A Refined Hardness Estimation of LWE in Two-step Mode

Wenwen Xia[1,3], Leizhang Wang[1], Geng Wang[2,3*], Dawu Gu[1,2*], and Baocang Wang[1]

[1] Xidian University
{xiawenwen, lzwang_2}@stu.xidian.edu.cn, bcwang@xidian.edu.cn
[2] Shanghai Jiao Tong University
{wanggxx, dwgu}@sjtu.edu.cn
[3] State Key Laboratory of Cryptology, P.O.Box 5159, Beijing, 100878, China

Abstract. Recently, researchers have proposed many LWE estimators, such as lattice-estimator (Albrecht et al, Asiacrypt 2017) and leaky-LWE-Estimator (Dachman-Soled et al, Crypto 2020), while the latter has already been used in estimating the security level of Kyber and Dilithium using only BKZ. However, we prove in this paper that solving LWE by combining a lattice reduction step (by LLL or BKZ) and a target vector searching step (by enumeration or sieving), which we call a Two-step mode, is more efficient than using only BKZ.

Moreover, we give a refined LWE estimator in Two-step mode by analyzing the relationship between the probability distribution of the target vector and the solving success rate in a Two-step mode LWE solving algorithm. While the latest Two-step estimator for LWE, which is the "`primal-bdd`" mode in lattice-estimator[1], does not take into account some up-to-date results and lacks a thorough theoretical analysis. Under the same gate-count model, our estimation for NIST PQC standards drops by 2.1∼3.4 bits (2.2∼4.6 bits while considering more flexible blocksize and jump strategy) compared with leaky-LWE-Estimator.

Furthermore, we also give a conservative estimation for LWE from the Two-step solving algorithm. Compared with the Core-SVP model, which is used in previous conservative estimations, our estimation relies on weaker assumptions and outputs higher evaluation results than the Core-SVP model. For NIST PQC standards, our conservative estimation is 4.17∼8.11 bits higher than the Core-SVP estimation. Hence our estimator can give a closer estimation for both upper bound and lower bound of LWE hardness.

Keywords: Lattice-based Cryptanalysis, Security Strength, LWE estimator, Two-step mode.

[*] Corresponding author.

[1] https://github.com/malb/lattice-estimator

1 Introduction

As an important branch in post-quantum cryptography, lattice-based cryptography has shown its potential in several cryptographic primitives, such as key establishment [1], digital signature [2,3], hash function [4] and other more advanced cryptography constructions like identity-based encryption [5], attribute-based encryption [6], functional encryption [7], and homomorphic encryption [8].

One of the advantages of lattice-based cryptography is that the security of lattice-based cryptography schemes is guaranteed by the hardness of lattice problems with worst-case to average-case reduction, such as the Learning with Errors problem (LWE). It has been proved that solving the LWE problem is at least as hard as some worst-case lattice problem like the Shortest Independent Vector problem (SIVP) or the Bound Distance Decoding (BDD) problem. In the post-quantum standardization process held by the National Institute of Standards and technique (NIST), many lattice-based cryptographic schemes (e.g. [1–3]) are selected as standards to resist the threat of quantum computer. One of the most important problems in standardization is the parameter selection. To select more compact but still safe security parameters for lattice-based schemes, it is necessary to give a concrete hardness estimation for lattice-based problems. In this paper, we focus on LWE, which is the most widely used lattice-based problem.

There are various methods for solving LWE, such as BDD attack [9], Arora–Ge attack [10], BKW attack [11], primal attack [12,13], dual attack [14] and hybrid attacks [15] based on lattice reduction algorithm. Among them, the primal attack [12,13] is most practical in breaking actual LWE-based schemes, and the concrete hardness of LWE is often estimated by calculating the cost of the primal attack. A primal attack translates LWE to a unique Shortest Vector Problem (uSVP) by constructing a special lattice basis with Kannan's embedding technique [16].

In particular, a long series of works, e.g. [12, 13, 17–20] have proposed the evaluation of the hardness of LWE under the primal attack. In 2015, the work of Albrecht et al. [17] gives concrete estimations for various families of LWE instances. Later, a simple yet conservative estimation method was given by [12] named the Core-SVP model. It proposed a success condition in solving LWE by BKZ with fixed blocksize β and estimated its cost as a single call to the SVP oracle, which is a lattice sieve with dimension β. Since the Core-SVP model ignores both the number of calls to the SVP oracle in one BKZ tour and the number of BKZ tours, the evaluation result by the Core-SVP model is often considered to be conservative enough. In 2017, Albrecht et al. [18] verified the attack success condition proposed in [12] by experiments.

However, the experiment results shown in [18] and [21] both illustrate that when the blocksize of BKZ is smaller than the estimation given in [12], it still has a non-negligible probability in solving the LWE instance. This phenomenon is mainly caused by the randomness of the target vector which actually follows the discrete Gaussian distribution rather than a fixed expected value. To solve this problem, Dachman-Soled et al. [20] proposed the first estimator which describes the relationship between the probability of successfully solving LWE and the blocksize β of BKZ used in solving LWE, which is called "leaky-LWE-Estimator". According to the experiment results of [22], the estimator proposed in [20] and their simplified version [22] can well predict the behavior of BKZ solving LWE with smaller blocksize β. In fact, the leaky-LWE-Estimator has been used for estimating the concrete security strength of the lattice-based post-quantum cryptography (PQC) standardization [1, 2] selected by NIST [23] in 2022.

Specifically, the leaky-LWE-Estimator first uses the technique in [20] to calculate the expected value of BKZ blocksize of solving LWE and calculate the total number of logic circuit gates needed to solve LWE by calling the gate-count algorithm proposed in [24]. It is noticeable that the leaky-LWE-Estimator also considers the influence of dimension-for-free (d4f) technique [25], which leads to a decrease in the estimation result. Moreover, it is worth pointing out that the Core-SVP model did not consider the influence brought by d4f, which threatens the conservativeness of the Core-SVP model.

However, the main problem in the leaky-LWE-Estimator and Core-SVP model is that they only use the BKZ algorithm as the underlying LWE solver, instead of combining BKZ reduction with a final search step (we call it a Two-step mode for solving LWE). In this work, we prove that the Two-step LWE-solving strategy is more efficient than the underlying LWE solver in earlier LWE estimators (such as leaky-LWE-Estimator [20]) which only uses BKZ, thus the BKZ-only estimators may output an over-optimistic estimation.

A Two-step LWE-solving strategy is divided into a lattice reduction step (by LLL or BKZ) and a target vector searching step (by enumeration or sieving). Although the Two-step mode is often considered a folklore approach to solving LWE, only few works bring it into practice. The first Two-step LWE-solving attack was proposed in [9], where they reduce LWE into a BDD problem, and call an enumeration for finding the closest vector in the last searching step. In [26], the authors show that an additional post-processing step using enumeration can increase the success rate in solving γ-SVP with $\gamma = 1.05$, but it is not known whether the post-processing step has same acceleration when applied to LWE. For solving LWE with sieving instead of enumeration, the G6K framework [27] presented a solving algorithm that is also a combination of BKZ and conditional sieving. However, it is different from the Two-step strategy in previous works, and its efficiency has not been theoretically analyzed.

In the context of LWE estimation, Albrecht et al use the "`primal_bdd`" function in lattice-estimator [28] to estimate the hardness of LWE through a primal attack using one BKZ reduction and a sieve in the searching step. However, in estimating the dimension of the last sieving, "`primal_bdd`" estimation only considers the expected norm of the target vector rather than analyzing the relationship between the probability distribution of the target vector and the solving success rate of Two-step mode. So it is necessary to give a more refined Two-step LWE estimator that considers the success probability of the last sieve algorithm and provides extensive experimental evidence of its accuracy. Besides, the Two-step attack proposed in [29] can also improve the efficiency of the Two-step attack in "`primal_bdd`" by applying the improved progressive BKZ reduction and allowing PnjBKZ with jump value > 1.

Furthermore, there is an open question proposed in Section 5.3 (Q7) of Kyber's document that a security estimation error interval exists in NIST lattice-based standardization. This security estimation error interval is caused by using different reduction strategies to evaluate the security. Particularly, the reduction strategy considered by leaky-LWE-Estimator [20] is a trivial progressive BKZ, and in [12, 17–19] they consider a fixed blocksize BKZ algorithm to solve LWE. The paper [30] mentioned that a large dimension of sieve in the final process costs less than a BKZ. The trivial reduction strategies above can be further improved by a more efficient reduction strategy like the optimized blocksize and jump selection strategy proposed in [29] which has already shown its efficiency in solving LWE instances [2]. To ensure the security and narrow the security estimation error interval of lattice-based NIST standard schemes, it is necessary to evaluate the impact of the combination of the Two-step solving strategy and the optimized blocksize and jump selection strategy on the security of NIST selected lattice-based schemes.

Contributions. In this paper, we improve the estimation of LWE hardness from the following aspects:

- We formally prove that the Two-step mode is more efficient in solving uSVP than the BKZ-only mode under Geometric Series Assumption, and extend the result to solving LWE which considers the distribution of LWE error term.

- We construct an LWE hardness estimator which underlying LWE solver is the Two-step LWE solving algorithm, and we calculate the success probability for solving LWE at each step. In the reduction phase, we give a heuristic assumption that each BKZ tour totally randomizes the lattice basis, which is also implicitly implied by the leaky-LWE-estimator [20], so that the success probability of different BKZ tours can be considered independent. In the searching phase, however, the success probability is accumulated after each step. By calculation of the success rate, we show that the expected cost for solving LWE by Two-step mode is much lower than by BKZ-only mode as in [20].

- To verify the accuracy of our estimation, we did extensive experiments of solving LWE by different sieving dimensions in the searching phase. The results of these experiments are consistent with our estimation, which means the expected time cost of solving LWE by our estimator is accurate. Moreover, we re-evaluate the security bit of NIST PQC schemes by our Two-step LWE hardness estimator. When using the same trivial reduction strategy in leaky-LWE-Estimator [20], the security bit drops by 2.1∼3.4 bits. Besides, when using the optimized blocksize and jump selection strategy proposed in paper [29], the security bit drops by 2.2∼4.6 bits.

[2] See latest TU Darmstadt LWE challenge records $(n, \alpha) \in \{(40, 0.035), (90, 0.005), (50, 0.025), (55, 0.020), (40, 0.040)\}$ in `https://www.latticechallenge.org/lwe_challenge/challenge.php`.

- We also give a more accurate lower bound estimation which is a conservative Two-step solving mode estimation for LWE. Compared with the commonly used Core-SVP model, our conservative estimation relies on weaker assumptions. Meantime, our conservative estimation has higher estimation results than the Core-SVP model (while d4f not considered). For NIST PQC standards, our conservative estimation is 4.17$\sim$8.11 bits higher than the Core-SVP estimation. Therefore, we give more accurate estimations on both the upper bound and lower bound of the hardness of LWE.

All detailed codes of our Two-step LWE Estimator with different reduction strategies are already open-sourced[3].

Organization In Section 2 we give the preliminaries, notations, and the basic knowledge of lattice problems. In Section 3 we prove that the Two-step solving mode is more efficient in solving uSVP than the BKZ-only mode. In Section 4 we construct a refined Two-step security estimator for solving LWE. The experiments results in Section 5 verify the accuracy of our Two-step security estimator and the efficiency of the Two-step solving mode. In Section 6 we give a conservative estimation for LWE from a Two-step solving algorithm. Based on our Two-step security estimator and lower bound estimation estimator we give more accurate both upper bound and lower bound estimation of LWE in NIST PQC schemes in Section 7.

2 Preliminaries

2.1 Notations and Basic Definitions

In this paper, all vectors are denoted by bold lowercase letters and are to be read as column vectors. We write a matrix $\mathbf{B}$ as $\mathbf{B} = (\mathbf{b}_0, \cdots, \mathbf{b}_{d-1})$ where $\mathbf{b}_i$ is the $(i+1)$-th column vector of $\mathbf{B}$. The Euclidean norm of a vector $\mathbf{v}$ is denoted by $\|\mathbf{v}\|$. A lattice $\mathcal{L}$ generated by the basis $\mathbf{B}$ is denoted by $\mathcal{L}(\mathbf{B}) = \{\mathbf{B}\mathbf{x} | \mathbf{x} \in \mathbb{Z}^d\}$. Here lattice basis matrix $\mathbf{B} \in \mathbb{R}^{d \times d}$ needs to be full rank d. We denote $\mathbf{B}^* = (\mathbf{b}_0^*, \cdots, \mathbf{b}_{d-1}^*)$ as the Gram-Schmidt orthogonalization of $\mathbf{B}$, in which $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=0}^{i-1} \mu_{i,j} \mathbf{b}_j^*$, $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \|\mathbf{b}_j^*\|^2$. We denote the orthogonal projection to the span of $(\mathbf{b}_0, \cdots, \mathbf{b}_{i-1})$ by π_i, for $i \in \{0, \cdots, d-1\}$, i.e. $\forall \mathbf{v}$, $\pi_i(\mathbf{v}) = \mathbf{v} - \sum_{j=0}^{i-1} \omega_j \mathbf{b}_j^*$, in which $\omega_j = \langle \mathbf{v}, \mathbf{b}_j^* \rangle / \|\mathbf{b}_j^*\|^2$. For $i, j \in \mathbb{Z}_d$ and $0 \leq i < j \leq d-1$, given an arbitrary d-dimensional vector $\mathbf{v} = (v_0, \cdots, v_{d-1})$, define $\mathbf{v}_{[i:j]}$ as $(v_i, \cdots, v_{j-1})$ with size $j-i$. For a lattice basis $\mathbf{B}$, let $\mathbf{B}_{[i:j]} \leftarrow (\mathbf{b}_i, \cdots, \mathbf{b}_{j-1})$. Moreover, we denote $\mathbf{B}_{\pi[i:j]}$ as the local projected block $(\pi_i(\mathbf{b}_i), \cdots, \pi_i(\mathbf{b}_{j-1}))$, and call $\mathcal{L}_{\pi[i:j]}$ the lattice generated by $\mathbf{B}_{\pi[i:j]}$. We use $\mathbf{B}_{\pi[i]}$ and $\mathcal{L}_{\pi[i]}$ as shorthands for $\mathbf{B}_{\pi[i:d]}$ and $\mathcal{L}_{\pi[i:d]}$. An important invariant value of the lattice $\mathcal{L}(\mathbf{B})$ is its volume $\mathrm{Vol}(\mathcal{L}(\mathbf{B})) = \prod_{i=0}^{d-1} \|\mathbf{b}_i^*\|$. The length of the shortest non-zero vector of a lattice $\mathcal{L}(\mathbf{B})$ can be denoted by $\lambda_1(\mathcal{L}(\mathbf{B}))$. We use the abbreviations $\mathrm{Vol}(\mathbf{B}) = \mathrm{Vol}(\mathcal{L}(\mathbf{B}))$ and $\lambda_1(\mathbf{B}) = \lambda_1(\mathcal{L}(\mathbf{B}))$.

