Hybrid Dual Attack on LWE with Arbitrary Secrets

Lei Bi[1,2⋆], Xianhui Lu[1,2], Junjie Luo[3], Kunpeng Wang[1,2], and Zhenfei Zhang[4]

[1] SKLOIS, Institute of Information Engineering, CAS
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] Nanyang Technological University
[4] Manta network, Boston, USA

Abstract. In this paper, we study the *hybrid dual attack* over Learning with Errors (LWE) problems for *any* secret distribution. Prior to our work, hybrid attacks are only considered for sparse and/or small secrets. A new and interesting result from our analysis shows that a hybrid dual attack can outperform a standalone dual attack, regardless of the secret distribution. We formulate our results into a framework of predicting the performance of the hybrid dual attacks. We also present a few tricks that further improve our attack. To illustrate the effectiveness of our result, we re-evaluate the security of *all* LWE related proposals in round 3 of NIST's post-quantum cryptography process, and improve the state-of-the-art cryptanalysis results by 1-9 bits, under the BKZ-core-SVP model.

Keywords: Learning with Errors, Lattice-based Cryptography, Cryptanalysis, Dual Attack, Hybrid attack, NIST PQC

1 Introduction

The Learning with Errors (LWE) problem, introduced by Regev [34] in 2005, is one of the most important problems in lattice-based cryptography. A variety of schemes, from public key encryptions and digital signatures to homomorphic encryptions, base their security on LWE family of the lattice problems. The LWE problem and its variants are conjectured to be hard to solve, even with a quantum computer. The schemes that base their security on LWE problems, are therefore, considered quantum-safe. Indeed, LWE and its variants contribute to 5 out of 15 schemes in round 3 of National Institute of Standards and Technology's post-quantum cryptography standardization process (NIST-PQC), namely Dilithium[23], Kyber[14], Saber[22], Frodo[13] and NTRULPrime[10]. This process has sparked a long list of cryptanalytic advancements [1,3,5,6,16,19,21,25,35], and is still calling for a better understanding of the concrete security of LWE and its variant problems.

Informally, the search version of LWE asks to recover a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, given a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ and a vector $\mathbf{b} \in \mathbb{Z}_q^m$ such, that $\mathbf{As} + \mathbf{e} = \mathbf{b} \bmod q$ for

⋆ bilei121@outlook.com

a short error vector $\mathbf{e} \in \mathbb{Z}_q^m$ sampled from some error distribution. The decision version LWE asks to distinguish between an LWE instance $(\mathbf{A}, \mathbf{b})$ and uniformly random $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$.

In the survey paper [6], Albrecht et al. summerized three strategies to analyze the concrete hardness of LWE:

- The first one tries to recover the secret directly, for example, the algebraic attack (i.e., using the Arora-Ge algorithm) [9,2] or exhaustive search.
- The second method tries to view an LWE problem as a Bounded Distance Decoding (BDD) problem. There are two subsequent attacks: the decoding attack (i.e., using the Nearest Plane algorithm) [32] and the primal attack [5].
- The last strategy solves decisional LWE by reducing it to a Short Integer Solutions (SIS) problem. There are also two subsequent attacks: the combinatorial attack (i.e., using BKW algorithm) [4] and the dual attack [1].

In a later paper, Albrecht et al. [3] studied the security of all lattice-based schemes from round 1 candidates of NIST-PQC, and concluded that the primal attack and the dual attack are the most effective ones from the cryptanalysis standpoint.

The primal attack is to find the closest lattice vector to $\mathbf{b}$ in the lattice spanned by the columns of $\mathbf{A}$ mod q [32] via bounded distance decoding. Then, one reduces the BDD problem to a unique Shortest Vector Problem (uSVP) in a higher dimension lattice via some embedding, and solves the uSVP with lattice reductions (e.g., BKZ [18]). The lattice, as of our cryptanalysis interest, is then denoted by

$$\Lambda_{\text{primal}} = \{\mathbf{x} \in \mathbb{Z}^{m+n+1} | (\mathbf{A}|\mathbf{I}_m|\mathbf{b})\mathbf{x} = \mathbf{0} \bmod q\}.$$

The dual attack is to solve the (Inhomogeneous) Short Integer Solutions ((I)SIS) problem, i.e., using a lattice reduction algorithm to find short vectors $\mathbf{w}$ or $(\mathbf{w}, \mathbf{v})$ in the following lattice:

$$\Lambda_{\text{dual}}^{\perp} = \{\mathbf{w} \in \mathbb{Z}^m : \mathbf{w} \cdot \mathbf{A} = \mathbf{0} \bmod q\},$$
$$\Lambda_{\text{dual}}^{E} = \{(\mathbf{w}, \mathbf{v}) \in \mathbb{Z}^m \times \mathbb{Z}^n : \mathbf{w} \cdot \mathbf{A} = \mathbf{v} \bmod q\}.$$

This allows one to distinguish an LWE sample $\mathbf{b}$ from a uniform vector $\mathbf{u}$ since $\langle \mathbf{w}, \mathbf{b} \rangle = \langle \mathbf{v}, \mathbf{s} \rangle + \langle \mathbf{w}, \mathbf{e} \rangle$ is small when $\mathbf{w}$, $\mathbf{v}$, $\mathbf{s}$ and $\mathbf{e}$ are all short [7].

One may additionally combine the above attacks with guessing. This method is known as the *hybrid attacks* in the literature [1,16,19,25,28,30,35,37,38]. Informally, a hybrid attack guesses part of the secret and performs some attack on the remaining part. As guess reduces the dimension of the problem, the cost of the lattice attack on the remaining part is reduced. Moreover, in general, the lattice attack component is reusable for multiple guesses; an optimal attack is achieved when the cost of guessing matches the cost of lattice attack. For simplicity, we refer to hybrid attacks where the lattice attack component is a primal attack as the *hybrid primal attack*, and accordingly, the *hybrid dual attack*.

Let us start with a typical example: we assume, with probability p, the attacker is able to guess all the entries for the guessing components. The cost of the hybrid attacks becomes that of the lattice attack components (with a success rate p). For (sparse) binary/ternary secrets, this strategy works well. For hybrid primal attacks over other secret distributions, there are mainly two obstacles. First, for secrets with more entropy, such as Gaussian, p will be reduced significantly with the increase of guessing dimension. Second, one need to solve a CVP (a decoding problem) rather than a uSVP (a primal attack) after guessing (see [35] for more details about the reduction). As a rule of thumb, a decoding attack requires better reduced lattice than a primal attack. Due to the above drawbacks, hybrid primal attacks are considered less efficient than standalone primal attacks when dealing with none (sparse) binary/ternary secrets.

Now let us turn to the focus of this paper: hybrid dual attacks. They differ from the hybrid primal attacks in that, after a guess, the resulting lattice component becomes a new LWE lattice with a smaller dimension; and the LWE lattice remains the same for all guesses. Note that the attacker does not need to solve a decoding problem. In other words, the second obstacle for the hybrid primal attack is no longer an issue for hybrid dual attack. Nonetheless, the community seems to have presumed the obstacles for the hybrid dual attack, and applying it over LWE with arbitrary secrets therefore remains a blind spot prior to this paper.

Related work. The very first hybrid attack was proposed by Howgrave-Graham [31] to analyze NTRU [29]. In the recent years, hybrid attacks have been extensively studied for LWE with sparse and/or small sparse secrets. We summarize those results in Table 1. The first work of hybrid attack on LWE [16] combined decoding attack with meet-in-the-middle (MITM) technique. Then a similar approach was conducted on primal lattices [35]. Albrecht [1] proposed the framework of hybrid dual attack and applied it over LWE with sparse and binary/ternary secrets. Cheon et al. [19] improved guessing in this attack via an MITM technique. We note that in a hybrid dual attack, the secret and errors will increase significantly. Therefore, the proposed MITM technique requires a gigantic modulus q to incorporate the new, larger error. Recently, Espitau et al. [25] proposed a further optimization for guessing, via an efficient matrix multiplication exploiting the recursive structure of the matrix whose columns form the whole guessing space.

