $i \in \{1, \dots, N\}$, denote $e^{2\pi i X_i}$ as Y_i. As $X_i \in \Big(-\frac{1}{2}, \frac{1}{2}\Big)$, $\Re(Y_i) \in (-1; 1]$.

First, we compute the expectation of $\bar{Y} = \frac{1}{N} \cdot \sum_{i=1}^{N} Y_i$ in the two possible cases where X_is are sampled from the uniform distribution, and where X_is are sampled from the modular Gaussian with standard deviation σ. Note that, in both cases, as X_is are sampled independently and identically from the same distribution, $\mathbb{E}\{\bar{Y}\} = \mathbb{E}\{Y_i\}$.

In case of the uniform distribution, the expectation of the real part of $\bar{Y}$ is equal to zero, because the function $\Re(e^{2\pi i x})$ is symmetric around the origin:

$$\mathbb{E}_U\{\Re(\bar{Y})\} = \int_{-1/2}^{1/2} e^{2\pi i x} dx = 0. \tag{6}$$

Now in case of the modular Gaussian distribution, we exploit the 1-periodicity of $t \mapsto e^{2i\pi t}$ to cancel out the modulus 1:

$$\mathbb{E}_G\{\bar{Y}\} = \int_{-1/2}^{+1/2} e^{2\pi i x} \sum_{k \in \mathbb{Z}} \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x+k)^2}{2\sigma^2}} dx \tag{7}$$

$$= \sum_{k \in \mathbb{Z}} \int_{-1/2}^{+1/2} e^{2\pi i x} \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x+k)^2}{2\sigma^2}} dx \tag{8}$$

$$= \int_{-\infty}^{+\infty} e^{2\pi i x} \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{x^2}{2\sigma^2}} dx \tag{9}$$

$$= e^{-2\pi^2\sigma^2} \cdot \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} e^{-\frac{(x-2i\pi\sigma)^2}{2\sigma^2}} dx = e^{-2\pi^2\sigma^2}, \tag{10}$$

the sum-integral exchange being justified by uniform convergence of the sum.

Now, using the expectations of $\bar{Y}$ and the Hoeffding's inequality, we can estimate the probability of getting a correct guess.

First, consider the probability wrongly guessing when the distribution of the input is uniform. By Line 3 of Algorithm 2, it is given by:

$$\mathbb{P}\{G|U\} = \mathbb{P}_U\{\bar{Y} > \frac{1}{2} \cdot e^{-2\pi^2\sigma^2}\}.$$

Since Y_is are bounded, i.e., for all $i \in \{1, \ldots, N\}$, $Y_i \in (-1; 1]$, we can use Hoeffding's inequality (see Theorem 2.5) to bound the probability $\mathbb{P}\{G|U\}$:

$$\mathbb{P}\{G|U\} \leqslant \exp\Big(-\frac{e^{-4\pi^2\sigma^2}}{8} \cdot N\Big). \tag{11}$$

Similarly, we get the same bound on the probability of the wrong guess when the distribution of the input is the modular Gaussian:
$\mathbb{P}\{U|G\} \leqslant \exp\Big(-\frac{e^{-4\pi^2\sigma^2}}{8} \cdot N\Big)$. Together with Equation (11), we get the bound on the probability of the successful guess, given by Equation (5).

Since Algorithm 2 consists of computing the average of the input sample and performing one comparison, it is polynomial in the size of the input. □

Theorem 3.3 implies that in order to distinguish the uniform distribution and the modular Gaussian distribution with the parameter σ with a non-negligible probability, we need to take a sample of size $N = O(e^{4\pi^2\sigma^2})$.

Remark 3.4. The original dual attack, proposed in [CGGI20], does not specify, which algorithm is used for distinguishing the uniform and the modular Gaussian distributions. Instead, to estimate the size of the sample, needed to distinguish the distributions, they estimate the statistical distance ε (see [CGGI20, Section 7, Equation(6)]; for an upper bound on the statistical distance between the modular Gaussian and the uniform distributions, see [BGMRT17, Appendix B]) between the distributions $\mathcal{U}$ and $\mathcal{G}_\sigma$ and use $O(1/\varepsilon^2)$ as an estimate for the required size of the sample. However, such an estimate does not allow a practical instantiation in the security analysis.

It turns out that the exponential kernel distinguisher, described in Algorithm 2, (ignoring some constant factors), has the same complexity as the statistical distance estimate from [CGGI20] suggests, while enjoying a sufficiently precise analysis to provide non-asymptotic parameters estimation.

3.3. **Complexity of the dual attack from TFHE article.** The distinguishing attack is summarized in Algorithm 3. It takes as input $m \times N$ samples from an unknown distribution, then transforms them into N samples which have the uniform distribution if the input of the attack was uniform and the modular Gaussian distribution if the input was from the LWE distribution. Then, the attack guesses the distribution of N samples using Algorithm 2 and outputs the corresponding answer.

Algorithm 3: Dual distinguishing attack (adapted from [CGGI20, Section 7])

```
input : $\{(\mathbf{A}_i, \mathbf{b}_i)\}_{i=1}^{N}$, where $\forall i$ $\mathbf{A}_i \in \mathbb{T}^{n \times m}$, $\mathbf{b}_i \in \mathbb{T}^m$, $\alpha > 0$, $S > 0$, $\delta \in (1; 1.1)$
output: guess for the distribution of the input: Uniform or LWE distribution
DistinguishingAttack($\{\mathbf{A}_i, \mathbf{b}_i\}_{i=0}^{N}$, $\alpha$, $S$, $\delta$):
    $X := \emptyset$
    $\sigma := \alpha \cdot \exp\left(2\sqrt{n \ln(S/\alpha)\ln(\delta)}\right)$
    for $i \in \{1, \ldots, N\}$ do
        $x \leftarrow \mathrm{LWEtoModGaussian}(\mathbf{A}_i, \mathbf{b}_i, S, \alpha, \delta)$
        $X \leftarrow X \cup x$
    if $(\mathrm{DistinguishGU}(X, \sigma) = \mathcal{G})$ then
        return LWE distribution
    else
        return Uniform
```

The following theorem states that the cost of the distinguishing attack can be estimated by solving a minimization problem. It revisits the estimate given in [CGGI20, Section 7].

Theorem 3.5. Let $\alpha > 0$ and $S \in (0; 1)$ be some fixed constants, $n \in \mathbb{Z}_{>0}$. Let $\mathbf{s} \in \{0,1\}^n$ be a binary vector such that all bits of s are sampled independently from a Bernoulli distribution with parameter S^2. Suppose that Assumption 1 holds. Then, the time complexity of solving Decision-LWE$_{\mathbf{s},\alpha}$ with probability of success p by the distinguishing attack described in Algorithm 3 is

$$T_{\text{TFHEattack}} = \min_{\delta} \Big(N(\sigma, p) \cdot T(\text{BKZ}_\delta) \Big), \tag{12}$$

where $\sigma = \alpha \cdot \exp\left(2\sqrt{n \ln(S/\alpha)\ln(\delta)}\right)$, $N(\sigma, p) = 8\ln(\frac{1}{1-p}) \cdot e^{4\pi^2\sigma^2}$.

Proof. The cost of the attack is the cost of the lattice reduction multiplied by the number of samples N needed to distinguish the uniform distribution and the modular Gaussian distribution with the parameter σ:

$$T = N \cdot T(\text{BKZ}_\delta). \tag{13}$$

By Theorem 3.3, Algorithm 2, given as an input a sample of size N, guesses its distribution correctly with the probability at least $1 - \exp\left(-N \cdot \frac{e^{-4\pi^2\sigma^2}}{8}\right)$. Thus, in order to achieve the probability p, we need to produce a sample of size $N(\sigma, p) = 8\ln(\frac{1}{1-p}) \cdot e^{4\pi^2\sigma^2}$.

The parameter σ of the discrete Gaussian distribution as a function of δ can be estimated using Theorem 3.1. Then, the time complexity can be obtained by optimizing the expression, given by Equation (13), as a function of δ. □

4. Our hybrid key recovery attack

In this section, we show how the dual distinguishing attack recalled in Section 3 can be hybridized with exhaustive search on a fraction of the secret vector to obtain a continuum of more efficient key recovery attacks on the underlying LWE problem. Let $\mathbf{s} \in \{0,1\}^n$ be a secret vector and let $\alpha > 0$ be a fixed constant. Our attack takes as input samples from the LWE distribution of form

$$(\mathbf{A}, \mathbf{b} = \mathbf{A}^t\mathbf{s} + \mathbf{e} \mod 1) \in (\mathbb{T}^{n\times m}, \mathbb{T}^m), \tag{14}$$

where $\mathbf{e} \in \mathbb{R}^m$ has centered Gaussian distribution with standard deviation α. The attack divides the secret vector into two fractions:

$$\mathbf{s} = (\mathbf{s}_1 || \mathbf{s}_2)^t, \quad \mathbf{s}_1 \in \{0,1\}^{n_1}, \quad \mathbf{s}_2 \in \{0,1\}^{n_2}, \quad n = n_1 + n_2.$$