[3] Batch “refined-lwe-estimator” in https://github.com/Summwer/lwe-estimator-with-pnjbkz.git

Notations for algorithms description. Let BKZ-β/PnjBKZ-(β, J) be an abbreviation of a one-tour BKZ/PnjBKZ with blocksize β and jump value J, and J is omitted when $J = 1$. Assume $\mathbf{B} = (\mathbf{b}_0, \cdots, \mathbf{b}_{d-1})$, its Gram-Schmidt basis is $\mathbf{B}^* = (\mathbf{b}_0^*, \cdots, \mathbf{b}_{d-1}^*)$. Let $\mathsf{rr}(\mathbf{B}) = (\|\mathbf{b}_0^*\|, \cdots, \|\mathbf{b}_{d-1}^*\|)$, abbreviate to rr. $\mathsf{rr}_{[i:j]} = (\|\mathbf{b}_{i-1}^*\|, \cdots, \|\mathbf{b}_{j-1}^*\|)$. Let $\mathsf{rr}[i]$ be the $(i+1)$-th element of rr.

Denote BKZSim as the BKZ simulator proposed in [31]. The simulation for PnjBKZ is denoted as PnjBKZSim$(\mathsf{rr}(\mathbf{B}), \beta, J, t)$ which simulates a PnjBKZ-(β, J) with t tours on lattice $\mathcal{L}(\mathbf{B})$ and return the new lengths, where the PnjBKZ simulator was proposed in [29]. Moreover, if we have a blocksize and jump strategy S that stores a series of (β_i, J_i), then PnjBKZSim$(\mathsf{rr}, \mathsf{S})$ means iteratively calling a tour of PnjBKZ-(β_i, J_i) simulator on rr, where $(\beta_i, J_i) \in \mathsf{S}$. Let BKZ-$\beta$ reduced basis be the lattice basis after calling sufficient tours of BKZ-β. For simplification, we use β to imply the quality of a BKZ-β reduced basis. Let $\sharp$tours(BKZ-β)/$\sharp$tours(PnjBKZ-(β, J)) be the minimum tours for BKZ-β/PnjBKZ-(β, J) to reach a BKZ-β/PnjBKZ-(β, J) reduced basis, abbreviated as $\sharp$tours. Denote t as the number of tours for implementing BKZ/PnjBKZ with a fixed blocksize (and jump) $\beta/(\beta, J)$.

Let $T_{\mathrm{BKZ}}(\beta)/T_{\mathrm{pnjBKZ}}(\beta, J)$ be the time cost of one BKZ/PnjBKZ tour with blocksize β and jump value J. For a specific blocksize and jump strategy $\mathsf{S} = [(\beta_0, J_0), \cdots, (\beta_{n-1}, J_{n-1})]$, we let $T_{\mathrm{BKZs}}(\mathsf{S})/T_{\mathrm{pnjBKZs}}(\mathsf{S})$ be total time cost for a series of BKZ/PnjBKZ reduction with strategy S, abbreviate it as $T_{\mathrm{BKZs}}/T_{\mathrm{pnjBKZs}}$.

In the searching step, we will consider a high dimension sieve and we denote $T_{\mathrm{sieve}}(d_{\mathrm{svp}})$ as the time cost of sieve dimension d_{svp}, abbreviate it as T_{sieve}. Let PSC be the expected sieve cost to find the target vector.

Definition 1. *(The Gaussian Distribution [22]) Let $\sigma, u \in \mathbb{R}$ be the standard deviation and the mean value respectively, a continuous Gaussian Distribution denoted as $N(u, \sigma^2)$. Its probabilistic density function $\rho_{N(u,\sigma^2)} = e^{-\frac{(x-u)^2}{2\sigma^2}} / \sigma\sqrt{2\pi}$.*

Definition 2. *(Chi-Squared Distribution [22]) Given n random variables $X_i \sim N(0,1)$, the random variables $X_0^2 + \cdots + X_{n-1}^2$ follows a chi-squared distribution χ_n^2 over $\mathbb{R}^*$ of mean n and variance $2n$ with probabilistic density function $\rho_{\chi_n^2}(x) = x^{\frac{n}{2}-1} e^{-\frac{x}{2}} / 2^{\frac{n}{2}} \Gamma(n/2)$. Given n random variables $Y_i \sim N(0, \sigma^2)$, the random variables $Y_0^2 + \cdots + Y_{n-1}^2$ follows a scaled chi-squared distribution $\sigma^2 \cdot \chi_n^2$ over $\mathbb{R}^*$ of mean $n\sigma^2$ and variance $2n\sigma^2$.*

Heuristic 1 *(Gaussian Heuristic [25]) The expected first minimum of a lattice $\mathcal{L}$ (denoted as $\lambda_1(\mathcal{L}(\mathbf{B}))$) according to the Gaussian Heuristic denoted by $\mathrm{GH}(\mathcal{L})$ is given by $\lambda_1(\mathcal{L}(\mathbf{B})) \approx \mathrm{GH}(\mathcal{L}) = \left(\Gamma(\frac{d}{2}+1) \cdot \mathrm{Vol}(\mathcal{L})\right)^{\frac{1}{d}} / \sqrt{\pi} \approx \sqrt{d/(2\pi e)} \cdot \mathrm{Vol}(\mathcal{L})^{\frac{1}{d}}$ We also write $\mathrm{GH}(\mathbf{B}) = \mathrm{GH}(\mathcal{L}(\mathbf{B}))$ and $\mathrm{GH}(\mathsf{rr}_{[i:j]}) = \mathrm{GH}(\mathbf{B}_{\pi[i:j]})$.*

Definition 3. *(HKZ reduction and BKZ reduction [25]) The basis $\mathbf{B}$ of a lattice $\mathcal{L}$ is HKZ reduced if $\mathbf{b}_i^* = \lambda_1(\mathcal{L}(\mathbf{B}_{\pi[i:d]}))$, for all $i < d$. $\mathcal{L}$ is BKZ-β reduced if $\mathbf{b}_i^* = \lambda_1(\mathcal{L}(\mathbf{B}_{\pi[i:\min\{i+\beta,d\}]}))$, for all $i < d$.*

Definition 4. *(Root Hermite Factor [32]) For a basis $\mathbf{B}$ of d-dimensional lattice, the root Hermite factor is defined as $\delta = \left(\|\mathbf{b}_0\| / \mathrm{Vol}(\mathbf{B})^{1/d}\right)^{1/d}$, for estimating* *the equality of the output vector of BKZ. For larger blocksize, it follows the asymptotic formula* $\delta(\beta)^{2(\beta-1)} = \frac{\beta}{2\pi e}(\beta\pi)^{1/\beta}$.

Heuristic 2 *(Geometric Series Assumption (GSA) [27]) Let* $\mathbf{B}$ *be a lattice basis after lattice reduction, then Geometric Series Assumption states that* $\|\mathbf{b}_i^*\| \approx \alpha \cdot \|\mathbf{b}_{i-1}^*\|$, $0 < \alpha < 1$. *Combine the GSA with root-Hermite factor (Definition 4) and* $Vol(\mathcal{L}(\mathbf{B})) = \prod_{i=0}^{d-1}\|\mathbf{b}_i^*\|$, *it infers that* $\alpha = \delta^{-\frac{2d}{d-1}} \approx \delta^{-2}$.

2.2 Lattice Hard Problems

Definition 5. *(unique Shortest Vector Problem(uSVP*$_\gamma$*) [33]) Given an arbitrary basis* $\mathbf{B}$ *on lattice* $\mathcal{L} = \mathcal{L}(\mathbf{B})$, $\mathcal{L}$ *satisfies the condition* $\gamma\lambda_1(\mathbf{B}) < \lambda_2(\mathbf{B})$ *(*$\gamma > 1$, $\lambda_2(\mathbf{B})$ *is norm of the second shortest vector which is linearly independent to the shortest vector), find the shortest non-zero vector* $\mathbf{v}$ *s.t.* $\|\mathbf{v}\| = \lambda_1(\mathbf{B})$.

Definition 6. *(LWE*$_{m,n,q,D_\sigma}$ *Distribution [34–36]) Given some samples* $m \in \mathbb{Z}$, *a secret vector length* $n \in \mathbb{Z}$, *a modulo* $q \in \mathbb{Z}$, *a probability distribution* D_σ. *Uniformly sample a matrix* $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ *and sample a secret vector* $\mathbf{s} \in \mathbb{Z}_q^n$ *from a specific distribution, randomly sample a relatively small noise vector* $\mathbf{e} \in \mathbb{Z}_q^m$ *from Gaussian distribution* D_σ *whose standard deviation is* σ. *The LWE distribution* Ψ *is constructed by the pair* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}) \in (\mathbb{Z}_q^{m\times n}, \mathbb{Z}_q^m)$ *sampled as above.*

Definition 7. *(Search LWE*$_{m,n,q,D_\sigma}$ *problem [34–36]) Given a pair* $(\mathbf{A}, \mathbf{b})$ *sampled from LWE distribution* Ψ *compute the pair* $(\mathbf{s}, \mathbf{e})$.

2.3 Primal Attack

Albrecht *et al* [37] firstly presented the primal attack for the LWE problem, which reduced Standard Form LWE problem to an uSVP$_\gamma$ by Kannan's embedding technique [16]. $(\mathbf{A}, \mathbf{b})$ are LWE instances and the form of the embedding lattice basis is as $\mathbf{B}_{\mathbf{A}',\mathbf{b}} = \begin{pmatrix} \mathbf{A}' & \mathbf{b} \\ \mathbf{0}^T & 1 \end{pmatrix}$, $\mathbf{A}' = \mathbf{P}^{-1}\begin{pmatrix} q\mathbf{I}_{m-n} & \bar{\mathbf{A}} \\ \mathbf{O} & \mathbf{I}_n \end{pmatrix}$, here $\mathbf{P} \in \mathbb{Z}^{m\times m}$ is a permutation matrix such that $\mathbf{P}\cdot\mathbf{A} = \left(\bar{\mathbf{A}}^T, \mathbf{I}_n\right)^T$. Then there is a unusually short lattice vector $\mathbf{v}$=($\mathbf{e}$,1) in this embedding lattice $\mathbf{B}_{\mathbf{A}',\mathbf{b}}$ whose norm $\|\mathbf{v}\| \approx \sigma\sqrt{m}$ is shorter than $\lambda_2(\mathcal{L})$. Thus LWE is reduced to a uSVP on the embedding lattice.

2.4 Core-SVP model [12]

Core-SVP model [12] only considers using the BKZ algorithm with a fixed blocksize β to perform Primal Attack and evaluate the time cost. [12] and [18] give a success condition of such attack: For the minimal blocksize β in the BKZ algorithm (or its variant) to ensure that the following inequality is satisfied $\|\mathbf{v}\|\sqrt{\beta/d} \leq \delta^{2\beta-d}\mathrm{Vol}(\mathcal{L}(\mathbf{B}))^{1/d}$, the unique shortest vector $\mathbf{v}$ will be found by BKZ in time $T(\beta)$ which is an exponential function of β. This success condition based on GSA (Heuristic 2) is a brief justification for the estimation given in [12]. Here δ is the root of the Hermit factor of lattice basis. [38]

gives the following relation between the blocksize and the root Hermite factor $\delta(\beta) \approx \left(((\pi\beta)^{1/\beta}\beta)/(2\pi e)\right)^{1/(2(\beta-1))}$.

Core-SVP model considers neither the number of calls to β-dimension SVP Oracle during one tour of the BKZ algorithm with blocksize β, nor the number of BKZ tours needed to satisfy the success condition. Therefore, the Core-SVP model [12] is considered a conservative LWE security evaluation model. The accurate upper bound number of BKZ tours needed to reach BKZ-β reduced basis is still unknown [39], but [31] suggests that a polynomial number of BKZ-β tours seems sufficient to obtain a lattice basis with Hermite factor near $\delta(\beta)$. When the SVP Oracle used by the BKZ algorithm is BDGL sieving [40], the time cost of solving LWE under Core-SVP model is $T(\beta) \approx O(2^{0.292\beta})$.

2.5 PnjBKZ

PnjBKZ is a BKZ-type reduction algorithm that uses `Pump` as its SVP oracle. Unlike classical BKZ, PnjBKZ performs the SVP oracle with an adjustable jump no less than 1. Specifically, runing a PnjBKZ with blocksize β and jump$=J$, after executing the SVP oracle on a certain block $\mathbf{B}_{[i:i+\beta]}$, the next SVP oracle will be executed on the $\mathbf{B}_{[i+J:i+\beta+J]}$ block with a jump count J rather than $\mathbf{B}_{[i+1:i+\beta+1]}$.

2.6 Dimension for Free (d4f) Technique

D4f technique [25] can bring sub-exponential time speedup and memory decrease for sieve algorithms. In this paper, we consider the theoretical d4f estimation given in [25] as $\mathrm{d4f}(\beta) = \beta \ln(4/3)/\ln(\beta/2\pi e)$, which means that solving β-dimension SVP needs only $\beta - \mathrm{d4f}(\beta)$ dimensional sieving.

2.7 Leaky-LWE-Estimator

The leaky-LWE-Estimator [20] proposed a probabilistic method in LWE estimation as opposed to the GSA-intersect, which relates the solving probability of LWE instance to BKZ blocksizes. The estimator was later applied to the NIST PQC standards such as Kyber and Dilithium along with the estimation in [40], which gives an accurate estimation for LWE rather than a conservative lower bound like Core-SVP model [12]. The leaky-LWE-Estimator [20] computes an expected value $\bar{\beta}$ of the blocksize needed to solve an LWE instance by simulating how the quality of the lattice basis changes during lattice reduction, and estimating the success probability in finding the target vector at each block of the progressive BKZ. Then it substitutes $\bar{\beta}$ into the gate count and memory cost by the list decoding estimation in [24] and obtains a cost estimation for LWE with specific input parameters. Besides, to simplify the calculation process in Leaky-LWE-Estimator, [22] presented a simpler version that has the same estimation results as [20].