Table 1. Hybrid attacks on LWE

	Lattice	Guessing	Secret
[16]	Decoding	MITM	Small
[35]	Decoding + Primal	MITM	Small + sparse
[1]	Dual	Pruning	Small + sparse
[19]	Dual	MITM	Small + sparse
[25]	Dual	Matrix Mul.	Small
This paper	Dual	Opt. Pruning + Mat. Mul.	Arbitrary

1.1 Contribution

In this work, we study the hybrid dual attack on LWE with *arbitrary* secrets. Our contributions are two-fold. From the theory side, we analyze the hybrid dual attack in details, and develop the following observation:

Hybrid dual attacks can out-perform dual attacks for LWE instances with arbitrary secrets.

This observation is based on a quite interesting and surprising phenomenon in our analysis that when the guessing dimension (r) increases, the BKZ blocksize (β) indeed reduces. We formulate this phenomenon into the following lemma.

Theorem 1 (Informal). *For a hybrid dual attack under the core-SVP model, for most cryptographic use cases, if we increase the guessing dimensions r, the minimum BKZ blocksize β that maintains a same level of success rate will be reduced.*

We will provide our intuition shortly. The proof will be present in Section 3.3. To quantify this effect, we make an additional Heuristic 2 with justification in Section 3.4.

For LWE with *short* secrets, it is straightforward to see that the observation is implied by Theorem 3 as long as Theorem 1 and Heuristic 2 hold. For LWE with *large* secrets, when enough LWE samples are given, we normalize it and invoke Theorem 3. The only remaining case is LWE with large secrets and limited samples, for which we prove separately in Section 4.

We also propose a few tricks that further improve the guessing complexity. This allows us to develop an estimator that may be of independent interest (our estimator is open sourced on Github[5]. For example, one may apply our estimator to other LWE based schemes, such as FHE [26,15,27] or lattice-based ZK proofs [12,24,11].

From the practical side, we re-evaluated *all* LWE-related candidates of NIST-PQC round 3, namely, Dilithium [23], Kyber [14], Saber [22], Frodo [13] and NTRULPrime [10]. Our results are summarized in Table 2. We improve state-of-the-art cryptanalyze results by 1-10 bits[6] for these candidates, under the classical/quantum core-SVP model [7,6,3]. We will give more details on the estimations in Section 7.

Our technique. Our baseline for comparison is the standalone dual attack. In combination with the dual attack, we propose two hybrid attacks, namely, Hybrid 1 and Hybrid 2, vary in the strategy to conduct searching.

[5] https://github.com/BiLei121/hybrid-dual-estimator.

[6] NIST-PQC process has been running for 4 years. Finalists (and also the alternate candidates) and their parameters are considered mature and stable, and the security estimations are fairly conservative: even a few bits improvement on an individual candidate may be considered as a valid contribution.

Table 2. Bit-security estimations under Core-SVP Model

Name	Security Level	Classical			Quantum		
		Dual	Ours	Δ	Dual	Ours	Δ
Kyber512	1	112	108	-4	101	99	-2
Kyber768	3	182	176	-6	165	161	-4
Kyber1024	5	254	245	-9	231	225	-6
Saber512	1	117	115	-2	107	105	-2
Saber768	3	189	184	-5	172	169	-3
Saber1024	5	258	250	-8	235	230	-5
Dilithium768	1	100	99	-1	91	90	-1
Dilithium1024	2	142	140	-2	129	128	-1
Dilithium1280	3	175	172	-2	158	157	-1
Frodo640	1	142	139	-3	129	127	-2
Frodo976	3	206	202	-4	187	185	-2
Frodo1344	5	271	264	-6	245	242	-3
NTRULPrime653	1	131	125	-5	118	115	-3
NTRULPrime761	2	155	148	-7	141	137	-4
NTRULPrime857	3	177	168	-9	160	155	-5

* Data for "Ours" uses Hybrid 2m estimator.
* For a fair comparison, data for "Dual" also comes from our estimator. Our estimated results closely match the reported bit-security from their NIST-PQC documentations, with a maximum difference of 1 bits.

We first compare the standalone dual attack with Hybrid 1, which exhaustively searches all candidates from the guessing space. We show that for most cryptographic use cases we can select a proper guess dimension for Hybrid 1 such that the overall cost is reduced. Therefore, Hybrid 1 can outperform the dual attack, *regardless the secret distribution*. We further assert that optimal blocksize of the BKZ decreases linearly as the guess dimension increases, i.e., Heuristic 2, and use BKZ simulator to validate this assertion. This allows us to derive a formula to estimate the improvement of Hybrid 1 compared to the dual attack on *arbitrary* secrets.

Before proceeding further, let us give our intuition of Theorem 1. When the guessing dimension r is increased, the determinant of the lattice in the hybrid attack will be reduced. Hence, we can use a larger root Hermite factor (which implies a smaller β) to produce a short vector, denoted by $(\mathbf{w}, \mathbf{v})$, of a similar ℓ_2-norm. Note that although each coefficient of $(\mathbf{w}, \mathbf{v})$ indeed increases, the ℓ_2-norm remains unchanged (since the lattice dimension drops). From a dual attack's standpoint, Lemma 2 says that the advantage only cares about the ℓ_2-norm of $(\mathbf{w}, \mathbf{v})$, rather than its individual coefficients. Hence, so long as this ℓ_2-norm remains stable, the success rate of the dual attack component is intact. We also remark that this is a key difference between a hybrid primal attack and a hybrid dual attack.

Our Hybrid 2 further improves upon Hybrid 1 with *optimal pruning*. This method works for center limited distributions that are common to most cryptosystems. Note that a main obstacle of hybrid dual attack for general secrets is the large secret space. The subtlety here is to find a better approach to guess instead of exhaustively searching. Straightforward methods, such as partitioning the search space, reduce the success probability of the attack (significantly). Our Hybrid 2 with a fine-tuned pruning allows for a high success probability over a fixed number of secrets; while having a minimal impact on the overall cost.

To achieve so, we present an algorithm to guess the secret with *optimal* success probability when the number of guesses is bounded. More precisely, we partition the secret space into ordered classes, sorted by the probability of a candidate being the correct secret. Then we greedily choose candidates from the class with the highest probability when the number of guesses permits. We give a theoretical analyses of this approach, as well as its impact on Hybrid 2; and show the advantage of Hybrid 2 over Hybrid 1.

As an orthogonal line of optimization, we also give an efficient algorithm for matrix multiplication which can be seen as a none-trivial generalization of the algorithm in [25]. Our improved algorithm decreases the computation time for each guess; consequently, we increase the number of guesses, given a fixed cost model. To be a bit more specific, assuming an integer multiplication takes a unit time, for an $M \times r$ matrix of arbitrary entries, and a $r \times \ell^r$ matrix whose columns consist of all vectors from Q^ℓ, where Q is a set of ℓ numbers, [25]'s algorithm improves the matrix multiplication cost from $\mathcal{O}(M \cdot \ell^r \cdot r)$ to $\mathcal{O}(M \cdot \ell^r)$. However, this algorithm is only applicable to matrices whose columns form the whole guessing space without pruning. We generalize it to all *closed matrices* (see Def. 3). We remark that this optimization can be used for both Hybrid 1 and Hybrid 2. We refer to the attacks with this additional optimization by Hybrid 1m and Hybrid 2m.