The matrix $\mathbf{A}$ is also fractionned into two parts corresponding to the separation of the secret $\mathbf{s}$:

$$\mathbf{A} = \left(\begin{array}{ccc} a_{1,1} & \dots & a_{1,m} \\ \vdots & & \vdots \\ a_{n_1,1} & \dots & a_{n_1,m} \\ \hline a_{n_1+1,1} & \dots & a_{n_1+1,m} \\ \vdots & \dots & \vdots \\ a_{n,1} & \dots & a_{n,m} \end{array}\right) = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{pmatrix} \tag{15}$$

Then, Equation (14) can be rewritten as

$$\mathbf{A}_1^t\mathbf{s}_1 + \mathbf{A}_2^t\mathbf{s}_2 + \mathbf{e} = \mathbf{b} \mod 1.$$

By applying lattice reduction to matrix $\mathbf{A}_1$ as described in Algorithm 1, we recover a vector $\mathbf{v}$ such that $\mathbf{v}^t(\mathbf{A}_1^t\mathbf{s}_1 + \mathbf{e})$ is small and it allows us to transforms m input LWE samples $(\mathbf{A}, \mathbf{b}) \in (\mathbb{T}^{n\times m}, \mathbb{T}^m)$ into one new LWE sample $(\hat{\mathbf{a}}, \hat{b}) \in (\mathbb{T}^{n_2}, \mathbb{T})$ of smaller dimension and bigger noise:

$$\underbrace{\mathbf{v}^t\mathbf{A}_2^t}_{\mathbf{a}}\mathbf{s}_2 + \underbrace{\mathbf{v}^t(\mathbf{A}_1^t\mathbf{s}_1 + \mathbf{e})}_{\hat{e}} = \underbrace{\mathbf{v}^t\mathbf{b}}_{\hat{b}} \mod 1. \tag{16}$$

The resulting LWE sample in smaller dimension can be used to find $\mathbf{s}_2$. Let $\mathbf{x} \in \{0,1\}^{n_2}$ be a guess for $\mathbf{s}_2$. If the guess is correct, then the difference

$$\hat{b} - \hat{\mathbf{a}}^t\mathbf{x} = \hat{b} - \hat{\mathbf{a}}^t\mathbf{s}_2 = (\hat{e} \mod 1) \sim \mathcal{G}_\sigma \tag{17}$$

is small.

If the guess is not correct and $\mathbf{x} \neq \mathbf{s}_2$, then there exist some $\mathbf{y} \neq \mathbf{0}$ such that $\mathbf{x} = \mathbf{s}_2 - \mathbf{y}$. Then, we rewrite $\hat{b} - \hat{\mathbf{a}}^t\mathbf{x}$ in the following way:

$$\hat{b} - \hat{\mathbf{a}}^t\mathbf{x} = (\hat{b} - \hat{\mathbf{a}}^t\mathbf{s}_2) + \hat{\mathbf{a}}^t\mathbf{y} = \hat{\mathbf{a}}^t\mathbf{y} + \hat{e}.$$

We can consider $(\hat{\mathbf{a}}, \hat{\mathbf{a}}^t\mathbf{y} + \hat{e})$ as a sample from the LWE distribution that corresponds to the secret $\mathbf{y}$. Therefore, we may assume that if $\mathbf{x} \neq \mathbf{s}_2$, the distribution of $\hat{b} - \hat{\mathbf{a}}^t\mathbf{x} \mod 1$ is close to uniform, unless the decision-LWE is easy to solve.

In order to recover $\mathbf{s}_2$, the attack generates many LWE samples with reduced dimension. Denote by R the number of generated samples and put them into matrix form as $(\hat{\mathbf{A}}, \hat{\mathbf{b}}) \in \mathbb{T}^{n_2\times R} \times \mathbb{T}^R$. There are 2^{n_2} possible candidates for $\mathbf{s}_2$. For each candidate $\mathbf{x} \in \{0,1\}^{n_2}$, the attack computes an R-dimensional vector $\mathbf{e}_{\mathbf{x}} = \mathbf{b} - \mathbf{A}^t\mathbf{s}$. The complexity of this computation for all the candidates is essentially the complexity of multiplying the matrices $\hat{\mathbf{A}}$ and $\mathbf{S}_2$, where $\mathbf{S}_2$ is a matrix whose columns are all binary vectors of dimension n_2. Naively, the matrix multiplication requires $O(n \cdot 2^{n_2} \cdot R)$ operations. However, by exploiting the recursive structure of $\mathbf{S}_2$, it can be done in time $O(R \cdot 2^{n_2})$.

Then, for each candidate $\mathbf{x}$ for s_2 the attack checks whether the corresponding vector $\mathbf{e}_{\mathbf{x}}$ is uniform or concentrated around zero distribution. The attack returns the only candidate $\mathbf{x}$ whose corresponding vector $\mathbf{e}_{\mathbf{x}}$ has concentrated around zero distribution.

The rest of this section is organized as follows. First, we describe the auxiliary algorithm for multiplying a matrix by the matrix of all binary vectors that let us speed up the search for the second fraction of the secret key. Then, we evaluate the complexity of our attack.

4.1. **Algorithm for computing the product of a matrix with the matrix of all binary vectors.** For any $d \in \mathbb{Z}_{>0}$, define the function $\mathbf{bin}_d : \mathbb{Z} \cap [0; 2^d] \to \{0,1\}^d$ that maps any positive integer $k \leqslant 2^d$ to $\mathbf{bin}_d(k)$ the d-dimensional binary vector obtained from the binary representation of k.

For any positive integer d, denote by $\mathbf{S}_{(d)}$ the matrix of all binary vectors of dimension d, in lexicographic order. Thus, the i-th column of $\mathbf{S}_{(d)}$ is equal to $\mathbf{bin}_d(i)$. These matrices can be constructed recursively. For $d = 1$ it is given by $\mathbf{S}_{(1)} = \begin{pmatrix} 0 & 1 \end{pmatrix}$, and for any $d > 1$ the matrix $\mathbf{S}_{(d)}$ can be constructed by concatenating two copies of the matrix $\mathbf{S}_{(d-1)}$ and adding a row which consists of 2^{d-1} zeros followed by 2^{d-1} ones as the first row to the resulting matrix:

$$\mathbf{S}_{(d)} = \begin{pmatrix} 0\dots 0 & 1\dots 1 \\ \mathbf{S}_{(d-1)} & \mathbf{S}_{(d-1)} \end{pmatrix}. \tag{18}$$

Let $\mathbf{a} = (a_1, \dots, a_d)^t$ be a d-dimensional vector. Our goal is to compute the scalar products of $\mathbf{a}$ with every column of $\mathbf{S}_{(d)}$. We can do it by using the recursive structure of $\mathbf{S}_{(d)}$. Assume that we know the desired scalar products for $\mathbf{a}_{(d-1)} = (a_2, \dots, a_d)^t$ and $\mathbf{S}_{(d-1)}$ Then, using Equation (18), we get

$$\mathbf{a}^t\mathbf{S}_{(d)} = \begin{pmatrix} a_1 & \mathbf{a}^t_{(d-1)} \end{pmatrix} \cdot \begin{pmatrix} 0\dots 0 & 1\dots 1 \\ \mathbf{S}_{(d-1)} & \mathbf{S}_{(d-1)} \end{pmatrix} = \begin{pmatrix} \mathbf{a}^t_{(d-1)}\mathbf{S}_{(d-1)} \\ \begin{pmatrix} a_1 \dots a_1 \end{pmatrix}^t + \mathbf{a}^t_{(d-1)}\mathbf{S}_{(d-1)} \end{pmatrix}, \tag{19}$$

that is, the resulting vector is the sum of the vector $\mathbf{a}^t_{(d-1)}\mathbf{S}_{(d-1)}$ concatenated with itself with the vector whose first 2^{d-1} coordinates are zeros and the last 2^{d-1} coordinates are all equal to a_1. The approach is summarized in Algorithm 4.

Lemma 4.1. Let d be a positive integer number. Algorithm 4, given as input a d-dimensional vector $\mathbf{a}$, outputs the vector $\mathbf{x}$ of dimension 2^d such that for all $i \in \{1, \dots, 2^d\}$ $x_i = \mathbf{a}^t\mathbf{bin}_d(i)$. The time complexity of the algorithm is $O(2^d)$.

Algorithm 4: Compute a scalar product of a vector with all binary vectors

```
input  : $\mathbf{a} = (a_1, \dots, a_d)^t$
output: $\mathbf{a}^t \mathbf{S}_{(d)}$, where $\mathbf{S}_{(d)} \in \{0,1\}^{2^d \times d}$ is the matrix whose columns are all binary
        vectors of dimension $d$ written in the lexicographical order
1 computeScalarProductWithBinaryVectors(a):
2     $\mathbf{x} \leftarrow (0, a_d)^t$
3     for $i \in \{d-1, \dots, 1\}$ do
4         $\mathbf{y} \leftarrow \mathbf{x}$
5         for $j \in \{1, \dots, 2^{d-i}\}$ do
6             $y_j \leftarrow y_j + a_i$
7         $\mathbf{x}' \leftarrow \mathbf{x} \cup \mathbf{y}$
8         $\mathbf{x} \leftarrow \mathbf{x}'$
9     return $\mathbf{x}$
```

Proof. The correctness of the algorithm follows from the recursive structure of the matrix $\mathbf{S}_{(d)}$ (see Equations (18) and (19)). The algorithm performs only additions of some coordinates of the vector $\mathbf{a}$. At the i-th iteration of the cycle (3-8) the algorithm performs 2^{d-i} additions. Number of iterations is $(d-1)$. The overall number of additions is $2 + 2^2 + \cdots + 2^{d-1} = 2^d - 2$. □

Corollary 4.2. Let $\mathbf{A}$ be a matrix with R rows and d columns. The product of $\mathbf{A}$ and $\mathbf{S}_{(d)}$ can be computed in time $O(R \cdot 2^d)$.