One main difference between leaky-LWE-Estimator and Core-SVP model is that leaky-LWE-Estimator uses the BKZ 2.0 simulator [31] denoted as BKZSim to simulate how the lattice basis changed during the reduction of progressive BKZ, which can be used to estimate the number of calls to SVP Oracle with different dimensions and the quality of the lattice basis reduced by a series of BKZ. Another difference is that the leaky-LWE-Estimator considers the length of the target vector as a random variable that follows the chi-square distribution rather than some fixed value. In addition, the leaky-LWE-Estimator uses the gate count method proposed in [24] instead of computational complexity to estimate the hardness of LWE. The detail of the leaky-LWE-Estimator is given in Alg. 1. Here χ^2_β in Alg. 1 is the chi-squared distribution with degree β of freedom.

```
input: $d$, $\mathbf{t}$;
output: $\bar{\beta}$;
Function LeakyLWEEstimator($d, \mathbf{t}$):
    $p_{\text{tot}} \leftarrow 0,\ \bar{\beta} \leftarrow 0$
    rr $\leftarrow$ GSA profile of an LLL reduced, rank $d$, LWE instance
    for $\beta \leftarrow 3$ to $d$ do
        rr $\leftarrow$ BKZSim(rr, $\beta$);
        $p_{\text{lift}} \leftarrow$ Pr[$\mathbf{t}$ recovered in $\lfloor d/\beta \rfloor$ rounds | $\pi_{d-\beta+1}(\mathbf{t})$ recovered this
          round]
        $p_{\text{rec}} \leftarrow \Pr[x \leftarrow \chi^2_\beta : x \leq (\mathsf{rr}[d-\beta])^2]$
        $p_{\text{new}} \leftarrow (1 - p_{\text{tot}}) \cdot p_{\text{rec}} \cdot p_{\text{lift}}$
        $\bar{\beta} \leftarrow \bar{\beta} + p_{\text{new}} \cdot \beta$
        if $p \geq 0.999$ then
            break
    return $\bar{\beta}$
```

Algorithm 1: leaky LWE Estimator proposed in [20]. [22] shows that p_{lift} is always close to 1 and could be deleted in the computation.

After calling Alg. 1 to obtain the expected value of BKZ blocksize $\bar{\beta}$ for solving LWE, leaky-LWE-Estimator will call the Gate-cout algorithm in [24] to calculate the number of gates (time cost): $\texttt{ppgate}(\bar{\beta}) = C^2 \cdot \texttt{agps20gates}(\bar{\beta} - \texttt{d4f}(\bar{\beta}))$ and memory cost: $\texttt{bit}(\bar{\beta}) = 8(\bar{\beta} - \texttt{d4f}(\bar{\beta})) \cdot \texttt{agps20vectors}(\bar{\beta} - \texttt{d4f}(\bar{\beta}))$ for solving the LWE respectively. Here the Gate-count algorithm [24] can analyze the cost of sieving with a classical and quantum circuit and $C = \frac{1}{1-2^{-0.292}}$ is a constant used to simulate the time cost of progressive sieving when BDGL16 sieving [40] is used and progressive BKZ blocksize. More detail about functions $\texttt{agps20gates}(\cdot)$ can be seen in [41].

2.8 PnjBKZ Simulator

The first step in the two-step solving mode is using a series of well-chosen BKZ tours to reduce the lattice basis. Compared with classical BKZ algorithm, the PnjBKZ algorithm (see Sec. 2.5 and Alg. 3 for more detail) is a more efficient lattice reduction algorithm which allows more flexible choice of blocks to be processed in BKZ which uses a sieving algorithm Pump as its SVP oracle.

The PnjBKZ simulator is a polynomial time algorithm to simulate how the quality of the lattice basis changes during the reduction by using the optimized reduction strategy of PnjBKZ-(β, J) with $J > 1$ in [29] without actually running the time-consuming (exponential time cost according to blocksize) PnjBKZ algorithm. The PnjBKZ simulator uses the Gaussian Heuristic and the property of HKZ reduction to estimate how the logarithms of the Gram-Schmidt norms of lattice basis changed after one tour of PnjBKZ-(β, J). For convenience, we declare the notation of PnjBKZ simulation in Sec. 2.

3 Efficiency of Two-step solving mode

In this section, we will show that the Two-step solving mode is more efficient in solving uSVP$_\gamma$ compared with the BKZ-only mode. We use Theorem 1 to illustrate this claim and give the corresponding proof under GSA.

Theorem 1. *Assume Gaussian Heuristic (Heuristic 1), GSA(Heuristic 2) and Heuristic 4 in [29] hold. Let d be the dimension of lattice, $d \geq 100$, we assume that the uSVP$_\gamma$ instance can be solved by BKZ-only mode through a BKZ-β reduced basis with $\frac{d+16}{9} \leq \beta \leq \frac{d}{2}$, and let the time cost for sieving on d-dimensional lattice be $2^{c \cdot d + c_0}$ where $c \leq 0.35$. Then there exists a parameter choice for the Two-step mode which solves the uSVP$_\gamma$ instance in less time than the BKZ-only mode.*

Proof. Let $\mathcal{L}$ be the lattice, $\mathbf{B}$ be its basis and d be the dimension of $\mathcal{L}$, suppose the unique shortest vector is $\mathbf{v}$. Without loss of generality, we set $\mathrm{Vol}(\mathcal{L}) = 1$, let $M = \|\mathbf{v}\|$ be the length of its unique shortest vector. Assume all the orthogonal projections of $\mathbf{v}$ onto the k-dimensional projection sub-lattice $\mathcal{L}_{\pi[d-k]}$ have expected norm $\sqrt{\frac{k}{d}} \cdot M$. Let $\delta(\beta)$ be the root Hermite factor of a BKZ-β reduced basis. Assuming GSA holds, the length of the basis can be estimated by $(\delta(\beta)^d, \delta(\beta)^{d \cdot \frac{d-3}{d-1}}, ..., \delta(\beta)^{-d})$.

Since the projection $\pi_{d-\beta}(\mathbf{v})$ is expected to be the shortest non-zero vector of $\mathcal{L}_{\pi[d-\beta]}$, i.e. $\|\pi_{d-\beta}(\mathbf{v})\| \leq \lambda_1(\mathcal{L}_{\pi[d-\beta]}) \approx \mathrm{GH}(\mathcal{L}_{\pi[d-\beta]})$, then

$$\sqrt{\frac{\beta}{2\pi e}} \cdot \delta(\beta)^{-\frac{d(d-\beta)}{d-1}} \geq \sqrt{\frac{\beta}{d}} \cdot M$$

Next, suppose that the same instance is solved by a Two-step mode on a BKZ-β' reduced basis with a d_{svp} dimensional sieving, where $\beta' < \beta < d_{\text{svp}}$. Then:

$$\sqrt{\frac{d_{\text{svp}}}{2\pi e}} \cdot \delta(\beta')^{-\frac{d(d-d_{\text{svp}})}{d-1}} \geq \sqrt{\frac{d_{\text{svp}}}{2\pi e}} \cdot \delta(\beta)^{-\frac{d(d-\beta)}{d-1}} \geq \sqrt{\frac{d_{\text{svp}}}{d}} \cdot M$$

We find a condition such that the inequality above holds. Since $d_{\text{svp}} > \beta$, we only need to ensure that $\delta(\beta')^{-\frac{d(d-d_{\text{svp}})}{d-1}} \geq \delta(\beta)^{-\frac{d(d-\beta)}{d-1}}$. Take logarithm on both sides, and consider that $\delta(\beta) = (\frac{\beta}{2\pi e} \cdot (\beta\pi)^{\frac{1}{\beta}})^{\frac{1}{2(\beta-1)}}$, we need to ensure that:

$$\frac{d-\beta}{d-d_{\text{svp}}} \geq \frac{\frac{1}{2(\beta'-1)} \cdot (\log \frac{\beta'}{2\pi e} + \frac{1}{\beta'}\log(\beta'\pi))}{\frac{1}{2(\beta-1)} \cdot (\log \frac{\beta}{2\pi e} + \frac{1}{\beta}\log(\beta\pi))}$$

Since $0 < \beta' < \beta$, it infers that $\beta' \log \frac{\beta'}{2\pi e} + \log(\beta'\pi) < \beta \log \frac{\beta}{2\pi e} + \log(\beta\pi)$ always holds. We only need to ensure that:

$$\frac{d-\beta}{d-d_{\text{svp}}} \geq \frac{\beta(\beta-1)}{\beta'(\beta'-1)}$$

Here we consider a special case where $\beta' = \beta - 1$, since if the condition is satisfied under this case, then it is surely satisfied for the optimal choice of β', d_{svp}. We write $d_{\text{gap}} = d_{\text{svp}} - \beta$. We choose d_{gap} to satisfy the condition above, which means that:

$$1 + \frac{d_{\text{gap}}}{d-\beta-d_{\text{gap}}} \geq 1 + \frac{2}{\beta-2}$$

Let T be the time to generate a BKZ-$(\beta-1)$ reduced basis, under the heuristic assumption that generating a BKZ-β reduced basis requires at least one BKZ-β tour, the time of BKZ-only mode $T_{\text{BKZ}-\text{only}} \geq T + T_{\text{BKZ}}(\beta)$, and the time of Two-step mode $T_{\text{Two}-\text{step}} = T + T_{\text{sieve}}(\beta + d_{\text{gap}})$, so we only need to show that for a choice of d_{gap} satisfies the condition above, $T_{\text{sieve}}(\beta + d_{\text{gap}}) \leq (d - \beta + 1)T_{\text{sieve}}(\beta) \leq T_{\text{BKZ}}(\beta)$. Let $T_{\text{sieve}}(d) = 2^{c \cdot d + c_0}$ be the sieve cost model, then we only need to show that $2^{c \cdot d_{\text{gap}}} \leq d - \beta + 1$.

Now we choose $d_{\text{gap}} = \frac{2(d-\beta+1)}{\beta-2}$ which satisfies the condition. By our assumption, $d \leq 9\beta - 16$, so we have that $d_{\text{gap}} \leq 16$. For $c \leq 0.35$, $2^{c \cdot d_{\text{gap}}} \leq 49$, since we assume that $d \geq 100$ and $d \geq 2\beta$, $d - \beta \geq 50$, thus the condition is satisfied. □

All current LWE estimators only consider the security strength of LWE under the BKZ-only solving mode. However, according to Theorem 1, we know that a Two-step mode is more efficient in solving uSVP$_\gamma$ and we should analyze the impact of Two-step mode on the hardness of LWE which can be reduced to uSVP$_\gamma$ under primal attack. So in the following section, we propose a refined Two-step LWE Estimator to evaluate the concrete hardness of LWE by considering Two-step solving mode.

4 A refined Two-step security estimator for solving LWE

In this section, we give the details of our refined Two-step security estimator for solving LWE. The detail of our Two-step LWE Estimator is shown in Section 4.1. Then the verification experiments of our Two-step LWE Estimator are shown in Section 5. In addition, we re-estimate the hardness of LWE instances in NIST

PQC schemes by our Two-step LWE Estimator under a trivial reduction strategy and an optimized reduction strategy [29] respectively in Section 7.1.

In this section, we build our estimator mainly based on leaky-LWE estimator [20]. In fact, constructing our Two-step LWE Estimator based on other security evaluators (such as the LWE-estimator by Albrecht et al. [17]) can also obtain similar conclusions that the Two-step mode of solving LWE will result in a decrease of the estimated security bit. More analysis and estimation results can be seen in Appendix A.

4.1 Two-step LWE Estimator with Trivial Strategy

In this part, we give the detail about our Two-step LWE Estimator(Alg. 2)[3].

```
input  : $n, m, q, \chi$, $\mathsf{S}$;
output: $\mathsf{GB}_{\rm min}$;
Function TwoStepLWEEsimator($n, m, q, \chi$, $\mathsf{S}$):
    $\mathsf{GB}_{\rm min} \leftarrow (+\infty, +\infty)$; $\mathsf{GB} \leftarrow (0,0)$; $\mathsf{GB}_{\rm pre} \leftarrow (0,0)$; $p_{\rm tot} \leftarrow 0$;
    $\mathsf{rr} \leftarrow$ expected length of GS-basis of an LLL reduced $\mathrm{LWE}_{n,m,q,\chi}$ instance;
    for $\beta \in \mathsf{S}$ or $(\beta, J) \in \mathsf{S}$ do
        $\mathsf{rr} \leftarrow \mathsf{BKZSim}(\mathsf{rr}, \beta)$; // $\mathsf{PnjBKZSim}(\mathsf{rr}, \beta, J)$ if $J > 1$;
        $P(\beta) \leftarrow \Pr\left[x \leftarrow \chi^2_\beta \middle| x \leq (\mathsf{rr}[d-\beta])^2\right]$;
        $\mathsf{GB}_{\rm cum} \leftarrow (\sum_{b=\beta_0}^{\beta} \texttt{pbgate}(b - \texttt{d4f}(b)), \texttt{bit}(\beta - \texttt{d4f}(\beta)))$;
        $\mathsf{GB}_{\rm pre} \leftarrow \mathsf{GB}_{\rm pre} + \mathsf{GB}_{\rm cum} \cdot (1 - p_{\rm tot}) \cdot P(\beta)$;
        $p_{\rm tot} \leftarrow p_{\rm tot} + (1 - p_{\rm tot}) \cdot P(\beta)$; $\mathsf{GB}_{\rm csieve} \leftarrow (0,0)$; $P(d_{\rm start} - 1) \leftarrow 0$;
        for $d_{\rm svp} \leftarrow d_{\rm start}$ to $d$ do
            $P(d_{\rm svp}) \leftarrow \Pr\left[x \leftarrow \chi^2_{d_{\rm svp}} \middle| x \leq (\mathrm{GH}(\mathsf{rr}_{[d-d_{\rm svp}:d]}))^2\right]$;
            $\mathsf{GB}_{\rm cum}[0] \leftarrow \mathsf{GB}_{\rm cum}[0] + \texttt{pgate}(d_{\rm svp} - \texttt{d4f}(d_{\rm svp}))$;
            $\mathsf{GB}_{\rm cum}[1] \leftarrow \max\{\mathsf{GB}_{\rm cum}[1], \texttt{bit}(d_{\rm svp} - \texttt{d4f}(d_{\rm svp}))\}$;
            $\mathsf{GB}_{\rm csieve} \leftarrow \mathsf{GB}_{\rm csieve} + \mathsf{GB}_{\rm cum} \cdot (1 - p_{\rm tot}) \cdot (P(d_{\rm svp}) - P(d_{\rm svp} - 1))$;
            if $p_{\rm tot} + (1 - p_{\rm tot}) \cdot P(d_{\rm svp}) \geq 0.999$ then
                break;
        $\mathsf{GB} \leftarrow \mathsf{GB}_{\rm pre} + \mathsf{GB}_{\rm csieve}$;
        if $\mathsf{GB}[0] < \mathsf{GB}_{\rm min}[0]$ then
            $\mathsf{GB}_{\rm min} \leftarrow \mathsf{GB}$;
    return $\mathsf{GB}_{\rm min}$;
```

Algorithm 2: Two-step LWE Estimator

Before we give details of our Two-step LWE Estimator, let us briefly review the leaky-LWE-Estimator which we mainly focus on. Leaky-LWE-Estimator is used by NIST selected PQC schemes [1, 2] to evaluate the security strength of LWE, and is more refined than previous LWE estimators as it considers the randomness of target vector rather than fixed expected value and uses BKZ simulator rather than an estimation from GSA. For BKZ reduction, it used the trivial progressive strategy where the blocksize β is increased by 1 each tour.