We conclude this section with a final remark. The advantage of Hybrid 1 and Hybrid 2 over standalone dual attack is independent of the underlying BKZ cost model. For example, Hybrid 1 will always out-perform dual attack, for core-SVP model, Practical model, or Frodo model (see Section 2.2 for definitions); the actual gain will vary depending on the cost model, nonetheless. For consistency, we will adopt the core-SVP model throughout the rest of the paper, unless otherwise stated.

2 Preliminaries

2.1 Notations

Logarithms are base 2 if not stated otherwise. We write ln for the natural logarithm. We denote vectors in bold, e.g. $\mathbf{v}$ and matrices in upper-case bold, e.g. $\mathbf{A}$. For a vector $\mathbf{v}$ The Euclidean norm of a vector $\mathbf{v} \in \mathbb{R}^m$ is $||\mathbf{v}||$. We denote by $\langle \cdot, \cdot \rangle$ the usual dot product of two vectors. For a compact set $S \in \mathbb{R}^n$, we denote by $\mathcal{U}(S)$ the uniform distribution over S.

2.2 Lattices and lattice reductions

Lattice. A lattice is a discrete additive subgroup of $\mathbb{R}^m$ for some $m \in \mathbb{N}$. In this case, m is called the *dimension* of the lattice. A lattice Λ is generated by a basis

$\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^m$ which is a set of n linearly independent row vectors and $\Lambda = \Lambda(\mathbf{B})$ can be represented as $\Lambda(\mathbf{B}) = \mathbf{B} \cdot \mathbb{Z}^m = \left\{\sum_{i \in [n]} z_i \cdot \mathbf{b}_i : z_i \in \mathbb{Z}\right\}$. We say that the *rank* of the lattice is n and its *dimension* is m. If $n = m$, the lattice is called a *full-rank lattice*. For the lattice $\Lambda = \Lambda(\mathbf{B})$, its *fundamental parallelepiped* is defined as $\mathcal{P}(\mathbf{B}) = \mathbf{B} \cdot [-\frac{1}{2}, \frac{1}{2})^n = \left\{\sum_{i \in [n]} c_i \cdot \mathbf{b}_i : c_i \in [-\frac{1}{2}, \frac{1}{2})\right\}$. The determinant of $\Lambda = \Lambda(\mathbf{B})$ denoted by $det(\Lambda)$ is defined as the m-dimensional volume of its fundamental parallelepiped.

A non-zero vector in a lattice Λ that has the minimum norm is named as the *shortest vector*. The norm of the shortest vector is denoted as $\lambda_1(\Lambda) = \min_{\mathbf{v} \in \Lambda, \mathbf{v} \neq 0} ||\mathbf{v}||$.

Lattice reductions. When given as input some basis of a lattice, a lattice reduction algorithm is to find a basis that consists of relatively short and relatively pairwise orthogonal vectors. The quality of basis returned by a lattice reduction algorithm is characterized by the *Hermite factor* δ_0^m:

$$\delta_0^m = \frac{||\mathbf{b}_1||}{det(\Lambda)^{\frac{1}{m}}},$$

where $\mathbf{b}_1$ is the first vector in the output basis. Refer to δ_0 itself, we call it the root-Hermite factor.

The BKZ algorithm [18] is a commonly used lattice reduction algorithm.

Heuristic 1. *BKZ with blocksize β yields root-Hermite factor*

$$\delta_0 \approx \left(\frac{\beta}{2\pi e}(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}}.$$

This heuristic is experimentally verified in [17].

BKZ cost models. To estimate the runtime of BKZ, there are several different cost models. The main differences between them are (1) whether they choose sieving or enumeration as the SVP oracle and (2) how many calls to the SVP oracle are expected to produce a vector of length $\delta_0^m \cdot det(\Lambda)^{\frac{1}{m}}$, where δ_0 is the root-Hermite factor, m is the dimension of lattice Λ. See [3] for more details.

Let us firstly list relevant cost models in this paper. As mentioned earlier, we will be focusing on the core-SVP model with sieving [7].

$$\text{Core-SVP Model: } T_{\text{BKZ}}(m, \beta) = \begin{cases} 2^{0.292\beta}, & \text{classical} \\ 2^{0.265\beta}, & \text{quantum} \end{cases}$$

We will also briefly compare with two additional models: a practical model, used by, for exmaple [1], where the number of calls is $8m$ rather than 1.

$$\text{Practical Model: } T_{\text{BKZ}}(m, \beta) = \begin{cases} 8m \cdot 2^{0.292\beta+16.4}, & \text{classical} \\ 8m \cdot 2^{0.265\beta+16.4}, & \text{quantum} \end{cases}$$

and the Frodo model [13]

$$\text{Frodo Model: } T_{\mathrm{BKZ}}(m,\beta) = \begin{cases} \beta \cdot 2^{0.292\beta}, & \text{classical} \\ \beta \cdot 2^{0.265\beta}, & \text{quantum} \end{cases}$$

In addition, when using sieving as the SVP oracle, [7] also made an assumption on the number of output short vectors from BKZ.

Assumption 1. *When using sieving as the SVP oracle, BKZ algorithm with blocksize β provides $2^{0.2075\beta}$ short vectors in one run, and they are almost as short as the shortest one produced by BKZ algorithm.*

To justify, [7] pointed out that a sieving algorithm maintains a list of $2^{0.2075\beta}$ vectors. When the sieving algorithm terminates, the list of vectors should be of approximately same length as the final output. This assumption has been adopted in most of the subsequent cryptanalysis work in this field.

2.3 The Learning with Errors problem

The Learning with Errors (LWE) problem, introduced by Regev [34], is a computational problem, whose presumed hardness (against quantum computers) gives rise to a large numbers of cryptographic constructions.

Definition 1 (LWE). *Let $n, q \in \mathbb{N}$, $\mathcal{S}$ be an distribution over $\mathbb{Z}_q^n$ and $\mathbf{s} \leftarrow \mathcal{S}$ be a secret vector. Let χ be a small error distribution over $\mathbb{Z}$. Denote $LWE_{n,q,\mathbf{s},\chi}$ the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\mathbf{a} \in \mathbf{Z}_q^n$ uniformly at random, choosing $e \xleftarrow{\$} \chi$ and returning $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.*

Given access to the outputs from $LWE_{n,q,\mathbf{s},\chi}$ distribution, we define two following problems:

- *Decision-LWE. Given m instances, distinguish $\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ and $LWE_{n,q,\mathbf{s},\chi}$ distribution for a fixed $\mathbf{s} \leftarrow \mathcal{S}$.*
- *Search-LWE. Given m instances sampled from $LWE_{n,q,\mathbf{s},\chi}$ distribution with fixed $\mathbf{s} \leftarrow \mathcal{S}$, recover $\mathbf{s}$.*

The LWE instances can be presented in the matrix form as follows:

$$(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q) \tag{1}$$

with $\mathbf{s} \leftarrow \mathcal{S}, \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}, \mathbf{e} \xleftarrow{\$} \chi^m, \mathbf{b} \in \mathbb{Z}_q^m$.

There is an useful lemma shows that given instances from $LWE_{n,q,\mathbf{s},\chi}$ with $\mathbf{s} \in \mathbb{Z}_q^n$, we can construct *normal-form* LWE instances, i.e., the secret follows the error distribution.

Lemma 1 ([8]). *Given the instances $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e)$ sampled from $LWE_{n,q,\mathbf{s},\chi}$ with $\mathbf{s} \in \mathbb{Z}_q^n$, we can construct instances of the form $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{e} \rangle + e)$ with $\mathbf{e} \xleftarrow{\$} \chi^n$ and $e \xleftarrow{\$} \chi$ at the loss of n instances overall.*

In this paper, we will also be dealing with LWE variant problems, such as Ring-LWE, module-LWE and module-LWR. We will treat those problems as LWE problems, following prior cryptanalysis.