Proof. In order to compute $\mathbf{A} \cdot \mathbf{S}_{(d)}$ we need to compute the product of each of the R rows of A with $\mathbf{S}_d$. By Theorem 4.1 it can be done in time $O(2^d)$. Then the overall complexity of multiplying the matrices is $O(R \cdot 2^d)$. □

4.2. **Complexity of the attack.** The pseudo-code corresponding to the full attack is given in Algorithm 5.

Theorem 4.3. Let $\alpha > 0$, $p \in (0;1)$, $S \in (0;1)$, and $n \in \mathbb{Z}_{>0}$ be fixed constants. Let $\mathbf{s} \in \{0,1\}^n$ and $\sigma > 0$. Suppose that Assumption 1 holds. Then, the time complexity of solving the Search-LWE$_{\mathbf{s},\alpha}$ problem with probability of success p by the attack described in Algorithm 5 is

$$T_{\text{attack}} = \min_{\delta, n_2} \left(\left(2^{n_2} + T(\text{BKZ}_\delta)\right) \cdot R(n_2, \sigma, p) \right), \tag{20}$$

where $R(n_2, \sigma, p) = 8 \cdot e^{4\pi^2\sigma^2}(n_2 \ln(2) - \ln(\ln(1/p)))$.

Algorithm 5: Hybrid key recovery attack

```
input : $\{(\mathbf{A}_i, \mathbf{b}_i)\}_{i=1}^{R}$, where $\forall i\ \mathbf{A}_i \in \mathbb{T}^{n\times m}$, $\mathbf{b}_i \in \mathbb{T}^m$, $\alpha > 0$, $S > 0$, $\delta > 1$,
        $n_1 \in \{2, \dots, n-1\}$
output: $\mathbf{s}_2 \in \{0,1\}^{n-n_1}$
recoverS($\{(\mathbf{A}_i, \mathbf{b}_i)\}_{i=1}^{R}$, $\alpha$, $S$, $\delta$, $n_1$):
    $n_2 \leftarrow (n - n_1)$
    $\sigma \leftarrow \alpha \cdot \exp\left(2\sqrt{n_1 \ln(S/\alpha)\ln(\delta)}\right)$
    $\hat{\mathbf{A}} \leftarrow \emptyset$ , $\hat{\mathbf{b}} \leftarrow \emptyset$
    /* lattice reduction part */
    for $i \in \{1, \dots, R\}$ do
        $\mathbf{A} \leftarrow \mathbf{A}_i$, $\mathbf{b} \leftarrow \mathbf{b}_i$
        $(\mathbf{A}_1, \mathbf{A}_2) \leftarrow$ SPLITMATRIX$(\mathbf{A}, n_1)$                ▷ see Equation (15)
        $\mathbf{v} \leftarrow$ COMPUTEV$(\mathbf{A}_1, S, \alpha, \delta)$                      ▷ Algorithm 1
        $\hat{\mathbf{A}} \leftarrow \hat{\mathbf{A}} \cup \{\mathbf{A}_2\mathbf{v}\}$, $\hat{\mathbf{b}} \leftarrow \hat{\mathbf{b}} \cup \{\mathbf{v}^t\mathbf{b}\}$
    /* search for $\mathbf{s}_2$ */
    $\mathbf{S}_{(n_2)} \leftarrow$ matrix of all binary vectors of dimension $n_2$ in lexicographical order
    $\hat{\mathbf{B}} \leftarrow (\hat{\mathbf{b}}, \dots, \hat{\mathbf{b}}) \in \mathbb{T}^{R\times 2^{n_2}}$
    $\hat{\mathbf{E}} \leftarrow \hat{\mathbf{B}} - \hat{\mathbf{A}}^t\mathbf{S}_{(n_2)} \mod 1$            ▷ see Theorem 4.2 and Algorithm 4
    for $i \in \{1, \dots, 2^{n_2}\}$ do
        $\hat{\mathbf{e}} \leftarrow \hat{\mathbf{E}}[i]$
        /* guess the distribution of $e$ (see Algorithm 2) */
        if (DISTINGUISHGU$(\hat{\mathbf{e}}, \sigma) = \mathcal{G}$) then
            return $\mathbf{S}_{(n_2)}[i]$
```

Proof. The attack can be divided in two steps: the lattice reduction step and the exhaustive search for the second fraction of the secret key. The first step of the attack takes $R \times m$ LWE$_{\mathbf{s},\alpha}$ samples and transforms them into R LWE$_{\mathbf{s}_2,\sigma}$ samples such that $\mathbf{s}_2$ is the second fraction of the secret key $\mathbf{s}$ and the noise parameter σ is bigger than the noise parameter α of the input. It takes time $R \cdot T(\text{BKZ}_\delta)$. Denote the matrix form of obtained LWE samples as $(\hat{\mathbf{A}}, \hat{\mathbf{b}}) \in (\mathbb{T}^{n_2\times R}, \mathbb{T}^R)$.

At the search step, the goal is to recover $\mathbf{s}_2$ using the obtained LWE samples. For each of the candidates for $\mathbf{s}_2$ the attack computes the error vector that corresponds to R LWE samples obtained at the previous step. It is equivalent to computing the following matrix expression:

$$\hat{\mathbf{E}} = \hat{\mathbf{B}} - \hat{\mathbf{A}}^t\mathbf{S}_{(n_2)} \mod 1,$$

where $\mathbf{S}_{(n_2)}$ is the matrix composed of all binary vectors of length n_2 written in lexicographic order and $\hat{\mathbf{B}} \in \mathbb{T}^{R \times 2^{n_2}}$ is the matrix formed of 2^{n_2} repetition of the vector $\hat{\mathbf{b}}$. The complexity of computing that expression is dominated by the complexity of computing the product of $\hat{\mathbf{A}}^t \in \mathbb{T}^{R \times n_2}$ and $\mathbf{S}_{(n_2)}$. By Theorem 4.2, it can be computed in $O(R \cdot 2^{n_2})$ operations. Once the attack obtain an error vector for each of the candidates, it guesses the distribution of each error vector using Algorithm 2 and returns the candidate whose error vector has concentrated around zero modular Gaussian distribution.

The time complexity of the attack is the sum of the complexities of the two steps:

$$T_{\text{attack}} = R \cdot \big(2^{n_2} + T(\text{BKZ}_\delta)\big). \tag{21}$$

Now the goal is to evaluate the number of samples R needed to recover $\mathbf{s}_2$ with probability p. By Theorem 3.3, using Algorithm 2, we can guess correctly the distribution of a sample of size R with probability at least $p_\sigma = 1 - \exp\Big(-\frac{e^{-4\pi^2\sigma^2}}{8} \cdot R\Big)$. In order to recover $\mathbf{s}_2$, we need successfully guess the distribution for each of 2^{n_2} candidates. Assume that the distributions, produced by the candidates are independent. Then, the probability to correctly recover $\mathbf{s}_2$ is at least $p_\sigma^{2^{n_2}}$. Thus, to recover $\mathbf{s}_2$ we need to choose the size of the sample R that satisfies:

$$p_\sigma^{2^{n_2}} = \left(1 - \exp\left(-\frac{e^{-4\pi^2\sigma^2}}{8} \cdot R\right)\right)^{2^{n_2}} \geqslant p. \tag{22}$$

Let R be given by the following expression:

$$R = 8 \cdot e^{4\pi^2\sigma^2}(n_2 \ln(2) - \ln(\ln(1/p))). \tag{23}$$

Combining Equations (22) and (23), we obtain:

$$p_\sigma^{2^{n_2}} = \left(1 - \frac{\ln(1/p)}{2^{n_2}}\right)^{2^{n_2}}. \tag{24}$$

Then, when $n_2 \to \infty$, $p_\sigma^{2^{n_2}} \to p$. Thus, the sample size R, given by Equation (23) is sufficient to recover $\mathbf{s}_2$ with the probability p.

By combining Equations (21) and (23) we obtain the time complexity of the attack. □

4.3. **Using sieving in the hybrid attack.** Assume that the BKZ algorithm uses the sieving algorithm (see for instance [BDGL16]) as an SVP oracle. At its penultimate step, the sieving algorithm produces many short vectors, so that by storing this pool of vectors, we may suppose that BKZ produces many short vectors in one run. Thus, if we need N short lattice vectors, we need to run the lattice reduction only $\lceil \frac{N}{m} \rceil$ times, where m is the number of short vectors, returned by the lattice reduction.

In the following corollary from Theorem 4.3, we use this property of the sieving algorithm to revisit the time complexity of our attack under the sieving BKZ cost model.