We use similar notations in [22]: W be the event of solving LWE during running Progressive PnjBKZ or the final high-dimension Pump of Two-step mode, $\mathrm{W}_{\beta}^{(1)}$ be the event of solving LWE by using BKZ-β, $\mathrm{F}_{\beta}^{(1)} = \neg \mathrm{W}_{\beta}^{(1)}$ and $\mathrm{W}_{(d_{\mathrm{svp}})}^{(2)}$ as the event that a d_{svp}-dimension Pump solved LWE. Here $\Pr[\mathrm{W}_{\beta}^{(1)}] = \Pr\left[x \leftarrow \chi_{\beta}^{2} \middle| x \leq (\mathsf{rr}[d-\beta])^{2}\right]$, and $\mathsf{rr}[d-\beta]$ is the length of the first Gram-Schmidt vector of projective sub-lattice $\mathcal{L}_{\pi[d-\beta:d]}$ of current lattice basis which has been reduced by Progressive BKZ with reduction strategy $\mathsf{S} = \{\beta_i = i+2 \mid i = 1, ..., \mathrm{end}\}$. In Two-step mode we partition W as:

$$\begin{aligned}\Pr[\mathrm{W}] =& \Pr[\mathrm{W}_{\beta_1}^{(1)}] + \Pr[\mathrm{W}_{\beta_2}^{(1)} \wedge \mathrm{F}_{\beta_1}^{(1)}] + \Pr[\mathrm{W}_{\beta_3}^{(1)} \wedge \mathrm{F}_{\beta_2}^{(1)} \wedge \mathrm{F}_{\beta_1}^{(1)}] \\ &+ \cdots + \Pr\left[\mathrm{W}_{\beta_{\mathrm{end}}}^{(1)} \wedge \bigwedge_{j=1}^{\mathrm{end}-1} \mathrm{F}_{\beta_j}^{(1)}\right] + \Pr\left[\mathrm{W}_{d_{\mathrm{svp}}}^{(2)} \wedge \bigwedge_{j=1}^{\mathrm{end}} \mathrm{F}_{\beta_j}^{(1)}\right] \\ =& \sum_{i=1}^{\mathrm{end}} \Pr\left[\mathrm{W}_{\beta_i}^{(1)} \wedge \bigwedge_{i>1,j=1}^{i-1} \mathrm{F}_{\beta_j}^{(1)}\right] + \Pr\left[\mathrm{W}_{d_{\mathrm{svp}}}^{(2)}\right] \cdot \Pr\left[\bigwedge_{j=1}^{\mathrm{end}} \mathrm{F}_{\beta_j}^{(1)}\right]\end{aligned} \tag{1}$$

Here $\mathrm{W}_{d_{\mathrm{svp}}}^{(2)}$ means during the process of the final sieve, d_{svp}-dimension progressive sieving finds the projection vector of the target vector and $\mathrm{F}_{d_{\mathrm{svp}}}^{(2)} = \neg \mathrm{W}_{d_{\mathrm{svp}}}^{(2)}$. Event $\mathrm{W}_{d_{\mathrm{svp}}}^{(2)}$ happened means all BKZ-β in the reduction step fail to find the target vector, other else it will not call the final high-dimension sieve. So event $\mathrm{W}_{d_{\mathrm{svp}}}^{(2)}$ is independent with all events $\mathrm{F}_{\beta_j}^{(1)}$. When evaluating the concrete hardness of a LWE instance, the value of d_{svp} will be set to solve this LWE with a probability above 0.999. Set end as the index of the last block in the BKZ reduced sequence and d_{start} is the dimension of the initial projection sub-lattice in the final sieve.

It is worth noticing that leaky-LWE-Estimator is based on a Heuristic assumption that events $\mathrm{W}_{\beta_i}^{(1)}$ and $\mathrm{F}_{\beta_j}^{(1)}$ for $i \neq j$ are independent. See the discussion in Section 4.1 of [22] or the implementation of leaky-LWE-Estimator: Alg. 1 for more details. The Heuristic assumption that events $\mathrm{W}_{\beta_i}^{(1)}$ and $\mathrm{F}_{\beta_j}^{(1)}$ for $i \neq j$ are independent which leaky-LWE-Estimator based on, is reasonable to some extent if we assume that the lattice basis will be re-randomized each time it is reduced by a stronger BKZ reduction. Below we reformulate this assumption formally:

Heuristic 3 *The lattice basis is randomized each time by a reduction of BKZ-β with larger β. Then events $\mathrm{W}_{\beta_i}^{(1)}$ and $\mathrm{F}_{\beta_j}^{(1)}$ for $i \neq j$ are independent.*

Besides, set event $\mathrm{E}_{\beta_i}^{(1)}$ for $i \in \{1, 2, ...\}$ as the event that solving LWE during the process of running Progressive BKZ: BKZ-$\beta_1, \cdots,$ BKZ-β_i. Based on Heuristic 3, we have $\Pr\left[\mathrm{W}_{\beta_i}^{(1)} \wedge \bigwedge_{i>1,j=1}^{i-1} \mathrm{F}_{\beta_j}^{(1)}\right] = \Pr\left[\mathrm{W}_{\beta_i}^{(1)}\right] \cdot \Pr\left[\bigwedge_{i>1,j=1}^{i-1} \mathrm{F}_{\beta_j}^{(1)}\right]$ and

$$\Pr[\mathrm{E}_{\beta_i}^{(1)}] = \Pr[\mathrm{E}_{\beta_{i-1}}^{(1)}] + \Pr[\mathrm{W}_{\beta_i}^{(1)}] \cdot \left(1 - \Pr[\mathrm{E}_{\beta_{i-1}}^{(1)}]\right). \tag{2}$$

We will use Eq. (2) to calculate the cumulative probability of solving LWE during reduction step, see line 8 of Alg. 2.

However, the same method cannot be used to calculate the probability of solving LWE during the final sieve. Specifically, we use a progressive sieve algorithm as the final sieve, thus we also need to calculate the probability of solving LWE during each step of the progressive sieve. Specifically, we use $\mathrm{W}_i^{(2)}, \mathrm{F}_i^{(2)}$ as the success rate and failing rate that LWE can be solved using a i-dimensional progressive sieve (which performs sieving on projected sub-lattices with dimensions from 2 to i). Unlike in a progressive BKZ, the lattice basis will not change during sieving. Therefore, the similar Heuristic assumption that events $\mathrm{W}_i^{(2)}$ and $\mathrm{F}_j^{(2)}$ for $i \neq j$ are independent cannot be established.

On the contrary, instead of considering that events $\mathrm{W}_{(i)}^{(2)}$ and $\mathrm{F}_{(j)}^{(2)}$ for $i \neq j$ are independent, we consider that there is an inclusive relationship between $\mathrm{W}_{(i)}^{(2)}$ and $\mathrm{W}_{(j)}^{(2)}$ for $j \leq i$, i.e $\mathrm{W}_{(i)}^{(2)} \supseteq \mathrm{W}_{(j)}^{(2)}$. Since the lattice basis will not change during the progressive sieving of Pump and running an i-dimension progressive sieving, it will run a j-dimension progressive sieving at first, for $j \leq i$.

Setting event $\mathrm{E}_\beta^{(2)}$ as the progressive sieving finds the projection of the target vector exactly after a β-dimensional sieve. More specifically, during one progressive sieving, all the sieving dimensions smaller than β failed to find the target vector but succeeded when the sieving dimension equals β. We give the following Heuristic assumption.

Heuristic 4 *For $i \in \{2, ..., d_{\mathrm{svp}}\}$, $\mathrm{W}_i^{(2)} \supseteq \mathrm{W}_{i-1}^{(2)} \supseteq \mathrm{W}_{i-2}^{(2)} \cdots \supseteq \mathrm{W}_2^{(2)}$. Then $\mathrm{E}_i^{(2)} = \mathrm{W}_i^{(2)} - \mathrm{W}_{i-1}^{(2)}$.*

Set $\Pr\left[\mathrm{W}_{d_{\mathrm{start}}-1}^{(2)}\right]$=0, based on Heuristic 4 we calculate $\Pr\left[\mathrm{E}_{d_{\mathrm{svp}}}^{(2)}\right]$ by

$$\Pr\left[\mathrm{E}_{d_{\mathrm{svp}}}^{(2)}\right] = \Pr\left[\mathrm{W}_{d_{\mathrm{svp}}}^{(2)}\right] - \Pr\left[\mathrm{W}_{d_{\mathrm{svp}}-1}^{(2)}\right], \tag{3}$$

which is the key equality to calculate the number of gate in searching step. Then, the cumulative probability of solving LWE in Two-step LWE estimator can be expressed by

$$\begin{aligned}\Pr[\mathrm{W}] &= \Pr[\mathrm{E}_{\beta_{\mathrm{end}}}^{(1)}] + \left(1 - \Pr[\mathrm{E}_{\beta_{\mathrm{end}}}^{(1)}]\right) \sum_{i=d_{\mathrm{start}}}^{d_{\mathrm{svp}}} \Pr\left[\mathrm{E}_i^{(2)}\right] \\ &= \Pr[\mathrm{E}_{\beta_{\mathrm{end}}}^{(1)}] + \left(1 - \Pr[\mathrm{E}_{\beta_{\mathrm{end}}}^{(1)}]\right) \Pr\left[\mathrm{W}_{d_{\mathrm{svp}}}^{(2)}\right],\end{aligned} \tag{4}$$

see the line 15 of of Alg. 2 for more details.

Gates count of reduction step In this part, we introduce how to count the number of Gates when we solved LWE in the reduction step. After we calculate each $\Pr[\mathrm{E}_{\beta_i}^{(1)}]$ value for $i \in \{1, 2, ...\}$ by using Eq. (2) in the reduction step, we can calculate the expected value of gate count G_1 of reduction step. We evaluate the expected value of gates counts G_1 of reduction step by Eq. (5), see line 7 of Alg. 2 for more details. Let $\texttt{gate}(\beta)$ be the gate count of a sieve algorithm with dimension β, $\texttt{pgate}(\beta) = C \cdot \texttt{gate}(\beta)$ be the gate count of a progressive sieve algorithm with dimension β and let $\texttt{pbgate}(\beta) = \texttt{pgate}(\beta) \cdot (d - \beta + 1)$be the gate count of a BKZ-β, then G_1 can be expressed as

$$G_1 = \sum_{i=1}^{\text{end}} \Pr[\mathrm{W}_{\beta_i}^{(1)}] \cdot \left(1 - \Pr[\mathrm{E}_{\beta_{i-1}}^{(1)}]\right) \cdot \left[\sum_{l=0}^{i} \texttt{pbgate}(\beta_l - \texttt{d4f}(\beta_l))\right]. \tag{5}$$

Gates count of searching step In this part, we introduce how to calculate the numbers of Gates when we solved LWE in the searching step. When we solved uSVP in the searching step, it meant that all the BKZ tours in the reduction step failed to find the target vector. Thus, based on Eq. (3) to calculate $\Pr\left[\mathrm{E}_i^{(2)}\right]$, $i \in \{d_{\text{start}}, \ldots, d_{\text{svp}}\}$, we use Eq. (6) to calculate the expected value of gates of the searching step, see line 14 of Alg. 2 for more details.

$$\begin{aligned} G_2 = \sum_{i=d_{\text{start}}}^{d_{\text{svp}}} & \Pr\left[\mathrm{E}_i^{(2)}\right] \cdot \left(1 - \Pr[\mathrm{E}_{\beta_{\text{end}}}^{(1)}]\right) \cdot \\ & \left[\left(\sum_{l=0}^{\text{end}} \texttt{pbgate}\left(\beta_l - \texttt{d4f}(\beta_l)\right)\right) + \texttt{pgate}\left(i - \ \texttt{d4f}(i)\right)\right] \end{aligned} \tag{6}$$

When considering the cost of solving LWE during the searching step, it means that all BKZ tours in the reduction step failed to find the target vector. We calculate the $\Pr\left[\bigwedge_{j=1}^{\text{end}} \mathrm{F}_{\beta_j}^{(1)}\right]$ in Eq. (6) to represent the probability of all BKZ tours in the reduction step failed to find the target vector. Besides, before starting the large dimensional sieve in the searching step, the total time cost of solving the uSVP in the searching step already contains the full-time cost of all BKZ tours in the reduction step. Therefore, the total gate count of the reduction step is $\sum_{l=0}^{\text{end}} \texttt{pbgate}(\beta_l - \texttt{d4f}(\beta_l))$ and when the dimension of SVP Oracle we considered equals to d_{svp}, the gate count of searching step is $\texttt{pgate}(d_{\text{svp}} - \texttt{d4f}(d_{\text{svp}}))$. Here, the $\texttt{d4f}(j)$ is calculated by Sec. 2.6.

Finally, the total gate count for the Two-step mode of solving LWE $G := G_1 + G_2$.

Memory count of Two-step LWE concrete estimator The memory count of the Two-step LWE concrete estimator is similar to gate count, just replace the function $\texttt{gate}(\beta)$ with the memory cost function $\texttt{bit}(\beta)$ which declares the memory cost of one sieve algorithm with dimension β. Since the memory cost of the final sieve and (progressive) BKZ with the same dimension is equal to the memory cost of one sieve algorithm with the same dimension, the memory count of the reduction process in our Two-step LWE Estimator is

$$B_1 = \sum_{i=1}^{\text{end}} \left[\Pr[\mathrm{W}_{\beta_i}^{(1)}] \cdot \left(1 - \Pr[\mathrm{E}_{\beta_{i-1}}^{(1)}]\right)\right] \cdot \texttt{bit}(\beta_l - \texttt{d4f}(\beta_l)), \tag{7}$$

and the memory count of searching step is

$$B_2 = \sum_{i=d_{\text{start}}}^{d_{\text{svp}}} \Pr\left[\mathrm{E}_i^{(2)}\right] \cdot \left(1 - \Pr[\mathrm{E}_{(\beta_{\text{end}})}^{(1)}]\right) \cdot \max\{\texttt{bit}\,(\beta_{\text{end}} - \texttt{d4f}(\beta_{\text{end}}))\,, \texttt{bit}\,(i - \ \texttt{d4f}(i))\}. \tag{8}$$

The total memory count for Two-step mode of solving LWE $B := B_1 + B_2$.