Secret distributions. Practical LWE (and its variants) based cryptosystems utilize various secret and error distributions. To list a few,

- $\mathcal{B}^+$ the distribution on $\mathbb{Z}_q^n$ where each component is independently sampled uniformly at random from $\{0,1\}$.
- $\mathcal{B}^-$ the distribution on $\mathbb{Z}_q^n$ where each component is independently sampled uniformly at random from $\{-1,0,1\}$.
- $\mathcal{B}_h^+$ the distribution on $\mathbb{Z}_q^n$ where each component is independently sampled uniformly at random from $\{0,1\}$ with the additional guarantee that the number of 1s is h.
- $\mathcal{B}_h^-$ where each component is independently sampled uniformly at random from $\{-1,0,1\}$ with the additional guarantee that the number of 1s and -1s are both h.

In this paper, we divide the existing secret distributions into two categories:

1. binary/ternary secret with fixed hamming weight,
2. general central discrete distribution (without fixed hamming weight):

Value	0	± 1	± 2	$\cdots$	$\pm t$
Probability	p_0	p_1	p_2	$\cdots$	p_t

Note 1. If the number of values is infinite (e.g. the Gaussian distribution), we truncate the distribution at a suitable place (also denoted by $\pm t$). Looking ahead, we will treat $\mathcal{B}^+$ as a category 2 distribution. It shares a same behavior as a central limited distribution for our analysis.

2.4 Best known attacks on LWE

To date, primal attacks and dual attacks are considered best known attacks against LWE and it variants. Their complexity are approximately the same for most cryptosystems.

Primal Attack. As mentioned in the introduction, the primal attack is to solve the search version LWE by viewing it as a Bounded Distance Decoding (BDD) problem. Then the attack reduces it to the unique Shortest Vector Problem (uSVP) via certain embedding technique, and solves uSVP with lattice reduction. We skip the details, since we will not focus on primal attacks in this paper.

Dual Attack. The dual attack, introduced by Micciancio and Regev [33], is to solve a decision-LWE by reducing it to a Shortest Integer Solution (SIS) problem, i.e., trying to find short vectors in the lattice

$$\Lambda_{\text{dual}}^{\perp} = \{\mathbf{w} \in \mathbb{Z}^m : \mathbf{w} \cdot \mathbf{A} = \mathbf{0} \bmod q\} .$$

If the input instances are from the $\mathrm{LWE}_{\mathbf{s},\sigma}$, then, $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$. In this case, given a short vector $\mathbf{w}$, we have

$$\langle \mathbf{w}, \mathbf{b} \rangle = \mathbf{w} \cdot (\mathbf{A}\mathbf{s} + \mathbf{e}) = \langle \mathbf{w}, \mathbf{e} \rangle \bmod q,$$

which will be short/small. Otherwise, $\langle \mathbf{w}, \mathbf{b} \rangle$ is uniform on $[-\frac{q}{2}, \frac{q}{2})$. With sufficient number of distinct $\mathbf{w}$ vectors, this attack can distinguish these two distributions with high probability.

Alkim et al. [7] presented an improved dual attack on normal-form LWE, which tries to solve an inhomogeneous SIS problem, and works over the *embedded lattice*:

$$\Lambda^E_{\mathrm{dual}} = \{(\mathbf{w}, \mathbf{v}) \in \mathbb{Z}^m \times \mathbb{Z}^n : \mathbf{w} \cdot \mathbf{A} = \mathbf{v} \bmod q\}.$$

Following the same strategy, if the instances are from the normal-form $\mathrm{LWE}_{\mathbf{s},\sigma}$, then we have

$$\langle \mathbf{w}, \mathbf{b} \rangle = \mathbf{w} \cdot (\mathbf{A}\mathbf{s} + \mathbf{e}) = \langle \mathbf{v}, \mathbf{s} \rangle + \langle \mathbf{w}, \mathbf{e} \rangle \bmod q,$$

the right part of the equation is small as $\mathbf{s}$ and $\mathbf{e}$ are both small.

In general, $(\mathbf{w}, \mathbf{v}) \in \Lambda^E_{\mathrm{dual}}(\mathbf{A})$ is produced by BKZ. There is an assumption on the quality of this vector.

Assumption 2 ([20,25])**.** *The coordinates of vectors produced by lattice reduction algorithms are balanced, i.e., each coordinate of* $(\mathbf{w}, \mathbf{v}) \in \mathbb{Z}^m \times \mathbb{Z}^n$ *follows a Gaussian distribution of mean 0 and standard deviation* $\frac{\ell}{\sqrt{m+n}}$*, where* $\ell = ||(\mathbf{w}, \mathbf{v})||$.

Under this assumption, the distribution of $t := \langle \mathbf{w}, \mathbf{b} \rangle$ can be viewed as a Gaussian distribution $\mathcal{G}_\rho$ with mean 0 and standard deviation $\rho = \ell\sigma$ [7]. Then the maximal variance distance between $\mathcal{G}_\rho$ and $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$ is bounded by $\varepsilon = 4\exp(-2\pi^2\tau^2)$, where $\tau = \ell\sigma/q$ [7]. According to these, the advantage of the attack is shown in the following lemma.

Lemma 2 ([7])**.** *Given m normal-form LWE instances* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \ \mathit{mod}\ q)$ *characterized by n, σ, q, and a vector* $(\mathbf{w}, \mathbf{v}) \in \Lambda^E_{dual}$ *of length ℓ, the dual attack solves the decision-LWE with advantage* $\varepsilon = 4\exp(-2\pi^2\tau^2)$ *where* $\tau = \frac{\ell\sigma}{q}$.

Then the success probability of the attack can be amplified by using about $1/\varepsilon^2$ many such vectors $(\mathbf{w}, \mathbf{v}) \in \Lambda^E_{\mathrm{dual}}$ of length ℓ. By Assumption 1, when using sieving as the SVP oracle, the attack needs to repeat BKZ $\lceil \frac{1}{2^{0.2075\beta}\varepsilon^2} \rceil$ times.

The analysis of Alkim et al. [7] about the cost and success probability of this attack does not specify how to distinguish the Gaussian distribution from the uniform to get the desired advantage (which equals to the statistical distance between the two distributions) and how to amplify this advantage. we present concrete algorithms in Appendix B to make the conclusion complete.

This attack [7] was introduced for the normal-form LWE. When the secret does not follow the error distribution, the attack also works by using the scaling technique introduced by Albrecht [1]. For the remaining part of this paper, we use this technique when it is needed and will not mention it separately.

Note that the [33] dual attack (referred to as *original dual attack*) works for arbitrary secrets; while the [7] dual attack (referred to as *embedded dual attack*) requires the secret to be somewhat short, so that $\langle \mathbf{v}, \mathbf{s} \rangle$ is small and distinguishable from uniform. Nonetheless, for practical cryptosystems (all NIST-PQC candidates use small secrets) the embedded dual attack is more efficient than the original dual attack. Therefore, for the remaining part of the paper, a (hybrid) dual attack stands for a (hybrid) embedded dual attack, unless otherwise stated.

3 Hybrid attack on short secrets

Now we are ready to proceed to our hybrid dual attack. We start with a naive strategy where we conduct "guess" via exhaustive search. We name this strategy Hybrid 1. We will be comparing intensively Hybrid 1 with a standalone dual attack.