Corollary 4.4. Let α, p, n, σ and $\mathbf{s} \in \{0;1\}^n$ be as in Theorem 4.3. Assume that the lattice reduction algorithm, used by Algorithm 3, uses the sieving algorithm from [BDGL16] as an oracle for solving SVP. Suppose that Assumption 3 holds. Then, the time complexity of solving the Search-LWE$_{\mathbf{s},\alpha}$ problem with probability of success p by the attack described in Algorithm 5 is

$$T_{\text{attack}} = \min_{\delta, n_2} \left(2^{n_2} \cdot R(n_1, \sigma, p) + T(\text{BKZ}_\delta) \cdot \left\lceil \frac{R(n_2, \sigma, p)}{(4/3)^{\beta/2}} \right\rceil \right), \tag{25}$$

where β is the smallest blocksize such that the lattice reduction with the blocksize β achieves the Hermite factor δ; $R(n_2, \sigma, p)$ is as defined in Theorem 4.3.

Proof. By Theorem 4.3, the time complexity of Algorithm 5 can be seen as the sum of complexities of the two parts of the algorithm. The first part is producing R short lattice vectors and the second part is evaluating R scalar products for each of 2^{n_2} candidates for the secret key. As in the sieving model one run of the lattice reduction produces $(4/3)^{\beta/2}$ short vectors, the first part of Algorithm 5 attack takes time $T(\text{BKZ}_\delta) \cdot \left\lceil \frac{R(n_2,\sigma,p)}{(4/3)^{\beta/2}} \right\rceil$, which implies that the complexity of Algorithm 5 in the sieving BKZ cost model is given by Equation (25). □

Remark on using the sieving model with the attack from Section 3. As the dual attack from [CGGI20] consists in running the BKZ algorithm many times, it can also benefit from using all the vectors, produced by the sieving subroutine. Then, the complexity of the dual attack from [CGGI20] in the sieving model is essentially divided by the number of vectors, produced by the sieving subroutine. See Theorem 4.5 for the complexity of the dual attack under the sieving BKZ cost model.

Corollary 4.5. Let α, S and $\mathbf{s} \in \{0, 1\}$ be as in Theorem 3.5. Suppose that Assumption 1 holds. Assume that the lattice reduction algorithm, used by Algorithm 3, uses the sieving algorithm from [BDGL16] as an oracle for solving SVP. Then, the time complexity of solving Decision-LWE$_{\mathbf{s},\alpha}$ with probability of success p by the distinguishing attack described in Algorithm 3 is given by

$$T_{\text{TFHEattack}} = \min_{\delta} \left(\left\lceil \frac{N(\sigma, p)}{(4/3)^{\beta/2}} \right\rceil \cdot T(\text{BKZ}_\delta) \right), \tag{26}$$

where β is the smallest blocksize such that the lattice reduction with the blocksize β achieves the Hermite factor δ; σ and $N(\sigma, p)$ are as defined in Theorem 3.5.

5. Bit-security estimation and experimental verification

We implement a Python script that, given parameters of an LWE problem and a BKZ cost model as an input, finds optimal parameters for the dual attack (see Section 3) and for our attack (see Section 4). Using this script, we evaluate the computational cost of the dual attack and our attacks for a wide range of LWE parameters and in particular for the parameters used in the TFHE scheme. In this section, we report the results of our numerical estimation and show that the security level of the TFHE scheme should be updated with regard to the hybrid attack. We support our argument by an implementation working on a small example.

5.1. Bit-security of LWE parameters.

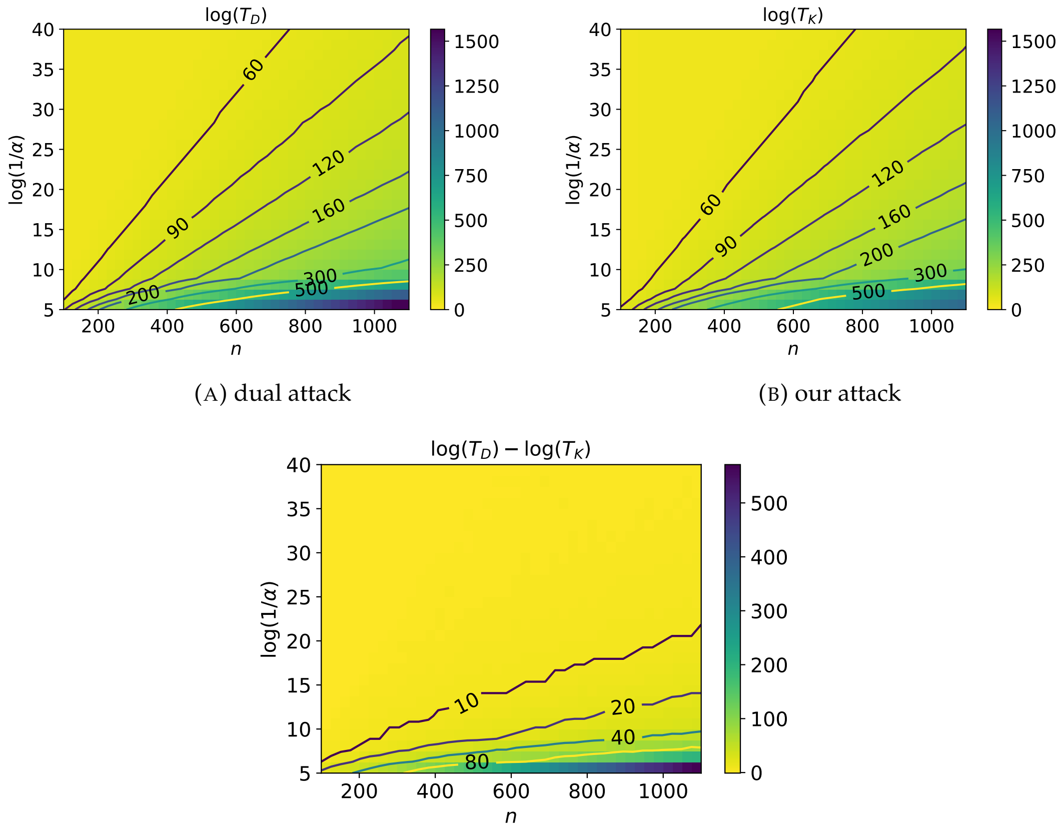
We numerically estimate the cost of solving LWE problem by the dual attack and by our attack for all pairs of parameters (n, α) from the following set: $(n, -\log(\alpha)) \in \{100, 125, \ldots, 1050\} \times \{5, 6.25, \ldots, 38.5\}$. In all cases, we take $S^2 = 1/2$, which corresponds to choosing the secret key uniformly at random from $\{0,1\}^n$ as done in the TFHE scheme. For each attack, we consider three BKZ cost models. For each case, we create a heatmap representing the cost of the attack as a function of parameters n and α. The results obtained using the sieving BKZ cost model are presented in Figure 1. The left heatmap in Figure 1 represents the logarithm of the time complexity of the dual attack, the right heatmap represents the logarithm of the time complexity of our attack. Figure 1 shows that for the same sets of parameters the cost of our attack is always less then or equal to the cost of the dual distinguishing attack and that the difference between the costs of the attacks grows with the hardness of the problem. We obtain similar pictures for the two other considered models. For completeness, we present the heatmaps for the other models in Appendix B.

5.2. Application to the TFHE scheme.

The TFHE scheme uses two sets of parameters: for the switching key and for the bootstrapping key. The security of the scheme is, in fact, defined by the security of the switching key, which is the weaker link.

In [CGGI20], the authors of the TFHE scheme describe two sets of parameters for each of the keys. The first set of the parameters is given by [CGGI20, Table 3] and coincides with the parameters given in the previous papers on TFHE [CGGI16, CGGI17]. The bit-security for the parameters from [CGGI20, Table 3] is estimated according to the dual attack, described in Section 3.

Another set of the parameters is introduced by [CGGI20, Remark 9] and specified in [CGGI20, Table 4]. The security of the updated parameters is evaluated according to the LWE estimator from [ACD+18].



(A) dual attack

(B) our attack

(C) difference

FIGURE 1. Comparison of the costs of the attacks under the enumeration BKZ cost model. Here, n and α denote the dimension and the standard deviation of the noise of LWE samples, T_D denotes the time complexity of the dual distinguishing attack, T_K denotes the time complexity of our key recovery attack.

Also, recently, an update of TFHE's implementation have appeared, introducing a new set of the parameters [G+16, v1.1 – updated security parameters release, date : 2020.02.21]. The security of the parameters from the implementation is also estimated using the LWE estimator from [ACD+18].

For completeness, we re-evaluate the security all the three sets of the parameters under the dual attack as it is described in Section 3 and under our hybrid attack. In Table 2, we we present the results of our estimates for the old parameters (given by [CGGI20, Table 3]), in Table 3, we present the bit-security of the new parameters from [CGGI20, Table 4], and in Table 4, we present the estimates for the updated parameters from the public implementation [G+16].

TABLE 2. Security of the parameters of the TFHE scheme from [CGGI20, Table 3] against dual attack (as described in Section 3) and hybrid dual attack (as described in Section 4). λ denotes security in bits, δ and n_1 are the optimal parameters for the attacks. "-" means that the distinguishing attack doesn't have the parameter n_1.