4.2 Two-step LWE Estimator with Refined Strategy

In this section, we adapt the Two-step LWE-estimator to improved progressive PnjBKZ [29], which calls a series of PnjBKZ to reduce the basis first and finds a good timing to use a Pump algorithm to search the unique shortest vector. The concrete process is as Alg. 3.

```
input  : $\mathbf{B}$, $F(\star, \mathcal{D})$;
output: The approximate shortest vector $\mathbf{v}$;
Function ProPnjBKZ($\mathbf{B}$, $F(\star, \mathcal{D})$):
    $\mathbf{B}$ = LLL($\mathbf{B}$);
    Generate Strategy $\mathsf{S}$ using EnumBS or BSSA [29];
    for  $(\beta, J, \sharp\text{tours}) \in \mathsf{S}$ do
        for $t$ from 1 to $\sharp$tours do
            $\mathbf{B} \leftarrow$ PnjBKZ($\mathbf{B}, \beta, J, \sharp\text{tours}$);
    $d_{\text{svp}}$, _ $\leftarrow$ ProSieveDimEst(rr($\mathbf{B}$), $F(\star, \mathcal{D})$); $f \leftarrow$ d4f($d_{\text{svp}}$);
    $\mathbf{B} \leftarrow$ Pump($\mathbf{B}$,$d - d_{\text{svp}}$, $d_{\text{svp}}$, $f$);
    return $\mathbf{v} \leftarrow \mathbf{b}_0$;
```

Algorithm 3: Improved Progressive PnjBKZ

We point out the main differences between the estimator with improved progressive PnjBKZ and the estimator in Section 4.1. First, the use of PnjBKZ allows us to adjust the reduction strategy more freely. Instead of the trivial reduction strategy $\mathsf{S} = \{(\beta_i = i + 2, J_i = 1) \mid i = 1, \cdots\}$ used in leaky-LWE-Estimator and Section 4.1, we can choose a more efficient reduction strategy given by the blocksize and jump strategy enumeration algorithm (EnumBS) in [29].

Secondly, we use the PnjBKZ simulator [29] instead of the original BKZ simulator to simulate how the quality of lattice basis changes during the reduction by a series of PnjBKZ with $J > 1$. The simulator is purely based on the Gaussian Heuristic, which avoids the problem that GSA (Heuristic 2) is not strictly held during the reduction of PnjBKZ. Also, the gate count of a PnjBKZ-(β, J) tour is calculated as $\texttt{pbgate}(\beta, J) = \texttt{pgate}(\beta) \cdot (d - \beta + 1)/J$.

The gate count and memory count can be calculated similarly, we only need to replace the events $\mathrm{W}_{\beta_i}^{(1)}, \mathrm{F}_{\beta_i}^{(1)}$ by $\mathrm{W}_{(\beta_i, J_i)}^{(1)}, \mathrm{F}_{(\beta_i, J_i)}^{(1)}$ which allow $J > 1$ when calculating the probability. We omit further details here.

5 Experiments on verifying the accuracy of Two-step LWE Estimator

In Section 5.1, we mainly focus on the success probabilities of solving LWE by Two-step mode, especially the success probabilities of the last sieve with different sieving dimensions. We give the detail of our verification experiments to verify Heuristic 4 and the accuracy of Eq. (3) and Eq. (4) which are the key equations to calculate the gate number of the searching step. Then we give an experiment to verify the efficiency of the Two-step mode compared with the BKZ-only mode in Section 5.2. Finally in Section 5.3 we compare the Two-step LWE Estimator using different reduction strategies with the leaky-LWE-Estimator.

5.1 Verification Experiments for Success Probability

In particular, we use different parameters of the LWE instances[4] to test the success probabilities of the final sieve when using different progressive sieving dimensions shown in Fig. 1.[5] We choose four different LWE parameters (n=40, α=0.005, q=1601, m=1600), (n=40,α=0.015, q=1601, m=1600), ($n = 60, \alpha = 0.005$, $q = 3607$), (n=45, α=0.010, q=2027, m=2025) for our experiments. For each LWE parameter, we initialize 100 random LWE instances to construct 100 different lattice bases. Each lattice basis corresponds to an uSVP instance with a different target vector. Then we use BKZ/PnjBKZ to do pre-processing by some trivial reduction strategy S. Using LWE parameter (n=40, α=0.005, q=1601, m=1600) for example, we set $\mathsf{S} = \{\beta_1 = 10, ..., \beta_{\text{end}} = 17\}$. Here, 100 different LWE instances under the same parameter are used to fit the distribution of the error vector.

After pre-processing, we set the key parameter $\kappa \in \{0, ..., d-1\}$ to determine the size of the final sieve in the searching phase. In [12], it is assumed that one can solve an LWE by solving a $d-\kappa$ dimension SVP on $\mathcal{L}_{\pi[\kappa:d]}$ as long as $\sigma\sqrt{d-\kappa} < \text{GH}(\mathcal{L}_{\pi[\kappa,d]})$. Here $\sigma\sqrt{d-\kappa}$ is the expected norm of the projected target vector. However, since we consider the square sum of the length of the projected target vector as a chi-squared distribution with $d-\kappa$ degrees of freedom, we calculate the cumulative probability of solving LWE when using a high-dimension Pump in Section 4 by Eq. (4), and the line 15 of Algorithm 2. To verify the Heuristic 4 and the accuracy of Eq. (4), we test the actual success rate of solving LWE under different lattice sieving with different κ value.

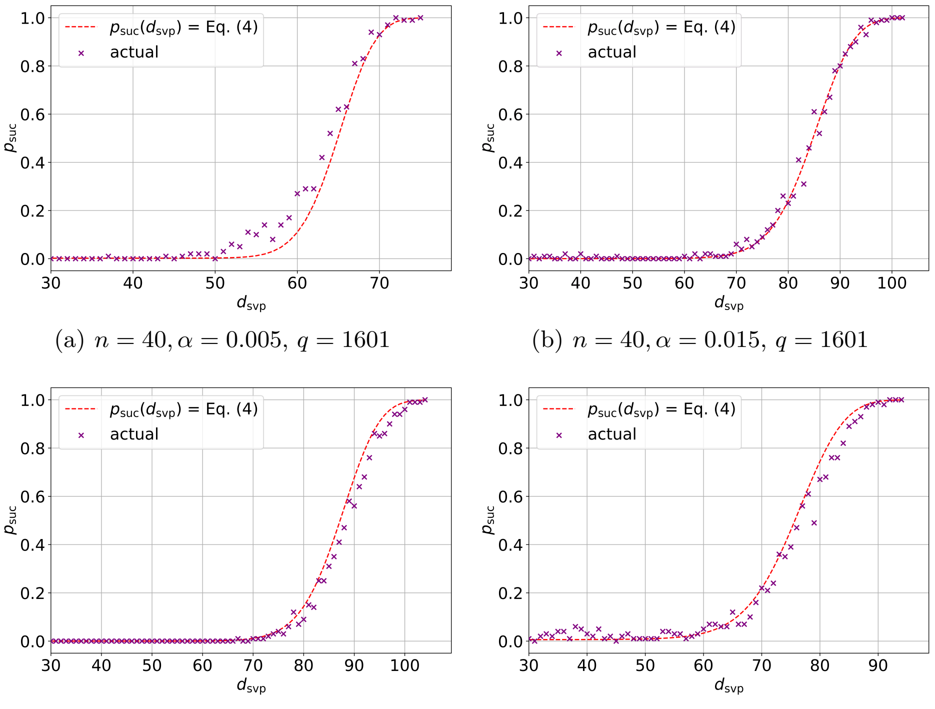
More precisely, we set $d_{\text{svp}} = d-\kappa$ in lattice sieving from 30 to d by adjusting the value of κ and use each sieve with different d_{svp} value to try to find the solution of LWE on 100 different lattice basis after pre-processing. Meantime, we record the actual success rate of each sieve with different d_{svp} values on 100 different lattice bases. Finally, we compare the actual success rate of each sieve

[4] https://www.latticechallenge.org/lwe_challenge/challenge.php

[5] We've uploaded our verification experiment in the file "lwe_prob_test.py " in the website https://github.com/Summwer/test-for-refined-lwe-estimator.

with different d_{svp} with our estimation success rate of solving LWE by the final sieve in Eq. (4), and the line 15 of Algorithm 2. See Fig. 1 for more detail.

From Fig. 1 we can see that the predication of the success rate of solving LWE given by Eq. (4) is consistent with the experimental results, which means our analysis and estimation in Section 4.1 is accurate.



(a) $n = 40, \alpha = 0.005,\ q = 1601$

(b) $n = 40, \alpha = 0.015,\ q = 1601$

(c) $n = 60, \alpha = 0.005,\ q = 3607$

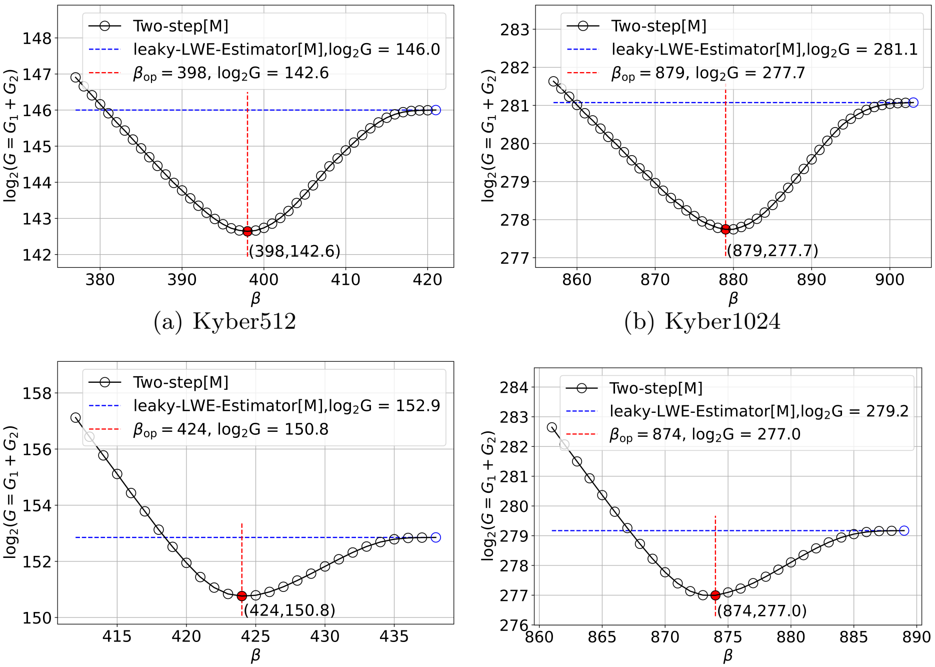
(d) $n = 45, \alpha = 0.010,\ q = 2027$

Fig. 1. Verification experiments of the fitness of the theoretical total success probability $P(d_{\mathrm{svp}})$ = Eq. (4) (the dashed line) to the actual success probability. Test 100 trials and count the success rate for each d_{svp}.

5.2 Verification Experiments for Efficiency of Two-step Mode

In this part, we give an experiment to verify the efficiency of the Two-step mode. In the experiment, we test the public keys of Kyber512, Kyber1024, Dilithium-II, and Dilithium-V as the input LWE instances, then call a Two-step Estimator with $\mathsf{S}[\beta] = \{\beta_i | 3 \le \beta_i \le \beta\}$. The estimator stops at $\beta = \beta_{\mathrm{end}}$ such that the accumulated probability of $\mathsf{S}[\beta_{\mathrm{end}}]$ is no less than 0.999, i.e. $\sum_{i=1}^{\mathrm{end}} \left[\Pr[W_{\beta_i}^{(1)}] \cdot \left(1 - \Pr[E_{i-1}^{(1)}]\right) \right] \ge 0.999$, which is also the condition in leaky-LWE-Estimator.

The Fig. 2 shows the gate count [24] of Two-step mode under different reduction strategy $\mathsf{S}[\beta]$, where $\beta \in \{3, \ldots, \beta_{\text{end}}\}$ and estimated the number of gates given by the leaky-LWE-Estimator. The x-axis of Fig. 2 is the final blocksize β in reduction strategy $\mathsf{S}[\beta]$. Fig. 2 reflects that in solving LWE, the Two-step mode is more efficient than that of using BKZ reduction only and the security estimation given by the leaky-LWE-Estimator [20] is indeed an over-optimistic estimation. Besides, there is optimal timing β_{op} for ending the reduction and entering the searching step as the quality of the lattice basis improved gradually by progressive BKZ.



(a) Kyber512

(b) Kyber1024

(c) Dilithium-II

(d) Dilithium-V

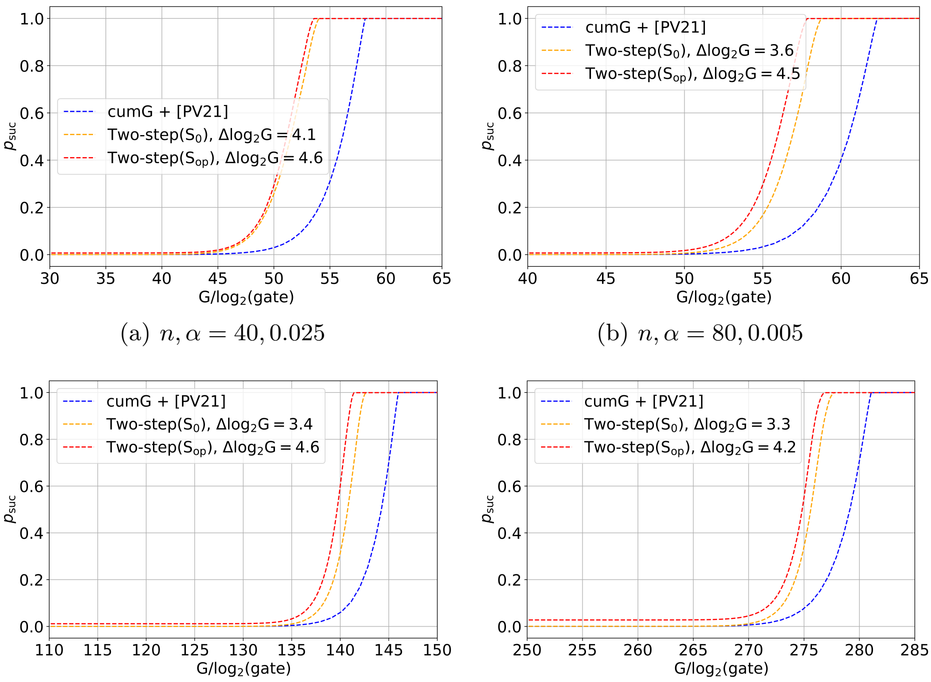
Fig. 2. Two-step efficiency verification Experiment.