3.1 The framework

A hybrid attack has two components, a lattice reduction phase and a guessing phase. We start with the lattice reduction phase. Given m LWE instances $(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \bmod q)$ as input, we divide the secret vector $\mathbf{s}$ and public matrix $\mathbf{A}$ into two parts, parameterized by r:

$$\mathbf{s} = \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} \in \mathbb{Z}_q^r \times \mathbb{Z}_q^{n-r}, \qquad \mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2) \in \mathbb{Z}_q^{m \times r} \times \mathbb{Z}_q^{m \times (n-r)}.$$

Looking ahead, our guessing phase works over vectors of dimension r, and tries to identify the coefficient of $\mathbf{s}_1$.

Similar to the dual attack, we define a lattice over $\mathbf{A}_2$:

$$\Lambda_{\text{dual}}^E(\mathbf{A}_2) = \left\{ (\mathbf{w}, \mathbf{v}) \in \mathbb{Z}^m \times \mathbb{Z}^{n-r} : \mathbf{w} \cdot \mathbf{A}_2 = \mathbf{v} \bmod q \right\}.$$

$\Lambda_{\text{dual}}^E(\mathbf{A}_2)$ has a dimension of $d = m + n - r$ and a volume of q^{n-r} w.h.p. Then, we assume that with lattice reduction algorithms we will obtain some short vector(s) $(\mathbf{w}, \mathbf{v}) \in \Lambda_{\text{dual}}^E$ that allow us to calculate $\langle \mathbf{w}, \mathbf{b} \rangle$ as

$$\begin{aligned} \langle \mathbf{w}, \mathbf{b} \rangle &= \mathbf{w}(\mathbf{A}\mathbf{s} + \mathbf{e}) \\ &= \mathbf{w}\mathbf{A}_1\mathbf{s}_1 + \mathbf{w}\mathbf{A}_2\mathbf{s}_2 + \langle \mathbf{w}, \mathbf{e} \rangle \\ &= \mathbf{w}\mathbf{A}_1\mathbf{s}_1 + \langle \mathbf{v}, \mathbf{s}_2 \rangle + \langle \mathbf{w}, \mathbf{e} \rangle \bmod q. \end{aligned}$$

This can be seen as a new LWE instance $(\hat{\mathbf{a}}, \hat{b} = \langle \hat{\mathbf{a}}, \mathbf{s}_1 \rangle + \hat{e})$, where

$$\begin{aligned} \hat{b} &= \langle \mathbf{w}, \mathbf{b} \rangle \bmod q, \\ \hat{\mathbf{a}} &= \mathbf{w}\mathbf{A}_1 \bmod q, \\ \hat{e} &= \langle \mathbf{v}, \mathbf{s}_2 \rangle + \langle \mathbf{w}, \mathbf{e} \rangle \bmod q. \end{aligned} \tag{2}$$

Next we proceed to the guessing phase. Denote by $\tilde{\mathbf{s}}_1$ a candidate from the guessing space. Then, $\hat{e} = \hat{b} - \langle \hat{\mathbf{a}}, \tilde{\mathbf{s}}_1 \rangle \bmod q$ is from a Gaussian distribution if $\tilde{\mathbf{s}}_1$ is a correct guess. Otherwise $\hat{e}$ must follow the uniform distribution on $\mathbb{Z}_q$.

In order to recover $\mathbf{s}_1$ completely, we will require a large number of short vectors from $\Lambda^E_{\text{dual}}(\mathbf{A}_2)$. This can be obtained from the lattice reduction phase, assuming Assumption 1.

We present the pseudo-code of the attack in Algorithm 1. Here we denote M the number of short vectors we need to sample from the dual lattice and denote N the number of calls to BKZ. Both values will be discussed in Section 3.2. In addition, we denote C a collection of the selected candidates $\tilde{\mathbf{s}}_1$ and let $L = |C|$.

Algorithm 1: Hybrid Dual Attack

Input: $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m, r \in \mathbb{Z}$
Output: LWE distribution or Uniform

1 $\mathbf{P} \overset{\$}{\leftarrow}$ permutation matrix;
2 $(\mathbf{A}_1, \mathbf{A}_2) \leftarrow \mathbf{A} \cdot \mathbf{P}$ with $\mathbf{A}_1 \in \mathbb{Z}_q^{m\times r}$ and $\mathbf{A}_2 \in \mathbb{Z}_q^{m\times(n-r)}$;
3 M short vectors $(\mathbf{w}_i, \mathbf{v}_i)_{i\in[M]} \leftarrow N$ calls to BKZ on $\Lambda^E_{\text{dual}}(\mathbf{A}_2)$;
4 **for** $i \in \{1, \cdots, M\}$ **do**
5   calculate $\hat{b}_i = \langle \mathbf{w}_i, \mathbf{b} \rangle \bmod q$ and $\hat{\mathbf{a}}_i = \mathbf{w}_i \mathbf{A}_1 \bmod q$; ▷ Eq.(2)
6 **for** *each* $\tilde{\mathbf{s}}_1 \in C$ **do** ▷ C is defined in Section 5.1
7   **for** $i \in \{1, \cdots, M\}$ **do**
8   calculate $\tilde{e}_i = \hat{b}_i - \langle \hat{\mathbf{a}}_i, \tilde{\mathbf{s}}_1 \rangle \bmod q$;
9 **if** $\tilde{e}_{i\in[M]}$ *follow Gaussian distribution* **then**
10   **return** *LWE distribution*;
11 **return** *Uniform;*

3.2 Analysis

The success probability of the attack is the product of two quantities:

1. $p_s :=$ the success probability of the distinguish algorithm,
2. $p_c :=$ the probability that C contains the right s_1.

We present the analysis of p_s in the remaining part of this section. The analysis of p_c is deferred to Section 5.1 as it depends on the specific secret distribution.

In Algorithm 1, The goal of lines 6-11 is to recover $\mathbf{s}_1$ using the new LWE instances. For each guessed candidate $\tilde{\mathbf{s}}_1$, we calculate the M distinct quantities $\tilde{e}_i$. If the input instances are from $\text{LWE}_{\mathbf{s},\sigma}$, the distribution of $\tilde{e}_i$ must follow a modular Gaussian distribution otherwise $\tilde{e}$ is uniform in $[-\frac{q}{2}, \frac{q}{2})$. In order to recover $\mathbf{s}_1$, we need to correctly identify the distribution for all candidates $\tilde{\mathbf{s}}_1 \in C$.

Denote $\tilde{p}_s$ the success probability of correctly guessing the distribution of one candidate $\tilde{\mathbf{s}}_1$, then the success probability of recovering $\mathbf{s}_1$ will be $\tilde{p}_s^L$. Similar to dual attack, using majority vote, we can amplify the success probability from $\frac{1}{2} + \frac{\varepsilon}{2}$ to $\tilde{p}_s = 1 - \exp\left(-\frac{\varepsilon^2}{2}M\right)$ by using M short vectors (see Lemma 9 in supplementary material B for more details). If we target a success probability of $p_s = 1 - \frac{1}{2^\kappa}$ for the hybrid dual attack, for a given security parameter κ, then we have $\tilde{p}_s^L \gtrapprox 1 - \frac{1}{2^\kappa}$ Therefore, we can derive M from $\left(1 - \exp\left(-\frac{\varepsilon^2}{2}M\right)\right)^L \approx 1 - \frac{1}{2^\kappa}$. As a result, when there are $M \approx \frac{\kappa + \ln L}{\varepsilon^2}$ short vectors $(\mathbf{w}_i, \mathbf{v}_i) \in \Lambda_{\text{dual}}^E(\mathbf{A}_2)$ of length ℓ, the success probability of Algorithm 1 is $p_s = 1 - \frac{1}{2^\kappa}$, where κ is the security parameter.