BKZ model	**switching key** $n = 500, \alpha = 2.43 \cdot 10^{-5}$				**bootstrapping key** $n = 1024, \alpha = 3.73 \cdot 10^{-9}$			
	attack	λ	δ	n_1	attack	λ	δ	n_1
delta-squared	dual	169	1.0052	-	dual	204	1.0046	-
	new attack	**119**	1.0059	406	**new attack**	**160**	1.0051	889
sieving	dual	102	1.0054	-	dual	117	1.0048	-
	new attack	**94**	1.0058	455	**new attack**	**112**	1.005	972
enumeration	dual	195	1.0052	-	dual	230	1.0046	-
	new attack	**137**	1.0062	388	**new attack**	**180**	1.0052	868

In all cases, the cost of our attack is lower than the cost of the dual attack. In addition, the lattice reduction part is always easier for our attack than for the dual attack, because the required quality parameter of lattice reduction δ is always bigger for our attack than for the dual attack. However, the difference of the costs depends on the choice of the model: it is bigger for models that predict higher complexity of BKZ. For example, for the old switching key parameters, the difference under the sieving model is 8 bits while under enumeration model it is 58 bits.

In Figure 8 we present an estimation of the bit-security of the revisited LWE parameters according to the combination of our attack and the collision attack, with time complexity $2^{n/2}$. Thus, Figure 8 represents the function $\min(T_{\text{ourAttack}}(n, \alpha), 2^{n/2})$, where $T_{\text{ourAttack}}(n, \alpha)$ is the cost of our attack for parameters n and α. Figure 8 is obtained under the enumeration BKZ cost model. See Appendix B for other models.

TABLE 3. Security of the parameters of the TFHE scheme from [CGGI20, Table 4] against dual attack (as described in Section 3) and hybrid dual attack (as described in Section 4). λ denotes security in bits, δ and n_1 are the optimal parameters for the attacks. "-" means that the distinguishing attack doesn't have the parameter n_1.

BKZ model	**switching key**				**bootstrapping key**			
	$n = 612, \alpha = 2^{-15}$				$n = 1024, \alpha = 2^{-26}$			
	attack	λ	δ	n_1	attack	λ	δ	n_1
delta-squared	dual	256	1.0043	-	dual	237	1.0043	-
	new attack	**169**	1.0051	474	**new attack**	**179**	1.0049	871
sieving	dual	127	1.0045	-	dual	126	1.0045	-
	new attack	**118**	1.0048	559	**new attack**	**120**	1.0047	970
enumeration	dual	279	1.0043	-	dual	261	1.0043	-
	new attack	**185**	1.0053	457	**new attack**	**179**	1.0049	871

Remark 5.1. The hybrid dual attack presented in Section 4 can be used to estimate security of any LWE-based cryptosystem with binary secrets. For example, it can be applied to get a concrete security estimate for the fully homomorphic encryption scheme FHEW [DM15] which also uses binary secrets. The parameters of the binary-LWE part in FHEW are the following: the dimension $n = 500$, the Gaussian parameter $\sigma = 2^{17}$, and the modulus $q = 2^{32}$. Translating these parameters into TLWE setting, we get $n = 500$ and $\alpha = 2^{-15}$. The bit-security of these parameters under our hybrid dual attack in the sieving model is 96 bits.

5.3. **Comparison with primal uSVP attack.** The security of the recent parameters from TFHE's implementation is evaluated using the LWE estimator from [APS15, ACD$^+$18]. As the results of this estimation suggest, under the sieving BKZ cost model, the best attack against the current parameters of the TFHE scheme among the attacks presented in the LWE estimator is the primal uSVP attack [BG14] (see also [APS15, Section 6.3] for the description of the attack). Therefore, it is interesting to compare our hybrid dual attack with the primal uSVP attack on a wider range of parameters.

TABLE 4. Security of the parameters of the TFHE scheme from the public implementation [G+16] (parameter's update of February 21, 2020) against dual attack (as described in Section 3) and hybrid dual attack (as described in Section 4). λ denotes security in bits, δ and n_1 are the optimal parameters for the attacks. "-" means that the distinguishing attack doesn't have the parameter n_1.

BKZ model	**switching key** $n = 630, \alpha = 2^{-15}$				**bootstrapping key** $n = 1024, \alpha = 2^{-25}$			
	attack	λ	δ	n_1	attack	λ	δ	n_1
delta-squared	dual	270	1.0042	-	dual	256	1.0042	-
	new attack	**176**	1.005	485	**new attack**	**190**	1.0048	862
sieving	dual	131	1.0044	-	dual	131	1.0044	-
	new attack	**121**	1.0047	576	**new attack**	**125**	1.0046	967
enumeration	dual	292	1.0042	-	dual	280	1.0041	-
	new attack	**192**	1.0052	469	**new attack**	**209**	1.0049	842

In order to compare our attack with the primal uSVP attack, we estimate the time complexity of both attacks for each pair of the parameters (n, α) from the following set: $(n, -\log(\alpha)) \in \{200, 250, \ldots, 1450\} \times \{10, 12, \ldots, 48\}$. We evaluate the cost of the primal uSVP attack using the LWE estimator [APS15, ACD+18]. For this comparison, we consider two BKZ cost models: sieving and enumeration. The results of our estimation are presented in Figures 3 and 4.

Figures 3 and 4 show that under both BKZ cost models, it is not so that one attack is better than another for all the sets of the parameters. Under both BKZ cost models, the primal uSVP attack outperforms the hybrid dual attack when dimension is high (i.e., $n > 800$) and the noise parameter is small (i.e., $\alpha < 2^{-35}$). For the rest of the parameters that we consider, the hybrid dual attack outperforms the primal uSVP attack. The difference in the cost of the attacks depends on the chosen BKZ cost model; for the enumeration BKZ cost model the difference between attacks in more significant than for the sieving model.

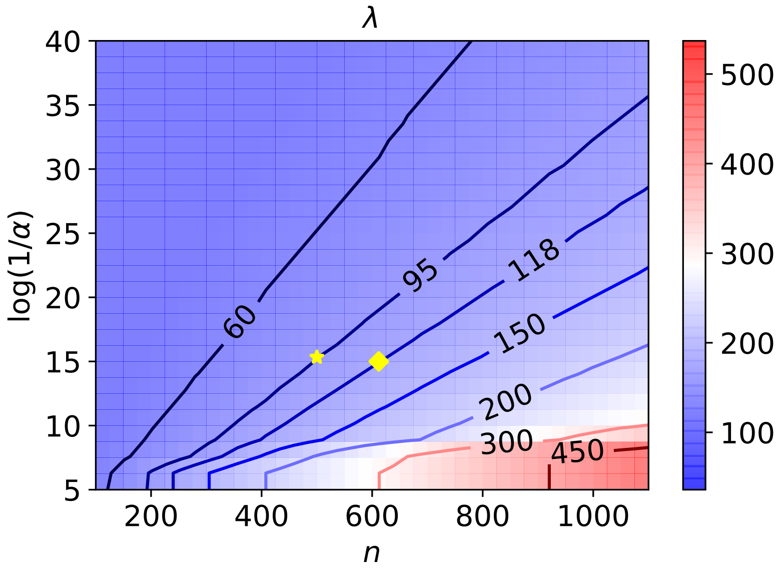


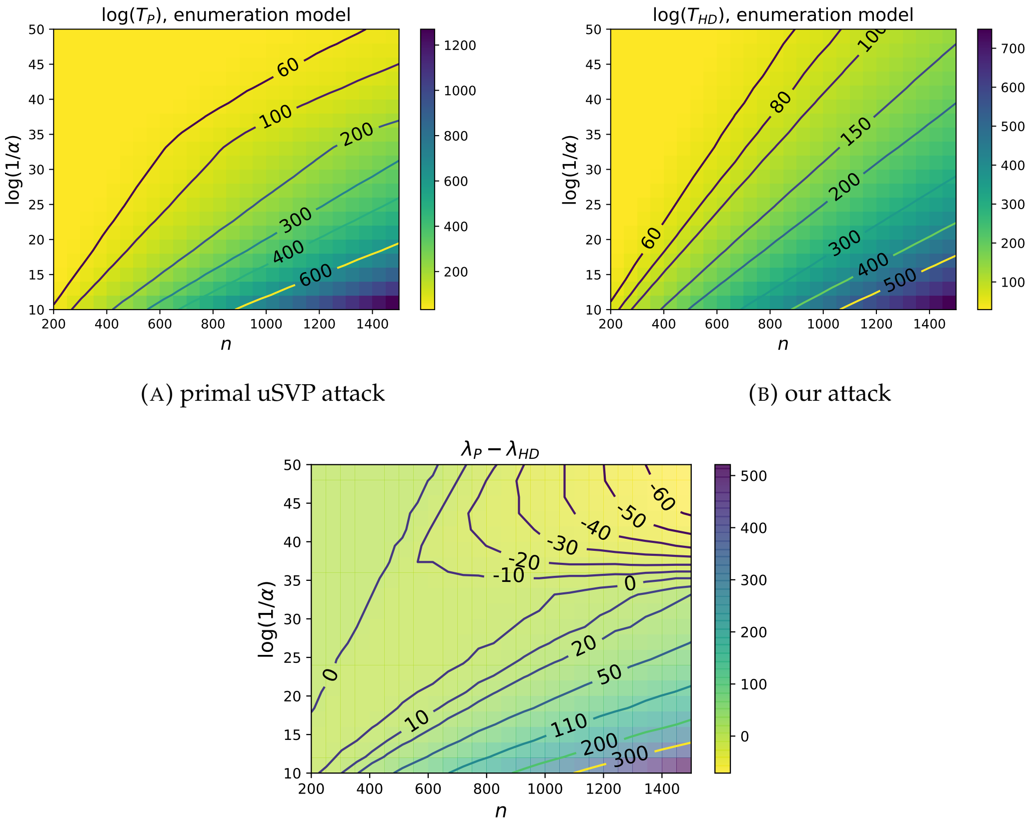
FIGURE 2. Bit-security as a function of the LWE parameters n and α assuming the sieving BKZ cost model. Here, n denotes the dimension, α denotes the standard deviation of the noise. The picture represents the security level λ of LWE samples, $\lambda = \log(\min(T_{\text{ourAttack}}(n,\alpha), 2^{n/2}))$. The numbered lines on the picture represent security levels. The star symbol denotes the key switching parameters from the implementation of the TFHE scheme, the diamond symbol denotes the key switching parameters recommended in [CGGI20, Table 4].