5.3 The comparison of different estimation modes

In this part, we compare our Two-step mode estimator using different reduction strategies and different gate count models with the leaky-LWE-Estimator.

We draw Fig. 3 to describe the relationship between the success rate of solving LWE estimated by different estimators and the corresponding number of gates. The blue line in Fig. 3 is the relationship between the expected gates count and the accumulation success probability of solving LWE by pure progressive

BKZ with trivial reduction strategy S_0. These Two-step lines in Fig. 3 are the relationship between the expected gates count and the accumulation success probability of solving LWE by Two-step mode whose reduction step also used a trivial reduction strategy $S_1 : S_1 \subsetneqq S_0$ (In Two-step mode the reduction step will end earlier than in BKZ-only mode). These Enumbs lines in Fig. 3 are also the relationship between the expected gates count and the accumulation success probability of solving LWE by Two-step mode while the reduction strategy is the optimized blocksize and jump selection strategy. [6]



(a) $n, \alpha = 40, 0.025$ (b) $n, \alpha = 80, 0.005$

(c) Kyber512 (d) Kyber1024

Fig. 3. The relation among the growth of cumulated cost and the success probability. Comparison between the output of cumulated Cost Version of [22](Algorithm 1) and Two-step mode(Algorithm 2, this work) for lwe challenge $(n, \alpha) \in \{(40, 0.025), (80, 0.005)\}$ and on Kyber 512 and Kyber 1024 [1]. "Two-step(S_0)" uses a trial progressive BKZ+Pump in Two-step mode to estimate security. "Two-step(S_{op})" uses a progressive BKZ+Pump with the optimized strategy selected by EnumBS [29] in Two-step mode to estimate security. We set $\Delta \log_2 \mathrm{G}$ as the gate count difference between our estimator and the leaky-LWE-estimator both using the same gate count[6].

[6] We use the gate-count model which adopts the improved list-decoding technique proposed in [42]. It fixed the estimate done in [24] of the list-decoding technique proposed in [40].

From Fig. 3 we can see that in both the LWE challenge instances and the LWE instances in NIST standard algorithms, the accumulation success probability of solving LWE by Two-step mode approaches 1 is much faster than that of the leaky-LWE-Estimator. In addition, the expected number of gates in the Two-step solving mode is smaller than that of the leaky-LWE-Estimator when the accumulation success probability of solving LWE approaches 1. Therefore, the evaluation result shows that the leaky-LWE-Estimator gives an optimistic estimation. Besides, both the optimized blocksize and jump selection strategy and the improved list-decoding technique proposed in [42], which fixed the estimate done in [24] of the list decoding technique proposed in [40], can further decrease the estimated security strength by replacing the trivial reduction strategy or gate-count model in Two-step mode. See Fig. 3 for more details about the difference between different estimation models.

6 Improved Conservative Estimation for LWE

Above we consider the LWE estimation by practical solving algorithms. However, since lattice solving algorithms have been developing fast in recent years, such estimation can hardly be considered stable. Many researchers in the field prefer using a theoretical and conservative estimation to estimate the security level of a lattice-based algorithm. In literature, the most used theoretical estimation for LWE-based cryptosystems is the Core-SVP model, first given in NewHope [12]. Many lattice-based algorithms including Kyber and Dilithium use the estimation result of the Core-SVP model to match the security level requirements proposed by NIST.

However, the Core-SVP model can hardly be called accurate. First, the Core-SVP model ignores many coefficients in the estimation, which lowers the estimation result from one aspect. Second, the dimension for free technique has not been taken into account, which causes the estimation result to be higher than expected from another aspect. Despite these two weaknesses of Core-SVP model, there is another main problem in the Core-SVP model, as the underlying solving algorithm in the Core-SVP model is of BKZ-only mode. So for the Core-SVP estimation to hold, it must implicitly assume that a BKZ-only mode lattice solving algorithm is optimal, while such assumption is overthrown by our discussion in Section 3 that a Two-step mode is more efficient than a BKZ-only mode.

In this section, we give a new theoretical lower-bound security estimation for LWE hardness, based on the Two-step solving mode, which relies on weaker assumption than the Core-SVP model. By our estimation result (see Section 7.2), our estimation is higher than the Core-SVP model without considering d4f. While taking d4f into consideration, our estimation turns out to be lower than the Core-SVP estimation, which shows that the original Core-SVP model is in fact not conservative enough without d4f.

6.1 Theoretical lower-bound security estimation of LWE hardness

The idea is simple: we use the time cost of the last lattice sieving in a Two-step mode to estimate the hardness of solving uSVP$_\gamma$ or LWE. Considering that Two-step mode is currently the most efficient way in solving uSVP$_\gamma$ and we omit the time cost of the BKZ reduction step, our estimation is conservative enough.

The main problem in constructing such a lower-bound estimation is to determine the lattice basis quality as the input of lattice sieving step, since we are impossible to give the optimal strategy for BKZ reduction step. So we take an alternative approach: we find the exact basis length rr, such that the best strategy for solving uSVP$_\gamma$ from a basis with length rr is by performing sieving algorithm on a d_{svp} dimensional sublattice rather than performing more BKZ tours before the final lattice sieving.

For simplicity reason, we also assume geometric series assumption (GSA, see Heuristic 2) as in Core-SVP model, so rr can be uniquely determined by the lattice volume V and the root Hermite factor (RHF) δ of the basis. Let $\mathsf{rhf}(\delta, \beta)$ be the new RHF of the basis after current basis with RHF δ reduced by a BKZ-β tour, and $d_{\text{svp}} = \mathsf{md}(\delta, M)$ be the minimum dimension such that a d_{svp} dimension sieving on $\mathcal{L}_{\pi[d-d_{\text{svp}}:d]}$ can recover the unique shortest vector of length M from a lattice basis with RHF δ.

Moreover, we can take dimension for free into account, and let the time cost of sieving on a d_{svp} dimensional lattice be $T_{\text{sieve}}(d_{\text{svp}}) = 2^{c(d_{\text{svp}} - \mathtt{d4f}(d_{\text{svp}}))}$, and $T_{\text{BKZ}}(\beta) = (d-\beta+1) \cdot 2^{c(\beta - \mathtt{d4f}(\beta))}$. Then the condition above can be expressed as the following inequality: $\forall \beta, T_{\text{sieve}}(\mathsf{md}(\delta, M)) \leq T_{\text{BKZ}}(\beta) + T_{\text{sieve}}(\mathsf{md}(\mathsf{rhf}(\delta, \beta), M))$.

It is not hard to show that if δ satisfies this condition, then any $\delta' < \delta$ also satisfies this condition. We only need to find the maximum δ satisfying this condition, and we use $T_{\text{sieve}}(\mathsf{md}(\delta, M))$ for the estimation. Next, we explain how to calculate the value $\mathsf{rhf}(\delta, \beta)$ and $\mathsf{md}(\delta, M)$.

Let $\delta(\beta)$ be the RHF of a BKZ-β reduced basis. Then if $\delta > \delta(\beta)$, using Gaussian Heuristic, the length of $\mathbf{b}_1$ in the lattice basis after a BKZ-β tour can be estimated as: $\text{GH}(\mathsf{rr}_{[0:\beta]} = (\delta^d V^{1/d}, \alpha\delta^d V^{1/d}, ..., \alpha^{\beta-1}\delta^d V^{1/d}))$, where $\alpha = \delta^{-\frac{d-1}{2d}}$ and d is the dimension of lattice basis. Then the RHF of the basis after a BKZ-β tour can be calculated by: $\mathsf{rhf}(\delta, \beta) \approx (\sqrt{\frac{\beta}{2\pi e}} \cdot \delta^{\frac{d \cdot (d-\beta)}{d-1}})^{\frac{1}{d}} = \delta^{\frac{d-\beta}{d-1}} \cdot (\sqrt{\frac{\beta}{2\pi e}})^{\frac{1}{d}}$ and for $\delta \leq \delta(\beta)$, we simply let $\mathsf{rhf}(\delta, \beta) = \delta$.

Next, we estimate the expected dimension of the last lattice sieving. Let M be the expected length of the unique shortest vector, and $M_{d_{\text{svp}}} = M \cdot \sqrt{d_{\text{svp}}/d}$ be the expected length of the projection of M on a d_{svp} dimensional sublattice. We should have that $M_{d_{\text{svp}}} < \text{GH}(\mathsf{rr}_{[d-d_{\text{svp}}:d]} = (\delta^{-d} \cdot V^{1/d} \cdot \alpha^{-d_{\text{svp}}+1}, ..., \delta^{-d} \cdot V^{1/d} \cdot \alpha^{-1}, \delta^{-d} \cdot V^{1/d}))$. Thus we have:

$$M \cdot \sqrt{d_{\text{svp}}/d} < V^{1/d} \cdot \sqrt{\frac{d_{\text{svp}}}{2\pi e}} \cdot \delta^{\frac{d \cdot (d_{\text{svp}} - d)}{d-1}}$$

and the minimum d_{svp} can be recovered by solving the inequation above.

Combining all the things above, we get a lower bound estimation for solving LWE using the Two-step mode. We also explicitly write out the algorithm for lower bound estimation by Alg. 4.

```
input  : $M$, $V \leftarrow \mathrm{Vol}(\mathcal{L})$;
output: $T$;
Function LowerBoundEst($M$, $V \leftarrow \mathrm{Vol}(\mathcal{L})$):
    for $\beta \leftarrow \beta_0$ to $d$ do
        con $\leftarrow$ true;
        $d_{\mathrm{svp}} \leftarrow \mathsf{md}(\delta(\beta), M)$;
        for $\beta' \leftarrow \beta + 1$ to $d$ do
            $\delta' \leftarrow \mathsf{rhf}(\delta(\beta), \beta')$;
            if $T_{\mathrm{sieve}}(d_{\mathrm{svp}}) > T_{\mathrm{BKZ}}(\beta') + T_{\mathrm{sieve}}(\mathsf{md}(\delta', M))$ then
                con $\leftarrow$ false; break;
        if con then
            $\beta_{\mathrm{optimal}} \leftarrow \beta$;
            return $\beta_{\mathrm{optimal}}$, $d_{\mathrm{svp}}$, $T_{\mathrm{sieve}}(d_{\mathrm{svp}})$;
```

Algorithm 4: Lower Bound Estimation

Note that in Alg. 4, we only perform searching on all BKZ-β reduced basis to ensure that the estimation can be done in a reasonable time. This may decrease the estimated time by a small amount, so the estimation only becomes more conservative.

We prove that the new estimation is conservative enough under GSA and two simple heuristic assumptions. We show that our assumptions are strictly weaker than the implicit assumptions in the Core-SVP model, so our estimation is in fact more solid than the Core-SVP estimation.

Heuristic 5 *BKZ is the optimal algorithm for lattice reduction, i.e. generating a lattice basis satisfying GSA.*

Since the Core-SVP model only uses BKZ to estimate the hardness of LWE and also assumes GSA on BKZ-β reduce basis, our assumption is obviously weaker than the implicit assumption in the Core-SVP model.

Heuristic 6 *The best way of solving uSVP$_\gamma$ or LWE is by performing lattice sieving on a projected sublattice on a reduced lattice basis satisfying GSA.*

We note that in the underlying solving algorithm of the Core-SVP model, the unique shortest vector is recovered by sieving on the last β-size block in the lattice, which is only a special case of our assumption. So our assumption is also strictly weaker than the implicit assumption in Core-SVP model.

Theorem 2. *Assume that Gaussian Heuristic (Heuristic 1), GSA(Heuristic 2), Heuristic 5, 6, and Heuristic 4 in [29] hold, then the estimated cost of our* *lower bound estimation is strictly lower than the actual cost for solving uSVP$_\gamma$ in almost all lattices.*

Proof. Let δ, d_{svp} be the intermediate result in our lower bound estimation, i.e. the unique shortest vector is found by performing d_{svp}-dimensional lattice sieving on a lattice basis satisfying GSA which RHF is δ.

Let $\mathcal{A}$ be the optimal algorithm in solving uSVP$_\gamma$. By Heuristic 6, we assume that $\mathcal{A}$ solves uSVP$_\gamma$ by performing d'_{svp}-dimensional lattice sieving on a lattice basis satisfying GSA which RHF is δ'. Furthermore, since in Heuristic 5, we assume that a lattice basis satisfying GSA should be found by BKZ, let β' be the blocksize of the last BKZ tour before the final sieving, and δ'' be the RHF of lattice basis before this BKZ-β' tour. We consider the following cases.

(1) $\delta' \geq \delta$, so $d'_{\text{svp}} \geq d_{\text{svp}}$, thus the running time of $\mathcal{A}$ is larger than $T_{\text{sieve}}(d_{\text{svp}})$.

(2) $\delta'' \leq \delta$, by the definition of δ, we can see that $T_{\text{BKZ}}(\beta') + T_{\text{sieve}}(d'_{\text{svp}}) > T_{\text{sieve}}(\mathsf{md}(\delta'', M))$, so by replacing the final BKZ-β' tour and lattice sieving with a single lattice sieving, the running time of $\mathcal{A}$ decreases, which contradicts with the optimality of $\mathcal{A}$.

(3) $\delta' < \delta < \delta''$. Then $\mathsf{rhf}(\delta, \beta') < \delta'$, so $T_{\text{BKZ}}(\beta') + T_{\text{sieve}}(d'_{\text{svp}}) > T_{\text{BKZ}}(\beta') + T_{\text{sieve}}(\mathsf{md}(\mathsf{rhf}(\delta, \beta'), M)) \geq T_{\text{sieve}}(d_{\text{svp}})$.

Thus we have the result. □

7 Two-step Security Estimation of LWE in NIST Schemes

In this section based on our refined Two-step security estimator, we give a more accurate upper bound estimation of LWE in NIST PQC schemes in Section 7.1. Next, based on our conservative estimation for LWE in Section 6, we give the lower bound estimation of LWE in NIST PQC schemes in Section 7.2.

7.1 Security Upper bound estimation of LWE in NIST PQC schemes

Two-step Security estimation of LWE of NIST PQC schemes. In this part, we will estimate the security strength of LWE instances of NIST PQC schemes by our Two-step LWE hardness estimator in Section 4.1. Besides we use the same blocksize and jump selection strategy: trivial $\mathsf{S}_0 = [\beta_0 = 3, \beta_1 = 4, ..., \beta_{\text{end}}]$ strategy in the reduction step of Two-step mode and the only difference between with leaky-LWE-Estimator is that we consider a Two-step LWE solving mode.