The cost of the attack is the sum of two main components:

1. $N \cdot T_{\text{BKZ}} := N$ calls to BKZ on $\Lambda_{\text{dual}}^E(\mathbf{A}_2)$,
2. $T_{\text{guess}} :=$ evaluate all L guesses $\tilde{\mathbf{s}}_1 \in C$ using the M instances,

According to Assumption 1, we need repeat the BKZ algorithm for $N = \lceil \frac{M}{2^{0.2075\beta}} \rceil$ times to produce M short vectors. If we use a naive way to evaluate all L guesses, we will have $T_{\text{guess}} = M \cdot L \cdot r$. We will give an improved algorithm for T_{guess} in Section 6.

In summary, we present the results formally as follows.

Theorem 2. *Given $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, the hybrid dual attack using Algorithm 1 can decide whether they are LWE instances $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e})$ mod q characterized by n, σ, q or they are from uniform distribution. The success probability $p = p_c \cdot p_s$, where p_c is presented in Section 5.1 and $p_s = 1 - \frac{1}{2^\kappa}$, where κ is a security parameter. The cost of dual attack is calculated as*

$$T = N \cdot T_{BKZ} + T_{guess},$$

where $N = \lceil \frac{M}{2^{0.2075\beta}} \rceil$ is the number of repeated times of the BKZ algorithm, $M = \frac{\kappa + \ln L}{\varepsilon^2}$ is the number of short vectors in the dual lattice, and $T_{guess} = M \cdot L \cdot r$ (see Section 6 for an improvement of T_{guess}).

Remark 1. In Algorithm 1, we first identify the distribution for each guess $\tilde{\mathbf{s}}_1 \in C$ independently, and then output "Uniform" if and only if all guesses are identified as "Uniform". An alternative approach is to use Algorithm 3 to identify the combination of M samples with L guesses in one shot. However, the advantage will then become $\frac{\varepsilon}{L}$. As a result, to achieve the success probability $p_s = 1 - \frac{1}{2^\kappa}$, we need to set $M = \frac{\kappa L^2}{\varepsilon^2}$, which will be worse than our adopted approach.

3.3 The advantage of the hybrid dual attack

We analyze the advantage of the hybrid dual attack by comparing the dual attack and Hybrid 1. Since we always set the probability $p_s = 1 - \frac{1}{2^\kappa}$ with $\kappa = 128$, it is safe to ignore p_s. Then we just need to compare the running time. Note that we can view dual attacks as a special case of hybrid dual attacks, with $r = 0$.

Let SV be the number of short vector provided by BKZ algorithm with blocksize β using sieving as the SVP oracle. We first show that for dual attack and Hybrid 1, under the optimal parameters, we should repeat the BKZ only once, i.e., $N = 1$. Moreover, the number of short vectors produced by sieving (SV) should be almost the same as the number of short vectors required (M) to achieve the desired success probability p_s.

Lemma 3. *Assuming $\beta > 50$, for a fixed r such that $T_{guess} \leq 2^{50} \cdot T_{BKZ}$,* [7] *the optimal β that minimizes $T_{\text{Hybrid 1}} = N \cdot T_{BKZ} + T_{guess}$ will satisfy $N = 1$, $\frac{SV}{2^{0.2075}} \leq M \leq SV$, and $\varepsilon^2 \leq 2^{-0.2\beta+7}$.*

Proof sketch. The full proof is deferred to supplementary material C.1. We first assume β is a real number and show that the optimal β will satisfy $M(\beta) = SV(\beta)$ and hence $N = 1$. Then the claim of the lemma follows when β has to be an integer. Let β^* be the real number such that $M(\beta^*) = SV(\beta^*)$. We consider two cases when $\beta \geq \beta^*$ and $\beta \leq \beta^*$, and show that in both cases the optimal β is β^*. The first case when $\beta \geq \beta^*$ is easy as in this case $N = \lceil \frac{M(\beta)}{SV(\beta)} \rceil = 1$. For the second case when $\beta \leq \beta^*$, we consider the continuous function $f(\beta)$ corresponding to $N \cdot T_{\text{BKZ}}$ defined as follows:

$$f(\beta) := \frac{M(\beta)}{SV(\beta)} \cdot T_{\text{BKZ}(\beta)} = \frac{M(\beta)}{2^{0.2075\beta}} \cdot 2^{0.292\beta} = M(\beta) \cdot 2^{0.0845\beta}.$$

We can show that $f(\beta)$ is decreasing in β. Then the optimal β minimizing $N \cdot T_{\text{BKZ}}$ is the maximum β such that $\beta \leq \beta^*$, i.e., the optimal β is β^*. The upper bound for ε^2 is due to $M = \frac{\kappa + \ln L}{\varepsilon^2} = SV = 2^{0.2075\beta}$. □

Next, we study the influence of the guessing dimension r on the number of required short vectors M. In Hybrid 1 when we guess r dimensions, the benefit is that the determinant of the dual lattice is decreased. If we use a same β as for dual attack, the length ℓ of the short vectors produced by sieving is decreased, which will help to increase the advantage ε and decrease M.

On the other hand, when we guess r dimensions with L candidates, for each candidate we need to achieve a higher success probability (than p_s) such that the overall success probability is p_s. For this, we will need more short vectors to amplify the success probability, i.e., M will be increased.

These two opposite effects can be seen from the calculation of $M = \frac{\kappa + \ln L}{\varepsilon^2}$, where both ε and L increase when r increases. The key problem is how does M change when r increases. Our simulations show that M decreases when r increases for all 5 schemes tested in Section 7. This can be intuitively explained by that $\ln L = r \ln R$, where R is the size of the support for each entry of the secret, is increasing linearly in r while ε^2 is increasing exponentially in r (from $2^{-\mathcal{O}(n)}$ when $r = 0$ to $\mathcal{O}(1)$ when $r = n$).

[7] This guarantees that we don't guess too much. In practice, we usually have $T_{\text{guess}} \leq T_{\text{BKZ}}$. For example, all 5 schemes tested in Section 7 have $T_{\text{guess}} \leq T_{\text{BKZ}}$ under the optimal parameters. So it is safe to assume that $T_{\text{guess}} \leq 2^{50} \cdot T_{\text{BKZ}}$.

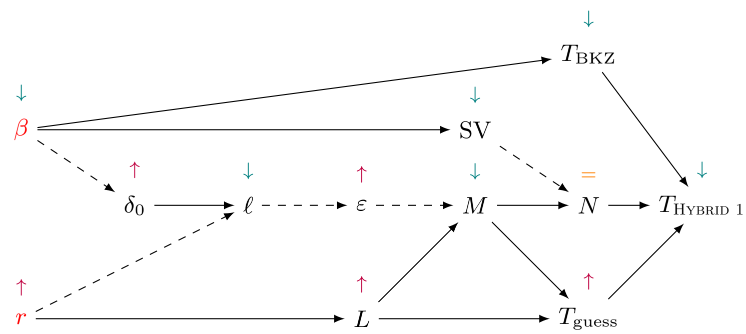


Fig. 1. Parameter relations and their value changes from dual attack to Hybrid 1. An arrow "→" (respectively, "⇢") from node A to node B means that increasing A will increase (respectively, decrease) B. "↑" and "↓" shows the direction that the values change from the dual attack to Hybrid 1 when r is increased and β is decreased while maintaining $M \approx SV$ and $N = 1$ unchanged.

We assume for now that M is decreasing in r and use this to explain why Hybrid 1 outperform dual attack. At the end of this section, we will show that this condition, M is decreasing in r, is satisfied under mild assumptions.