5.4. **Experimental verification.** In order to verify the correctness of our attack, we have implemented it on small examples. Our implementation recovers 5 bits of a secret key for LWE problems with the following two sets of parameters: $(n,\alpha) = (30, 2^{-8})$ and $(n,\alpha) = (50, 2^{-8})$.

For implementation purposes, we rescaled all the elements defined over torus $\mathbb{T}$ to integers modulo 2^{32}. For both examples, we use BKZ with blocksize 20, which yields the quality of the lattice reduction around $\delta \lesssim 1.013$. We computed the values of parameters of the attack required to guess correctly 5 bits of the key with probability 0.99 assuming that quality of the output of BKZ. The required parameters for both experiments are summarized in Table 5.

The first experiment was repeated 20 times, the second was repeated 10 times. For both experiments, the last five bits of the key were successfully recovered at all attempts.

The correctness of both attacks rely on assumptions made in Theorem 3.1 for approximating the distribution of $\mathbf{v}^t(\mathbf{A}^t\mathbf{s}+\mathbf{e}) \mod 1$ by modular Gaussian distribution $\mathcal{G}_\sigma$. In order to verify



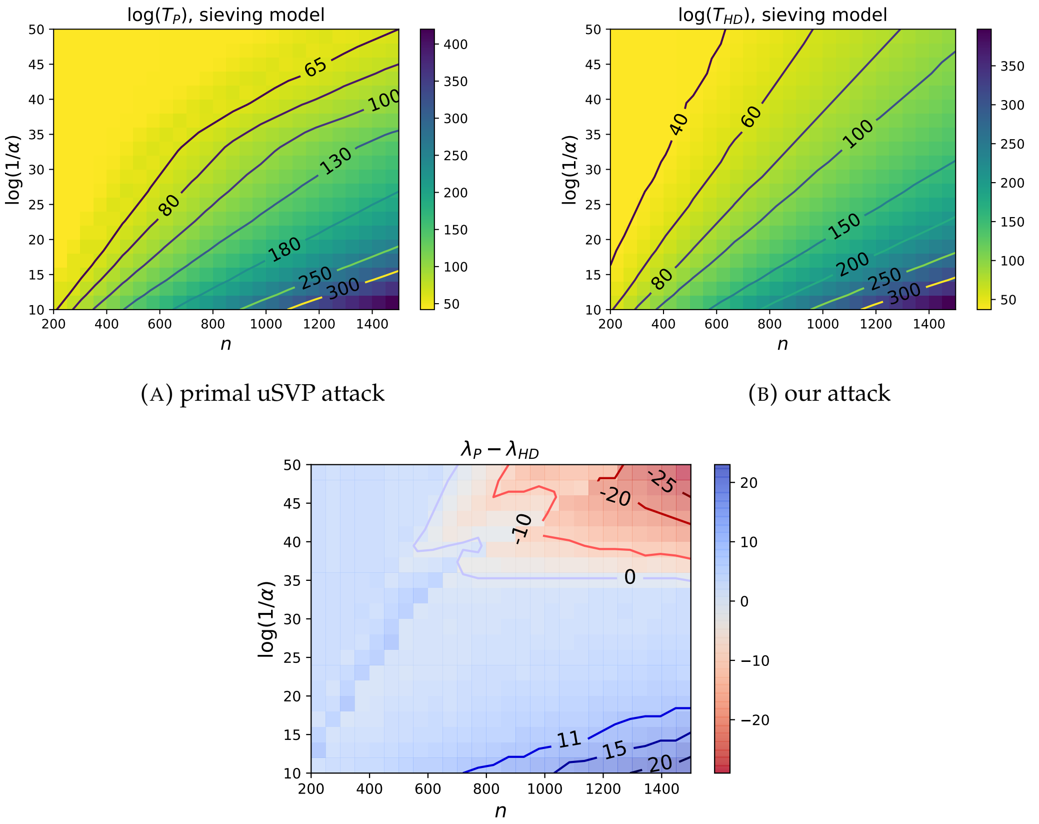
(A) primal uSVP attack

(B) our attack

(C) difference

FIGURE 3. Comparison of the costs of the hybrid dual attack and primal uSVP attack from [BG14] under the enumeration BKZ cost model. Here, n and α denote the dimension and the standard deviation of the noise of LWE samples, T_P denotes the time complexity of the primal uSVP attack, T_{HD} denotes the time complexity of our hybrid dual attack, $\lambda_P - \lambda_{HD} := \log(T_P) - \log(T_{HD})$.

these assumptions, while running both experiments we have collected samples to check the distribution: each time when the attack found correctly the last bits of the secret key $\mathbf{s}_2$, we collected the corresponding $\tilde{e} = \tilde{\mathbf{b}} - \tilde{\mathbf{a}}^t\mathbf{s}_2 = \mathbf{v}^t(\mathbf{A}^t\mathbf{s}_1 + \mathbf{e})$. For the first experiment, the size of the collected sample is $20 \times R_1 = 640$, for the second experiment, it is $10 \times R_2 = 740$. The collected data is presented in Figure 5.



(A) primal uSVP attack

(B) our attack

(C) difference

FIGURE 4. Comparison of the costs of the hybrid dual attack and primal uSVP attack from [BG14] under the sieving BKZ cost model. Here, n and α denote the dimension and the standard deviation of the noise of LWE samples, T_P denotes the time complexity of the primal uSVP attack, T_{HD} denotes the time complexity of our hybrid dual attack, $\lambda_P - \lambda_{HD} := \log(T_P) - \log(T_{HD})$.

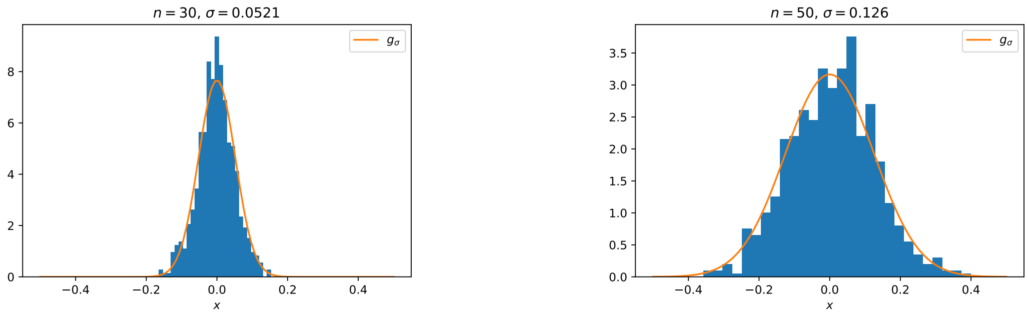
In Table 6, we compare theoretical predictions and estimations obtained from the experiments for the parameters of modular Gaussian distribution $\mathcal{G}_\sigma$. Experimental estimations of mean and variance in both cases match closely theoretical predictions.

6. CONCLUSION

In this work, we demonstrated that the dual lattice attack used to estimate the security of the TFHE scheme can be improved by applying a hybrid approach consisting in a dual attack on a

$(n, -\log(\alpha))$	m	σ	R
(30,8)	76	0.0521	32
(50,8)	90	0.126	74

TABLE 5. Parameters required for guessing 5 bits of the key with $\delta = 1.013$. m is the number of samples needed for one lattice reduction (30), σ is the parameter of modular Gaussian distribution $\mathcal{G}_\sigma$ (Theorem 3.1), R is the number of samples needed to distinguish distributions $\mathcal{G}_\sigma$ and $\mathcal{U}$ (23).