The evaluation results show that even without further optimizing the blocksize and jump selection, the Two-step mode strategy can effectively reduce the estimated security bit of LWE instances in NIST PQC schemes. In particular, under the RAM model, i.e, it assumes that access into even exponentially large memory is free, the estimated security bit of LWE in NIST schemes [23] can be reduced by 2.1∼3.4 bits. See Table 1 for details. Here G and B in Table 1 respectively represent the total log number of logic circuits for event W happened and

Table 1. Security Upper bound Estimation results of different estimators for NIST schemes with different blocksize and jump solving strategies.$^{\natural}$

	$\log_2 \mathrm{G}/\log_2(\text{gates})$			$\log_2 \mathrm{B}/\log_2(\text{bit})$			$\Delta \log_2 \mathrm{G}$	
	Previous	Two-step		Previous	Two-step			
		S_0	S_{op}		S_0	S_{op}	S_0	S_{op}
Kyber512	146	142.6	141.4	93.97	99.1	98.1	3.4	4.6
Kyber768	208.9	205.5	204.4	138.73	144.0	143.2	3.4	4.5
Kyber1024	281.07	277.7	276.9	189.78	195.4	194.6	3.3	4.2
Dilithium-II	152.85	150.8	150.6	97.95	104.3	104.4	2.1	2.3
Dilithium-III	210.23	207.9	207.9	138.8	145.3	145.3	2.3	2.3
Dilithium-V	279.17	277.0	277.0	187.52	194.1	194.1	2.2	2.2

$\natural$ The column "Previous" is the security estimation in the statement of Kyber and Dilithium. Strategy "S_0" uses a trial progressive BKZ+Pump in Two-step mode to estimate security. Strategy "S_{op}" uses a progressive BKZ+Pump with the optimized strategy selected by EnumBS [29] in Two-step mode to estimate security. $\Delta \log_2 \mathrm{G}$ is the difference between "Previous" and "Two-step" under the RAM model in strategy S_0 and S_{op} in the logarithm of gate count with base 2. The gate count of all estimations in this Table uses the same improved list-decoding technique proposed by MATZOV [42]. [6]

the maximum memory needed for event W happened, that both are calculated by Gate-count algorithm [24].

Optimized blocksize and jump selection strategy and Two-step mode. In this part, we quantitatively analyze the impact of the combination of the Two-step LWE solving mode and optimized blocksize and jump selection strategy proposed in [29] on NIST PQC schemes. We change the reduction strategy in the reduction step of Two-step mode from trivial S_0 to the optimized blocksize and jump selection strategy S_{op} proposed in [29]. In other words, we still use Eq. (1), but the reduction strategy used in Eq. (1) is replaced by the optimized blocksize and jump selection strategy proposed in [29].

The evaluation results show that the combination of the optimized blocksize and jump selection and the Two-step mode strategy can indeed effectively reduce the estimated security bit of LWE. Specifically, under the RAM model, the estimated security bit of LWE in NIST schemes [23] can be reduced by 2.2∼4.6 bits. See Table 1 for details. Here G and B in Table 1 respectively represent the total log number of logic circuits for event W happened and the maximum memory needed for event W happened, that both are calculated by Gate-count algorithm [24] under the optimized blocksize and jump selection.

In practice, without considering the RAM model, a large `Pump` dimension in Two-step mode will indeed lead to an extra cost while accessing exponentially large memory, which will somewhat partially offset the above-claimed decrease of security hardness. However, it is unclear what the practical influence of increasing memory cost is on the total time cost. In fact, it is still an open question, see Q5 in Section 5.3 of [1]. Besides, although [43] gave an experimental analysis of an idealized model for the sieve algorithm, its theoretical analysis of hidden probabilistic overhead in near-neighbors search still remains an open problem. So our analysis in this section does not address these two parts.

7.2 Lower bound estimation of LWE in NIST PQC schemes

In this part, we will calculate the lower-bound security estimation of NIST lattice-based standardization. As the dimension of the embedding lattice basis $d = m + n + 1$ can be further optimized by appropriately choosing the number of LWE samples $m \in \{1, ..., m_{\max}\}$. We numerically optimize the number of LWE samples m to minimize the lower-bound security estimation by Alg. 5. See Table 2 for more detail. Table 2 illustrates that by optimizing the number of LWE samples m, compared with the conservative estimation given by the Core-SVP model, the lower-bound security estimation of NIST lattice-based standardization calculated by Alg. 5 increased by $4.17 \sim 8.11$ bits. However, when considering d4f technique, compared with the conservative estimation given by the Core-SVP model, the security bit of NIST lattice-based standardization will decrease by $3.42 \sim 14.76$ bits under our new lower-bound security estimation. It indicates that the Core-SVP model is not conservative enough to offset the influence of the d4f technique.

Furthermore, Table 2 also shows that there indeed exist a β_{optimal} s.t $d_{\text{svp}} = \mathsf{md}(\delta(\beta_{\text{optimal}}), M)$, for any $\beta' \in \{\beta_{\text{optimal}} + 1, ..., d\}, \delta' = \mathsf{rhf}(\delta(\beta), \beta')$ satisfied $T_{\text{sieve}}(d_{\text{svp}}) < T_{\text{BKZ}}(\beta') + T_{\text{sieve}}(\mathsf{md}(\delta', M))$ under the parameter of NIST lattice-based standardization [1,2]. See Table 2 for more detail.

Table 2. The security lower bound estimation of NIST lattice-based standardization[†].

	Kyber512	Kyber768	Kyber1024	DilithiumII	DilithiumIII	DilithiumV
Lattice Dim d	1003	1424	1885	2049	2561	3582
BKZ β	406	625	877	423	624	863
CoreSVP	118	182	256	123	182	252
Lattice Dim d	1025	1477	1954	2039	2672	3461
β_{optimal}	392	608	857	415	614	853
d_{svp}	423	641	891	449	649	889
LBE	123.52	187.17	260.17	131.11	189.51	259.59
LBE (d4f)	112.44	172.32	241.24	119.57	174.52	240.69
ΔHardness	5.52	5.17	4.17	8.11	7.51	7.59
ΔHardness (d4f)	-5.56	-9.68	-14.76	-3.43	-7.48	-11.31

[†] Here the row of "LBE" is the lower bound estimation evaluated by Algorithm 5, the row of "LBE (d4f)" is the lower bound estimation by considering d4f technique, and the row of "CoreSVP" represents the security strength evaluated by the CoreSVP model. "Lattice Dim d" is the dimension for constructing the embedding lattice in primal attack.

8 Conclusion

In this paper, we construct a Two-step LWE hardness estimator which estimates the hardness of LWE under primal attack using a combination of BKZ and sieving. To verify the accuracy of our Two-step LWE hardness estimator, we did extensive experiments, and the experiment results are consistent with our estimation. Besides, we also propose a conservative estimation for LWE considering the attack in Two-step mode. Compared with the most conservative Core-SVP model, our conservative estimation relies on weaker assumptions.

To figure out the influence of the Two-step mode on security estimation of NIST PQC schemes, We re-evaluate the concrete hardness of schemes by our Two-step LWE hardness estimator with a trivial reduction strategy and optimized blocksize and jump selection strategy. Evaluation results show that the the upper bound security estimation given by the leaky-LWE-Estimator [20] is an over-optimistic estimation and the security bit drops by 2.1 ∼ 3.4 bits under trivial reduction strategy and drops by 2.2 ∼ 4.6 bits under optimized blocksize and jump selection strategy. For the lower bound security bit of NIST PQC schemes, our conservative estimation is 4.17 ∼ 8.11 bits higher than the Core-SVP estimation. Therefore, we give more accurate estimations on both the upper bound and lower bound of the hardness of LWE.

```
input  : $m_{\max}$, $n$, $\sigma$, $q$;
output: $\beta_{\text{optimal}}$, $d_{\text{svp}}$, $T_{\text{sieve}}(d_{\text{svp}})$;
Function LowerBoundEstWithOptimalM($m_{\max}$, $n$, $\sigma$, $q$):
    $d^*_{\text{svp}} \leftarrow m_{\max}+n+1$; $m_{\text{optimal}} \leftarrow m_{\max}$; $\beta_{\text{optimal}} \leftarrow m_{\max}+n+1$;
    for $m \leftarrow m_{\max}$ to 1 do
        $d \leftarrow n+m+1$; $M \leftarrow \sigma \cdot \sqrt{d}$; $V \leftarrow q^m$;
        $\beta_{\text{current}}, d_{\text{svp}}, T_{\text{sieve}}(d_{\text{svp}}) \leftarrow$ LowerBoundEst$(M, V)$;
        if $d^*_{\text{svp}} > d_{\text{svp}}$ then
            $d^*_{\text{svp}} \leftarrow d_{\text{svp}}$; $m_{\text{optimal}} \leftarrow m$; $\beta_{\text{optimal}} \leftarrow \beta_{\text{current}}$;
    $d_{\text{optimal}} \leftarrow m_{\text{optimal}}+n+1$;
    return $d_{\text{optimal}}$, $\beta_{\text{optimal}}$, $d^*_{\text{svp}}$, $T_{\text{sieve}}(d^*_{\text{svp}})$;
```

Algorithm 5: Lower Bound Estimation with Optimal m

Acknowledgments This work was supported in part by the National Key R&D Program of China under Grant No. 2023YFB4403500, the National Natural Science Foundation of China (Nos. U2336210, 62272362), and Shanghai Science and Technology Innovation Action Plan (No. 23511100300). L. Wang is funded by the China Scholarship Council (No. 202006960055).

A Appendix. Two-step LWE estimator based on classicial LWE estimator.

In this Appendix A, we will show that based on other lwe estimators for instance the lwe estimator proposed by Albrecht *et al* in [34], to consturct Two-step LWE estimator. The evaluation results show that Two-step mode also is more efficient in solving LWE than using BKZ algorithm only.

In particular, same as the LWE estimation proposed in [34], we also based on GSA to give a time cost model of the Two-step solving mode. Then we show that there is an optimal timing for ending the reduction and entering the searching step as the quality of lattice basis improved gradually by progressive BKZ.

Two-step LWE solving mode The Two-step mode we considered here will call progressive BKZ first for lattice reduction and call a Pump algorithm to find the target vector on the well-reduced lattice basis at the searching step. We set $T_1(\beta)$ as the time cost of obtaining a BKZ-β reduced basis in the reduction step. Then we set $T_2(\beta)$ as the time cost of calling Pump to find the target vector on BKZ-β reduced basis in the searching step. Then the total cost of such a special Two-step LWE solving mode control by reduction parameter β is $T_1(\beta)+T_2(\beta)$.

In the following parts, we will consider that Two-step mode is more efficient than bkz-only mode in the evaluation model of lattice-estimator [17].

Simple time cost model of reduction step Let $\sharp$tours be the minimum number of tours that each BKZ-β during progressive BKZ reduction needed to obtain the BKZ-β reduced basis. First of all, we set $T_1(\beta) = \sharp\text{tours} \cdot \sum_{i=2}^{\beta}(d-i+1)\cdot 2^{c\cdot i} \leq \sharp\text{tours} \cdot C(d-\beta+1)\cdot 2^{c\beta}$ be the time cost of progressive BKZ from BKZ-2 to BKZ-β. Here c is the coefficients related to the sieving algorithm. For example, when using BDGL sieving $c = 0.292$ by using the classical computer and $c = 0.265$ by using the quantum computer. And $C = 1/(1-2^{-c}) \approx$5.46 as the limit of ratio between $\sum_{i\leq\beta} 2^{ci+o(i)}$ and $2^{c\beta+o(\beta)}$ when β grows. When using the heuristic used in lattice-estimator [17] $\sharp$tours=1, which assumed that after reduction of the progressive BKZ-β for each blocksize running once only, one can obtain the BKZ-β reduced basis. Thus, we get the time cost $T_1(\beta)$ of the reduction step which can obtain a BKZ-β reduced basis.

Simple time cost model of search step Secondly, to evaluate the time cost of the search step, we need to describe the relationship between the quality of the lattice basis and the time cost of a final pump in the search step of the Two-step mode. We choose the estimation in [44] which is based on GSA and uses the information of the current lattice basis to give a more refined upper bound of d4f value when solving LWE compared to the asymptotically upper bound of the d4f value proposed in [25]. The asymptotically upper bound of d4f value given in [25] holds if and only if that lattice $\mathcal{L}$ needs to be BKZ-$d/2$ reduced. Here d is the dimension of the lattice basis. Since we use a progressive reduction to gradually improve the quality of the lattice basis, at the beginning of reduction, the quality of the initial lattice basis is far away from BKZ-$d/2$ reduced. Therefore the asymptotically upper bound of d4f value given in [25] is inaccurate until proper reduction is done to achieve that lattice $\mathcal{L}$ is BKZ-$d/2$ reduced.

Here we directly use the result of [44] that for the embedding lattice, $\mathcal{L}$ in prime attack with dimension d and the root hermit factor of this lattice basis is $\delta(\beta)$, to find the target vector $\mathbf{t} = (\mathbf{e}, 1)$ in this lattice $\mathcal{L}$ by final sieve. Here $\mathbf{e}$ is the error vector of the LWE instance. Based on GSA, $\lambda_1\left(\mathcal{L}_{[f:d]}\right) = \sqrt{\frac{d-f}{2\pi e}\frac{|\det(L)|^{\frac{1}{d}}}{\delta(\beta)^f}}$ and Substituting it into the optimistic condition $\mathrm{GH}\left(\mathcal{L}_{[f:d]}\right)\sqrt{4/3} \geq \pi_f(\lambda_1(\mathcal{L})) \approx \sqrt{\frac{d-f}{d}}\sigma\sqrt{d}$ from [25], we get the sieving dimension of the last sieve in searching step $d_{\text{sieving}}(\delta(\beta))$:

$$d_{\text{sieving}}(\delta(\beta)) \geq d - \log_{\delta(\beta)} \frac{|\det(\mathcal{L})|^{\frac{1}{d}}\sqrt{2}}{\sigma\sqrt{3\pi e}} \tag{9}$$

where the optimistic condition $\mathrm{GH}\left(\mathcal{L}_{[f:d]}\right)\cdot\sqrt{4/3} \geq \pi_f(\lambda_1(\mathcal{L}))$ given in [25] by accounting the fact that the $d-f$ dimensional sieving algorithm on $\mathcal{L}_{[f:d]}$ heuristically finds all vectors up to norm $\mathrm{GH}\left(\mathcal{L}_{[f:d]}\right)\cdot\sqrt{4/3}$ and σ is the standard deviation of LWE instances. In standard form LWE the expected length of the target vector in the embedding lattice is $\sigma\sqrt{d}$[6]. Here $\delta(\beta)$ is the root Hermit factor of the current lattice basis which is one of the lattice basis quality measurement values controlled by β.