Lemma 4. *Assume M is decreasing in r (when β is fixed), then when we increase the guessing dimension r, the optimal BKZ blocksize β that minimizes $N \cdot T_{BKZ}$ and maintains a same level of success probability will be reduced.*

Proof. To ease analysis, we will take β as a real number (instead of an integer), and show that the optimal (real number) β will always be reduced when r increases. According to Lemma 3, the optimal β will always satisfy $N = 1$ and $M = SV$[8], which means that the optimal β will maintain $M = SV$ when we increase r. Since decreasing β will increase M and decrease $SV = 2^{0.2075\beta}$, and we assume that M will be reduced when r increases, to maintain $M = SV$, the optimal β will be reduced when r increases. □

Now we can explain why Hybrid 1 outperform dual attack. According to Lemma 3, we have $N = 1$ for dual attack and Hybrid 1 under the optimal choice of β. Hence, for dual attack we have $T_{\mathrm{dual}} = T_{\text{BKZ-d}}$ and for Hybrid 1 we have $T_{\text{Hybrid 1}} = T_{\text{BKZ-h}} + T_{\mathrm{guess}}$. Note that we can take dual attack as a special case of Hybrid 1 with $r = 0$ and $T_{\mathrm{guess}} = 0$. Compared with dual attack, in Hybrid 1 we can increase r and decrease β while maintaining $SV \approx M$ and $N = 1$. As a result, T_{BKZ} is decreased and T_{guess} is increased. As long as T_{guess} does not exceed T_{BKZ}, we can increase r almost "for free" (at the expense of at most one bit when $T_{\mathrm{guess}} = T_{\mathrm{BKZ}}$) and decrease β such that the overall running time

[8] Lemma 3 claims $\frac{SV}{2^{0.2075}} \le M \le SV$ as β is an integer. The proof of Lemma 3 shows that $M = SV$ when β is taken as a real number.

$T_{\text{HYBRID 1}} = T_{\text{BKZ-h}} + T_{\text{guess}}$ decreases. Our simulations show that the optimal r and β for HYBRID 1 will satisfy $T_{\text{BKZ}} \approx T_{\text{guess}}$. Figure 1 shows how parameter changes from dual attack to HYBRID 1.

Example. To give a more intuitive explanation of the advantage of the hybrid framework, we take Kyber512 as an example and use a figure to show how T_{BKZ}, T_{guess}, and $T_{\text{HYBRID 1}}$ change as r increases. In fact, T_{guess} and $T_{\text{HYBRID 1}}$ depend on both r and β. However, since we need to guarantee $N = 1$ (according to Lemma 3) to minimize the the total cost $T_{\text{HYBRID 1}}$, the value of β can be determined once the value of r is chosen. This allows us to estimate T_{BKZ}, T_{guess}, and $T_{\text{HYBRID 1}}$ as functions of r. The results are shown in Figure 2. As expected, as r increases (and β decreases), T_{guess} increases and T_{BKZ} decreases. Hence, as r increases, $T_{\text{HYBRID 1}}$ first deceases and then increases, and the optimal $T_{\text{HYBRID 1}}$ is achieved when the two lines cross. From Figure 2, we can see that the cross point ($T_{\text{HYBRID 1}}$) is smaller than the starting point, which has $r = 0$ and represents a standalone dual attack.

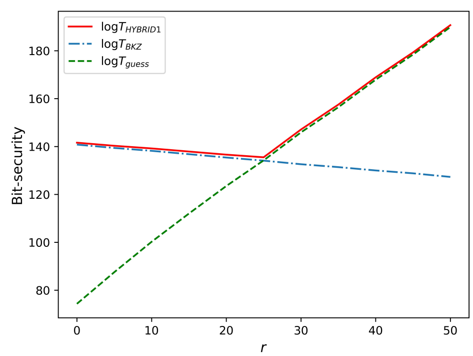


Fig. 2. Example: $T_{\text{HYBRID 1}}$, T_{BKZ} and T_{guess} for Kyber512.

***M* is decreasing in *r*.** To show that M indeed decreases when r increases, we need to make two minor assumptions. The first one is that $50 < \beta < 1990$, which implies that the cost of the BKZ is from 15 bits to 580 bits, and this covers most LWE instances we are interested in. Specifically, this covers all 5 schemes tested in Section 7, whose optimal β is in $(30, 1000)$. The second assumption is that the cost of guessing only one dimension should not exceed the cost of the standalone dual attack, as otherwise it is not helpful at all to use the hybrid framework. We state this formally in the following assumption.

Assumption 3. *Let T_{BKZ} be the cost of the standalone dual attack with the optimal β on a LWE instance, then the number M of short vectors needed for* *the attack and the number SV of short vectors produced by BKZ satisfy that $M \geq \frac{SV}{2^{0.2075}}$ (according to Lemma 3). Let R be the size of the support for each entry of the secret. Assume the cost of guessing only one dimension is less than T_{BKZ}, i.e., $R \cdot M \leq T_{BKZ}$, then*

$$R \leq \frac{2^{0.2075} \cdot T_{BKZ}}{SV} = 2^{0.0845\beta+0.2075}.$$

Now we can show that M decreases when r increases.

Lemma 5. *Assuming $50 \leq \beta \leq 1990$ and Assumption 3, the number M of short vectors required to achieve the success probability p_s is decreasing in the guessing dimension r.*

Proof sketch. The full proof is deferred to supplementary material C.2. To ease analysis, we will take β as a real number (instead of an integer). Once r and the corresponding optimal β are fixed, the optimal number m of equations to use is given by

$$m = \sqrt{\frac{(n-r)\log q}{\log \delta_0}} - (n-r)[33]^9.$$

So the number of samples we need is

$$F(r) := M(r) = \frac{\kappa + \ln L(r)}{\varepsilon^2(r)} = \frac{\kappa + \ln L(r)}{4e^{\frac{-4\pi^2\sigma^2(\delta_0^4)^{\sqrt{\frac{(n-r)\log q}{\log \delta_0}}}}{q^2}}}.$$

Our goal is to show that M decreases when r increases, i.e., $F(r)$ is decreasing in r. After some computation, we can get

$$\frac{F(r+1)}{F(r)} = \frac{\kappa + \ln L(r+1)}{\kappa + \ln L(r)} \left(\varepsilon^2(r)\right)^{1-X(r)^{\sqrt{\frac{n-r-1}{n-r}}-1}},$$

where $X(r) = (\delta_0^4)^{\sqrt{\frac{(n-r)\log q}{\log \delta_0}}}$. Using Assumption 3 and Lemma 3, we can upper bound the right hand side of the above equation by a function that only depends on β. Then it is easy to check that this function is less than 1 when $50 < \beta < 1990$. □

Remark 2. We emphasize that the range $50 < \beta < 1990$ is not a necessary condition and hence it should not be taken as a criterion to predict when M is decreasing in r. For $\beta \geq 1990$, we could add other restrictions such that the conclusion still holds. We decide to choose this restriction $(50 < \beta < 1990)$ since it covers most cryptographic use cases, specifically, the 5 schemes tested in Section 7.

Combining Lemma 4 and Lemma 5, we get the following conclusion.

[9] The formular in [33] is $\sqrt{\frac{n \log q}{\log \delta_0}}$ since [33] considers the original dual attack.