(A)

(B)

FIGURE 5. Distribution of $\tilde{e} = \mathbf{v}^t(\mathbf{A}^t\mathbf{s}_1 + \mathbf{e}) \mod 1$.
Figure 3a represents data from the experiment with parameters $(n, \alpha) = (30, 2^{-8})$, figure 3b – from the experiment with parameters $(n, \alpha) = (50, 2^{-8})$. Blue histograms denote observed data, orange lines – theoretical predictions of the distribution.

projected sublattice, lazy modulus switching, and an efficient batch computation of the leaves of the enumeration tree, performed using a fast matrix multiplication that exploits the recursive structure of the space that we are searching in. This techniques offer an asymptotic speed up and allow to re-evaluate the actual security level of the TFHE scheme, using the most recent estimates and models for lattice reduction costs. Besides, we also show that we can generically leverage the pool of vectors produced by sieving in BKZ-type algorithms to reduce the global complexity of the attack.

TABLE 6. Estimated mean and variance. σ is the parameter of the modular Gaussian distribution $\mathcal{G}_\sigma$, $\mathrm{Var}(\mathcal{G}_\sigma)$ is variance of $\mathcal{G}$

(n, α)	sample size	σ	$\mathrm{Var}(\mathcal{G}_\sigma)$	estimated variance	average of sample
$(30, 2^{-8})$	640	0.0521	0.002714	0.002619	-0.00207
$(50, 2^{-8})$	740	0.126	0.1587	0.14515	0.0064

The obtained asymptotic speed-up is used to re-evaluate the actual security level of the TFHE scheme. We estimated the complexity of the proposed attack under several widely used BKZ cost models. Even if it is still an open question to determine which model gives the most accurate predictions of the behavior of lattice reduction, our results show that the security claim of TFHE is overestimated.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for valuable comments on this work, as well as Paul Kirchner for interesting discussions related to the LWE problem.

REFERENCES

[ACD$^+$18] Martin R Albrecht, Benjamin R Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the {LWE, NTRU} schemes! In *International Conference on Security and Cryptography for Networks*, pages 351–367. Springer, 2018.

[ACF$^+$15] Martin R Albrecht, Carlos Cid, Jean-Charles Faugere, Robert Fitzpatrick, and Ludovic Perret. On the complexity of the BKW algorithm on LWE. *Designs, Codes and Cryptography*, 74(2):325–354, 2015.

[AFG13] Martin R Albrecht, Robert Fitzpatrick, and Florian Göpfert. On the efficacy of solving lwe by reduction to unique-SVP. In *International Conference on Information Security and Cryptology*, pages 293–310. Springer, 2013.

[Ajt98] Miklós Ajtai. The shortest vector problem in l 2 is np-hard for randomized reductions. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 10–19. ACM, 1998.

[Alb17] Martin R Albrecht. On dual lattice attacks against small-secret lwe and parameter choices in HElib and SEAL. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 103–129. Springer, 2017.

[APS15] Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 10–24. Society for Industrial and Applied Mathematics, 2016.

[BG14] Shi Bai and Steven D Galbraith. Lattice decoding attacks on binary lwe. In *Australasian Conference on Information Security and Privacy*, pages 322–337. Springer, 2014.

[BGMRT17] Sauvik Bhattacharya, Oscar Garcia-Morchon, Ronald Rietman, and Ludo Tolhuizen. spkex: An optimized lattice-based key exchange. *IACR Cryptology ePrint Archive*, 2017:709, 2017.

[BGPW16] Johannes Buchmann, Florian Göpfert, Rachel Player, and Thomas Wunderer. On the hardness of LWE with binary error: Revisiting the hybrid lattice-reduction and meet-in-the-middle attack. In *International Conference on Cryptology in Africa*, pages 24–43. Springer, 2016.

[BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):13, 2014.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

[BLP^{+}13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584. ACM, 2013.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Annual cryptology conference*, pages 505–524. Springer, 2011.

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachene. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *international conference on the theory and application of cryptology and information security*, pages 3–33. Springer, 2016.

[CGGI17] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster packed homomorphic operations and efficient circuit bootstrapping for tfhe. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 377–408. Springer, 2017.

[CGGI20] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Tfhe: fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91, 2020.

[Che13] Yuanmi Chen. *Réduction de réseau et sécurité concrete du chiffrement completement homomorphe*. PhD thesis, Paris 7, 2013.

[CHHS19] Jung Hee Cheon, Minki Hhan, Seungwan Hong, and Yongha Son. A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret lwe. *IEEE Access*, 7:89497–89506, 2019.

[CLP17] Hao Chen, Kim Laine, and Rachel Player. Simple encrypted arithmetic library-seal v2. 1. In *International Conference on Financial Cryptography and Data Security*, pages 3–18. Springer, 2017.

[CN11] Yuanmi Chen and Phong Q Nguyen. BKZ 2.0: Better lattice security estimates. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 1–20. Springer, 2011.

[CS15] Jung Hee Cheon and Damien Stehlé. Fully homomophic encryption over the integers revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 513–536. Springer, 2015.

[DM15] Léo Ducas and Daniele Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 617–640. Springer, 2015.

[FV12] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

[G$^+$09] Craig Gentry et al. Fully homomorphic encryption using ideal lattices. In *Stoc*, volume 9, pages 169–178, 2009.

[G$^+$16] Nicolas Gama et al. Github repository. TFHE: Fast fully homomorphic encryption library over the torus. `https://github.com/tfhe/tfhe`, 2016.

[GNR10] Nicolas Gama, Phong Q Nguyen, and Oded Regev. Lattice enumeration using extreme pruning. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 257–278. Springer, 2010.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Annual Cryptology Conference*, pages 75–92. Springer, 2013.

[HG07] Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In *Annual International Cryptology Conference*, pages 150–169. Springer, 2007.

[HPS11] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In *Annual Cryptology Conference*, pages 447–464. Springer, 2011.

[HS15] Shai Halevi and Victor Shoup. Bootstrapping for helib. In *Annual International conference on the theory and applications of cryptographic techniques*, pages 641–670. Springer, 2015.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[LN13] Mingjie Liu and Phong Q Nguyen. Solving BDD by enumeration: An update. In *Cryptographers' Track at the RSA Conference*, pages 293–309. Springer, 2013.

[LP11] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In *Cryptographers' Track at the RSA Conference*, pages 319–339. Springer, 2011.

[MW16] Daniele Micciancio and Michael Walter. Practical, predictable lattice basis reduction. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 820–849. Springer, 2016.

[Reg05] O Regev. On lattices, learning with errors, random linear codes, and cryptography, 2005. In *STOC*, pages 84–93. ACM, 2005.

[SC19] Yongha Son and Jung Hee Cheon. Revisiting the hybrid attack on sparse and ternary secret LWE. *IACR Cryptology ePrint Archive*, 2019:1019, 2019.

[Wun16] Thomas Wunderer. Revisiting the hybrid attack: Improved analysis and refined security estimates. *IACR Cryptology ePrint Archive*, 2016:733, 2016.

APPENDIX A. PROOF OF THEOREM 3.1.

Proof. Under Assumption 1, the coordinates of $\mathbf{w}_q$ are independent and distributed according to the Gaussian distribution with expectation 0 and standard deviation $\delta^{n+m}/\sqrt{n+m}$. Since $\mathbf{w}_q = (q \cdot \mathbf{x} \,||\, q^{-n/m} \cdot \mathbf{v})^t$, the coordinates of vectors $\mathbf{x}$ and $\mathbf{v}$ also have centered Gaussian distribution, but with different standard deviations. Let

$$\sigma_{\mathbf{x}} = \frac{1}{q} \cdot \frac{\delta^{m+n}}{\sqrt{m+n}} \quad \text{and} \quad \sigma_{\mathbf{v}} = q^{n/m} \cdot \frac{\delta^{m+n}}{\sqrt{m+n}}$$

be the standard deviation of coordinates of $\mathbf{x}$ and of $\mathbf{v}$ correspondingly. Consider the distribution of

$$\mathbf{v}^t\mathbf{b} = \mathbf{x}^t\mathbf{s} + \mathbf{v}^t\mathbf{e} = \sum_{i=1}^{n} x_i \cdot s_i + \sum_{i=1}^{m} v_i \cdot e_i.$$

$\mathbf{v}^t\mathbf{b}$ is a sum of $m+n$ independent random variables and, therefore, its distribution can be approximated by a Gaussian distribution according to the Central Limit Theorem. In order to learn the parameters of the Gaussian, we need to obtain the expectations and variances of $x_1 \cdot s_1$ and $v_1 \cdot e_1$.

First, consider the distribution of $x_1 \cdot s_1$. As s_1 has a Bernoulli distribution with parameter S^2, $x_1 s_1$ is a random variable from the distribution that can be obtained by sampling 0 with probability S^2 and sampling from a Gaussian distribution with mean 0 and variance $\sigma_{\mathbf{x}}^2$ with probability $1 - S^2$. Therefore, $\mathbb{E}(x_1 \cdot s_1) = 0$ and $\mathrm{Var}(x_i \cdot s_i) = S^2\sigma_{\mathbf{x}}^2$.

Then, consider $v_1 e_1$. As $\mathbf{v}$ and $\mathbf{e}$ are independent and $\mathbb{E}(v_1) = \mathbb{E}(e_1) = 0$, $\mathbb{E}(v_1 e_1) = \mathbb{E}(v_1)\mathbb{E}(e_1) = 0$ and $\mathrm{Var}(v_1 e_1) = \mathrm{Var}(v_1) \cdot \mathrm{Var}(e_1) = \alpha^2\sigma_{\mathbf{v}}^2$.