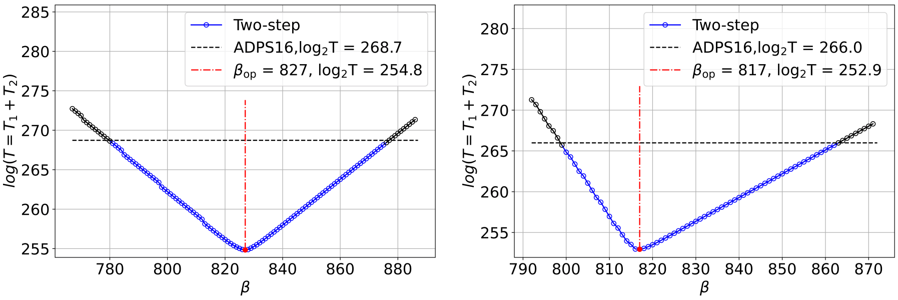
Then we set $T_2(\beta) = 2^{c\cdot d_{\text{sieving}}(\delta(\beta))}$ as the simple time cost evaluation of the searching step which calls a pump on a BKZ-β reduced lattice basis with sieving dimension $d_{\text{sieving}}(\delta(\beta))$. Here minimum $d_{\text{sieving}}(\delta(\beta))$ is calculated by Eq. (9) which describes the relationship between the current quality of lattice basis and the sieving dimension of the last pump in searching step. It shows that a better BKZ-β reduced basis (smaller δ value) can decrease the time cost of $T_2(\beta)$ since a more reduced lattice basis can obtain a bigger dimension for free value when we use the high dimension pump to search the projected of target vector.

Simple time cost model of Two-step mode Finally, we give a simple time cost model $T(\beta)$ of a special Two-step mode that ends BKZ reduction when the lattice basis is BKZ-β reduced and use the Pump to find the target vector by considering the length of the target vector as fixed expected value. We calculate $T(\beta)$ by Eq. (10):

$$T(\beta) = T_1(\beta) + T_2(\beta) = \sharp\text{tours}\cdot C(d-\beta+1)\cdot 2^{c\beta} + 2^{c\cdot d_{\text{sieving}}(\delta(\beta))} \tag{10}$$

[6] Note that we consider the length of the target vector by its expected value the same as what lattice-estimator did.

Comparison between Two-step Mode and BKZ-only Mode We set $T_{\text{bkz}-\text{only}} = \sharp\text{tours} \cdot C(d-\beta_{\text{adps16}}+1) \cdot 2^{c\beta_{\text{adps16}}}$. Here β_{adps16} is the blocksize estimated from [12] which is minimal β_{adps16} s.t $\sigma\sqrt{\beta_{\text{adps16}}} \approx \|\pi_{d-\beta_{\text{adps16}}}(\mathbf{t})\| \leq \text{GH}(\mathbf{B}_{\pi[d-\beta_{\text{adps16}}]})$ holds. This β_{adps16} is minimal blocksize to make BKZ-only mode successful. In the calculation of both $T_1(\beta)$ and $T_{\text{bkz}-\text{only}}$, we use the same heuristic that after one tour reduction of the progressive BKZ-β that blocksize β gradually increased from 2 to β, one can obtain the BKZ-β reduced basis. Finally, we choose the parameters of Kyber-1024 and Dilithum-V as an example, we set $c = 0.292$ and respectively calculate $T(\beta)$ under different lattice reduction qualities and $T_{\text{bkz}-\text{only}}$. See Fig. 4 for more detail.



(a) Kyber-1024

(b) Dilithium-V

Fig. 4. $T(\beta)$ by Eq. (10) with different reduction quality.

From Fig. 4 we can see that as the quality of the lattice basis increased (the x-axis reflects that the current lattice basis is BKZ-β reduced), the total time cost $T(\beta)$ of solving LWE by using Two-step mode under our simple time cost model will decrease first then increase. So there is an optimal timing of entry in the searching step to make the $T(\beta)$ minimum, which we use the red line to indicate this location. Besides the minimum total cost of solving LWE by using Two-step mode $T(\beta)$ is smaller than that of $T_{\text{bkz}-\text{only}}$ which only uses progressive BKZ to solve LWE. In fact all blue points in Fig. 4 have a smaller time cost compared with $T_{\text{bkz}-\text{only}}$. Therefore, we observed that in solving LWE, the Two-step mode is more efficient than that of using BKZ reduction only based on the model of Albrecht et al. to construct the Two-step estimator.

References

1. R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "Kyber(Round 3)," p. 42, 2020.
2. L. Ducas, T. L. Eike Kiltz, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, *Dilithium(Round 3)*. NIST PQC probject, 2020.

3. T. Pornin and T. Prest, “More efficient algorithms for the ntru key generation using the field norm,” in *Public-Key Cryptography – PKC 2019* (D. Lin and K. Sako, eds.), (Cham), pp. 504–533, Springer International Publishing, 2019.
4. R. Steinfeld, S. Contini, K. Matusiewicz, J. Pieprzyk, J. Guo, S. Ling, and H. Wang, “Cryptanalysis of lash,” in *Fast Software Encryption* (K. Nyberg, ed.), (Berlin, Heidelberg), pp. 207–223, Springer Berlin Heidelberg, 2008.
5. L. Ducas, V. Lyubashevsky, and T. Prest, “Efficient identity-based encryption over ntru lattices,” in *Advances in Cryptology – ASIACRYPT 2014* (P. Sarkar and T. Iwata, eds.), (Berlin, Heidelberg), pp. 22–41, Springer Berlin Heidelberg, 2014.
6. X. Boyen, “Attribute-based functional encryption on lattices,” in *Theory of Cryptography* (A. Sahai, ed.), (Berlin, Heidelberg), pp. 122–142, Springer Berlin Heidelberg, 2013.
7. J. M. B. Mera, A. Karmakar, T. Marc, and A. Soleimanian, “Efficient lattice-based inner-product functional encryption,” in *Public-Key Cryptography – PKC 2022* (G. Hanaoka, J. Shikata, and Y. Watanabe, eds.), (Cham), pp. 163–193, Springer International Publishing, 2022.
8. J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic encryption for arithmetic of approximate numbers,” in *Advances in Cryptology – ASIACRYPT 2017* (T. Takagi and T. Peyrin, eds.), (Cham), pp. 409–437, Springer International Publishing, 2017.
9. M. Liu and P. Q. Nguyen, “Solving BDD by Enumeration: An Update,” in *Topics in Cryptology – CT-RSA 2013* (E. Dawson, ed.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 293–309, Springer, 2013.
10. S. Arora and R. Ge, “New algorithms for learning in presence of errors,” in *Automata, Languages and Programming* (L. Aceto, M. Henzinger, and J. Sgall, eds.), (Berlin, Heidelberg), pp. 403–415, Springer Berlin Heidelberg, 2011.
11. P. Kirchner and P.-A. Fouque, “An improved bkw algorithm for lwe with applications to cryptography and lattices,” in *Advances in Cryptology – CRYPTO 2015* (R. Gennaro and M. Robshaw, eds.), (Berlin, Heidelberg), pp. 43–62, Springer Berlin Heidelberg, 2015.
12. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, “Post-quantum Key Exchange—A New Hope,” pp. 327–343, 2016.
13. M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer, “Revisiting the Expected Cost of Solving uSVP and Applications to LWE,” in *Advances in Cryptology – ASIACRYPT 2017* (T. Takagi and T. Peyrin, eds.), (Cham), pp. 297–322, Springer International Publishing, 2017.
14. M. R. Albrecht, “On Dual Lattice Attacks Against Small-Secret LWE and Parameter Choices in HElib and SEAL,” in *Advances in Cryptology – EUROCRYPT 2017* (J.-S. Coron and J. B. Nielsen, eds.), (Cham), pp. 103–129, Springer International Publishing, 2017.
15. T. Espitau, A. Joux, and N. Kharchenko, “On a Dual/Hybrid Approach to Small Secret LWE: A Dual/Enumeration Technique for Learning with Errors and Application to Security Estimates of FHE Schemes,” in *Progress in Cryptology – INDOCRYPT 2020* (K. Bhargavan, E. Oswald, and M. Prabhakaran, eds.), vol. 12578, pp. 440–462, Cham: Springer International Publishing, 2020. Series Title: Lecture Notes in Computer Science.
16. R. Kannan, “Improved algorithms for integer programming and related lattice problems,” in *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, STOC ’83, (New York, NY, USA), pp. 193–206, Association for Computing Machinery, Dec. 1983.

17. M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," *Journal of Mathematical Cryptology*, vol. 9, no. 3, pp. 169–203, 2015.
18. M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer, "Revisiting the Expected Cost of Solving uSVP and Applications to LWE," in *Advances in Cryptology – ASIACRYPT 2017* (T. Takagi and T. Peyrin, eds.), vol. 10624, pp. 297–322, Cham: Springer International Publishing, 2017. Series Title: Lecture Notes in Computer Science.
19. M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, and T. Wunderer, "Estimate All the {LWE, NTRU} Schemes!," in *Security and Cryptography for Networks* (D. Catalano and R. De Prisco, eds.), vol. 11035, pp. 351–367, Cham: Springer International Publishing, 2018. Series Title: Lecture Notes in Computer Science.
20. D. Dachman-Soled, L. Ducas, H. Gong, and M. Rossi, "Lwe with side information: Attacks and concrete security estimation," in *Advances in Cryptology – CRYPTO 2020* (D. Micciancio and T. Ristenpart, eds.), (Cham), pp. 329–358, Springer International Publishing, 2020.
21. S. Bai, S. Miller, and W. Wen, "A refined analysis of the cost for solving lwe via usvp," in *Progress in Cryptology – AFRICACRYPT 2019* (J. Buchmann, A. Nitaj, and T. Rachidi, eds.), (Cham), pp. 181–205, Springer International Publishing, 2019.
22. E. W. Postlethwaite and F. Virdia, "On the Success Probability of Solving Unique SVP via BKZ," in *Public-Key Cryptography – PKC 2021* (J. A. Garay, ed.), vol. 12710, pp. 68–98, Cham: Springer International Publishing, 2021. Series Title: Lecture Notes in Computer Science.
23. I. T. L. C. S. R. CENTER, "Post-quantum cryptography pqc selected algorithms 2022." https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.
24. M. R. Albrecht, V. Gheorghiu, E. W. Postlethwaite, and J. M. Schanck, "Estimating quantum speedups for lattice sieves," in *Advances in Cryptology – ASIACRYPT 2020* (S. Moriai and H. Wang, eds.), (Cham), pp. 583–613, Springer International Publishing, 2020.
25. L. Ducas, "Shortest Vector from Lattice Sieving: A Few Dimensions for Free," in *Advances in Cryptology – EUROCRYPT 2018* (J. B. Nielsen and V. Rijmen, eds.), (Cham), pp. 125–145, Springer International Publishing, 2018.
26. Y. Aono, Y. Wang, T. Hayashi, and T. Takagi, "Improved Progressive BKZ Algorithms and Their Precise Cost Estimation by Sharp Simulator," in *Advances in Cryptology – EUROCRYPT 2016* (M. Fischlin and J.-S. Coron, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 789–819, Springer, 2016.
27. M. R. Albrecht, L. Ducas, G. Herold, E. Kirshanova, E. W. Postlethwaite, and M. Stevens, "The General Sieve Kernel and New Records in Lattice Reduction," in *Advances in Cryptology – EUROCRYPT 2019* (Y. Ishai and V. Rijmen, eds.), (Cham), pp. 717–746, Springer International Publishing, 2019.
28. M. R. Albrecht, C. Yun, and H. Hunt, "lattice-estimator." https://github.com/malb/lattice-estimator.
29. W. Xia, L. Wang, GengWang, D. Gu, and B. Wang, "Improved progressive bkz with lattice sieving." Cryptology ePrint Archive, Paper 2022/1343, 2022. https://eprint.iacr.org/2022/1343.
30. Z. Zhao and J. Ding, "Practical Improvements on BKZ Algorithm," in *Cyber Security, Cryptology, and Machine Learning: 7th International Symposium, CSCML 2023, Be'er Sheva, Israel, June 29–30, 2023, Proceedings*, (Berlin, Heidelberg), pp. 273–284, Springer-Verlag, June 2023.

31. Y. Chen and P. Q. Nguyen, “BKZ 2.0: Better Lattice Security Estimates,” in *Advances in Cryptology – ASIACRYPT 2011* (D. H. Lee and X. Wang, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 1–20, Springer, 2011.
32. P.-Q. Chen, Yuanmi; Nguyen, *Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe*. PhD Thesis, 2013.
33. V. Lyubashevsky and D. Micciancio, “On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem,” in *Advances in Cryptology - CRYPTO 2009* (S. Halevi, ed.), vol. 5677, pp. 577–594, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. Series Title: Lecture Notes in Computer Science.
34. M. R. Albrecht, R. Player, and S. Scott, “On the concrete hardness of Learning with Errors,” *Journal of Mathematical Cryptology*, vol. 9, Jan. 2015.
35. C. Peikert, “A Decade of Lattice Cryptography,” *Found. Trends Theor. Comput. Sci.*, vol. 10, pp. 283–424, Mar. 2016. Place: Hanover, MA, USA Publisher: Now Publishers Inc.
36. K. Xagawa, “Cryptography with Lattices,” p. 244, 2010.
37. M. R. Albrecht, R. Fitzpatrick, and F. Göpfert, “On the Efficacy of Solving LWE by Reduction to Unique-SVP,” in *Information Security and Cryptology – ICISC 2013* (H.-S. Lee and D.-G. Han, eds.), (Cham), pp. 293–310, Springer International Publishing, 2014.
38. Y. Chen, “Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe,” 2013.
39. G. Hanrot, X. Pujol, and D. Stehlé, “Analyzing blockwise lattice algorithms using dynamical systems,” in *Advances in Cryptology – CRYPTO 2011* (P. Rogaway, ed.), (Berlin, Heidelberg), pp. 447–464, Springer Berlin Heidelberg, 2011.
40. A. Becker, L. Ducas, N. Gama, and T. 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*, SODA ’16, (USA), pp. 10–24, Society for Industrial and Applied Mathematics, Jan. 2016.
41. L. Ducas and M. Rossi, “leaky-lwe-estimator.” https://github.com/lducas/leaky-LWE-Estimator/tree/NIST-round3.
42. MATZOV, “Report on the Security of LWE: Improved Dual Lattice Attack,” Apr. 2022.
43. L. Ducas, “Estimating the Hidden Overheads in the BDGL Lattice Sieving Algorithm,” in *Post-Quantum Cryptography* (J. H. Cheon and T. Johansson, eds.), Lecture Notes in Computer Science, (Cham), pp. 480–497, Springer International Publishing, 2022.
44. L. Wang, Y. Wang, and B. Wang, “A trade-off svp-solving strategy based on a sharper pnj-bkz simulator,” in *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, ASIA CCS ’23, (New York, NY, USA), p. 664–677, Association for Computing Machinery, 2023.