Thus, the distribution of $\mathbf{v}^t\mathbf{b}$ is close to the Gaussian distribution with expectation 0 and variance

$$\sigma^2 = n\,\mathrm{Var}(x_1 s_1) + m\,\mathrm{Var}(v_1 e_1) = nS^2\sigma_{\mathbf{x}}^2 + m\alpha^2\sigma_{\mathbf{v}}^2 = \frac{\delta^{2(m+n)}}{m+n}\Big(\frac{nS^2}{q^2} + m\alpha^2 q^{2n/m}\Big). \tag{27}$$

Our goal is to obtain a distribution that is as concentrated around zero as possible. Hence we choose parameters m and q in order to minimize variance of $\mathbf{v}^t\mathbf{b}$.

First, we find the optimal value of q by differentiation of Equation (27) :

$$\frac{\partial\sigma^2}{\partial q} = \frac{\delta^{2(m+n)}}{m+n} \cdot \Big(-\frac{2nS^2}{q^3} + \frac{2n}{m} \cdot m\alpha^2 q^{\frac{2n}{m}-1}\Big) = 0 \quad \rightarrow \quad q_{\text{opt}} = \Big(\frac{S}{\alpha}\Big)^{\frac{m}{m+n}}.$$

After replacing q by q_{opt} in Equation (27) we obtain:

$$\sigma^2 = \left(S\delta^{m+n}\Big(\frac{\alpha}{S}\Big)^{\frac{m}{m+n}}\right)^2. \tag{28}$$

Also, for $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{v}}$ we obtain the following relation

$$\frac{\sigma_{\mathbf{x}}}{\sigma_{\mathbf{v}}} = \frac{q^{-n/m}}{q} = \frac{\alpha}{S}. \tag{29}$$

Then, we find the optimal value of m by differentiating $\ln(\sigma)$:

$$\frac{\partial \ln(\sigma)}{\partial m} = \ln(\delta) + n \ln\left(\frac{\alpha}{S}\right) \cdot \frac{1}{(m+n)^2} = 0 \quad \rightarrow \quad m_{\text{opt}} = \sqrt{n \cdot \frac{\ln(S/\alpha)}{\ln(\delta)}} - n \tag{30}$$

Now, replacing m by m_{opt} in Equation (28), we find:

$$\sigma(\delta, n, S, \alpha) = \sigma(\hat{m}, \delta, n, S, \alpha) = \alpha \cdot \exp\left(2\sqrt{n \ln(S/\alpha)\ln(\delta)}\right).$$

The distance between the distribution of $\mathbf{v}^t\mathbf{b}$ and the Gaussian distribution with mean 0 and variance σ^2 can be estimated by the Berry-Esseen inequality (see Theorem 2.3). To use this inequality, we need to compute the third absolute moments of $x_1 s_1$ and $v_1 e_1$.

We start with $x_1 s_1$. As x_1 and s_1 are independent,

$$\mathbb{E}\{|x_1 s_1|^3\} = \mathbb{E}\{|x_1|^3\}\mathbb{E}\{|s_1|^3\}.$$

By Theorem 2.4, $\mathbb{E}\{|x_1|^3\} = 2\sqrt{2/\pi}\sigma_x^3$. As s_1 has the Bernoulli distribution with parameter S^2, $\mathbb{E}\{|s_1|^3\} = \mathbb{E}\{s_1\} = S^2$. Putting two parts together, we get

$$\rho_{x_1 s_1} = \mathbb{E}\{|x_1 s_1|^3\} = 2\sqrt{2/\pi}S^2\sigma_{\mathbf{x}}^3. \tag{31}$$

In the same way, we obtain

$$\rho_{v_1 e_1}\mathbb{E}\{|v_1 e_1|^3\} = \frac{8}{\pi}\alpha^3\sigma_{\mathbf{v}}^3. \tag{32}$$

Denote the cumulative distribution function of $\mathbf{v}^t\mathbf{b}$ by $F_{\mathbf{v}^t\mathbf{b}}$, and denote the cumulative distribution function of the Gaussian distribution with mean 0 and variance σ^2 by Φ_σ. By the Berry-Esseen inequality, there exists a constant C_0 such that

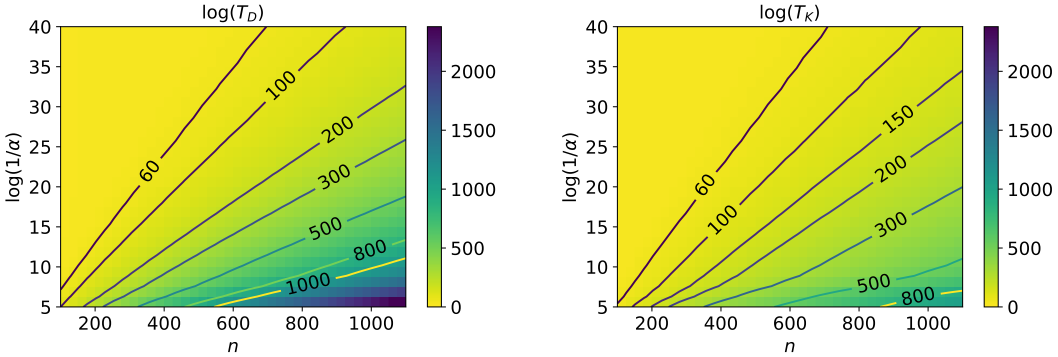
$$\sup_{x\in\mathbb{R}} |F_{\mathbf{v}^t\mathbf{b}}(x) - \Phi_\sigma(x)| \leqslant C_0 \cdot \frac{n\rho_{x_1 s_1} + m\rho_{v_1 e_1}}{(nS^2\sigma_{\mathbf{x}}^2 + m\alpha^2\sigma_{\mathbf{v}}^2)^{3/2}}. \tag{33}$$

Then, using Equations (29) and (31) to (33), for the distance between the distributions we get:

$$\sup_{x\in\mathbb{R}} |F_{\mathbf{v}^t\mathbf{b}}(x) - \Phi_\sigma(x)| \leqslant C_0\sqrt{\frac{8}{S^2\pi}} \cdot \frac{n + mS\sqrt{8/\pi}}{(m+n)^{3/2}} \leqslant C_0 \cdot \frac{8}{\pi S} \cdot \frac{1}{\sqrt{m+n}}. \tag{34}$$

□

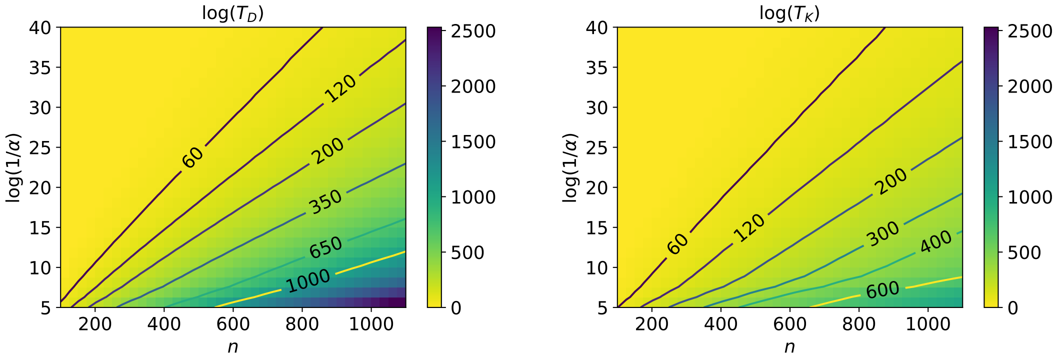
Appendix B. Heatmaps for the enumeration and delta-squared BKZ cost models



(A) dual attack

(B) our attack

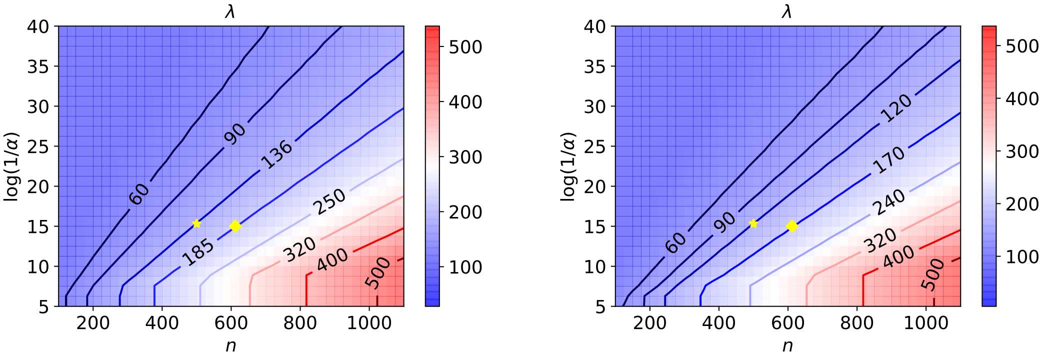
FIGURE 6. Comparison of the costs of the attacks under the enumeration BKZ cost model.



(A) dual attack

(B) our attack

FIGURE 7. Comparison of the costs of the attacks under the delta-squared BKZ cost model.



(A) enumeration

(B) delta-squared

FIGURE 8. Bit-security as a function of LWE parameters n and α under the sieving and delta-squared BKZ cost models.