Theorem 1. *For* HYBRID 1 *under the core-SVP model, for any LWE instance with arbitrary secrets, assuming* $50 < \beta < 1990$ *and Assumption 3, when we increase the guessing dimension* r*, the optimal BKZ blocksize* β *that minimizes* $N \cdot T_{BKZ}$ *and maintains a same level of success rate will be reduced.*

Remark 3. Similarly to the remark for Lemma 5, we emphasize that Theorem 1 should be taken as an exemplary evidence for that, for most cryptographic use cases, the blocksize β of BKZ can be reduced when we guess some entries, and hence HYBRID 1 can outperform dual attack. However, the range $50 < \beta < 1990$ in Theorem 1 should not be taken as a criterion to predict when HYBRID 1 will outperform dual attack. Except for the reason that $50 < \beta < 1990$ is not a necessary condition for Lemma 5, a more important reason is that the condition "increasing r will decreases β" in Theorem 1 is not necessary for HYBRID 1 to outperform dual attack. It could happen that at the beginning increasing r makes β increased, but if we can afford to increase r a little bit larger (while maintaining $T_{\text{guess}} \leq T_{\text{BKZ}}$), then overall β will be decreased. For example, if we can afford to guess 6 entries of the secret (increase r from 0 to 6), then β will be reduced for any $50 < \beta < 2^{30}$. To sum up, one should not reply on Theorem 1 to decide whether HYBRID 1 will outperform dual attack (although it covers most cryptographic use cases). Instead, one should always consider HYBRID 1 when dual attack is considered, and in practice, one just need to run our estimator to see whether HYBRID 1 outperforms dual attack. In the next section, we will give a predictor that estimate the improvement of HYBRID 1.

3.4 Predicting improvement of Hybrid 1

We now proceed to a predictor that estimate the advantage of HYBRID 1 over dual attacks under the aforementioned core-SVP model. We give our theoretical results in Theorem 3. We also compare the predictor's outputs (i.e., advantage + dual attacks) with our HYBRID 1 estimator, for sanity checking the correctness of the predictor. The results are shown in Table 3.

Let us first expand the result of Theorem 1. Our simulations show that, for all 5 schemes, the value of the optimal β decreases *linearly* as r increases. However, the slopes differ among the schemes. We could have computed the slope from m, n, σ, b and q, but it's hard to derive a concrete formula from them. For simplicity, our predictor uses pre-computed slopes that we derived from our simulations. As a consequence, our predictor relies on the following heuristic.

Heuristic 2. *Fix* $N = 1$*. The optimal* β *decreases linearly as* r *increases. The slope, denoted by* α*, for 5 schemes are shown in Table 3.*

Next, our simulations show that the optimal r and β for HYBRID 1 will satisfy $T_{\text{BKZ}} \approx T_{\text{guess}}$, i.e., we should increase r till the cost of guessing is about the same as the cost of BKZ. To ease analysis, we will assume $T_{\text{BKZ}} = T_{\text{guess}}$ and take parameters r and β as real numbers in our predictor. Since $N = 1$ (Lemma 3), we have $T_{\text{HYBRID 1}} = 2T_{\text{BKZ}} = 2T_{\text{guess}}$. Note that this approximation differs

Table 3. Comparison of Hybrid 1 and the Predictor

Name	Parameters			Dual	Hybrid 1	Predictor
	b_1	R	α			
Kyber512	382	5	-0.9179	112	109	109
Kyber768	622	5	-0.9574	182	177	177
Kyber1024	870	5	-0.9764	254	246	246
Saber512	402	11	-0.956	117	115	116
Saber768	648	9	-1	189	185	185
Saber1024	885	7	-1	258	252	252
Dilithium768	343	13	-0.6296	100	99	100
Dilithium1024	485	11	-0.6301	142	140	140
Dilithium1280	598	7	-0.5993	175	172	173
Frodo640	486	25	-0.9768	142	140	140
Frodo976	705	21	-0.8828	206	203	203
Frodo1344	927	13	-0.8174	271	266	267
NTRULPrime653	447	3	-0.8306	131	126	126
NTRULPrime761	532	3	-0.8453	155	150	150
NTRULPrime857	605	3	-0.8509	177	170	170

from the optimal $T_{\text{Hybrid 1}}$ by at most one bit, since increasing r will increase T_{guess} and decreasing r will increase β, which will increase T_{BKZ}.

Finally, we are ready to present our predictor, captured via Theorem 3.

Theorem 3. *Using Heuristic 2 and assume $T_{BKZ} = T_{guess}$ for* Hybrid 1*, for an LWE instance with parameters n and R, where R is the size of the support for each entry of the secret, let b_1 be the optimal β for the dual attack, then the cost of* Hybrid 1 *is $T_{\text{Hybrid 1}} = 2^{0.292 b_2 + 1}$, where $b_2 = b_1 \frac{\log R}{\log R - 0.0845\alpha}$ is the optimal β for* Hybrid 1 *and α is the slope, and the guess dimension is $r = \frac{0.0845 b_2}{\log R}$.*

Proof. According to Lemma 3, we have $M = SV = 2^{0.2075 b_2}$ (when β is taken as a real number). Using $T_{\text{guess}} = T_{\text{BKZ}} = 2^{0.292 b_2}$, we get $L = \frac{T_{\text{guess}}}{M} = \frac{T_{\text{BKZ}}}{SV} = 2^{0.0845 b_2}$. Since $L = R^r$, we get $r = \frac{0.0845 b_2}{\log R}$. According to Heuristic 2, $b_2 - b_1 = \alpha r \Rightarrow r = \frac{b_2 - b_1}{\alpha}$. Combining $r = \frac{0.0845 b_2}{\log R}$ and $r = \frac{b_2 - b_1}{\alpha}$, we get $b_2 = b_1 \frac{\log R}{\log R - 0.0845\alpha}$. □

Note that in the proof we have $L = \frac{T_{\text{BKZ}}}{SV} = 2^{0.0845 b_2}$. This means the guessing space is determined by the difference between the running time of sieving and the number of short vectors produced by sieving. If this difference becomes larger (e.g., using other cost models), then we can guess more and the improvement of Hybrid 1 compared with dual attack will be larger.

We use the result of Theorem 3 to predict the bit-security of all 5 schemes, and present the results in in Table 3. The Predictor data is computed as the sum of dual attack and the predicted advantage. The Predictor results are very close to those from our Hybrid 1 estimator, with a difference of one bit in worst cases.

4 Hybrid attack on arbitrary secrets

Essentially, there are two methods to deal with uniform secrets:

1. Attack the LWE samples directly with the original dual attacks;
2. Convert the uniform LWE samples into normal-form LWE samples (Lemma 1), and then use embedded dual attacks.

The second option requires more samples, but is believed to be more efficient in general when the number of samples permits. Via normalizing the uniform LWE, we obtain an LWE problem with short secrets. Hence we can adopt the strategy in Section 3. There are also cases where an attacker must use the original dual attacks (perhaps due to the limitation of samples, etc.). We emphasis that this setting (uniform secret and limited samples) does not reflect any real-world cryptosystem. Nonetheless, it is interesting to show that hybrid dual attacks are still better than dual attacks with both approaches, from a theoretical point of view.

To see this, we start with the first option. We can still adopt the strategy in Section 3, and combine an original dual attack with guess to obtain a hybrid original dual attack. In addition, we can still invoke the predictor from Theorem 3, via setting $R = q$, and α to a value close to -1 for simplicity (Table 3 shows that α is close to -1 and the scope is $(-0.6, -1)$. The advantage would be larger if we have larger absolute value of α.). According to Theorem 3, we have

$$r = \frac{0.0845 b_2}{\log q} \text{ and } b_2 - b_1 = \frac{b_1 \cdot 0.0845\alpha}{\log q - 0.0845\alpha} \approx -\frac{0.0845 b_1}{\log q}.$$

For a larger q the cost of guessing even a single entry becomes too high. Therefore, we can guess very few entries and the improvement is limited. Taking Regev's original scheme [34] as an example, where $q \approx n^2$ and $\sigma = \frac{q}{2\pi\sqrt{n}\log^2 n}$, we consider two different restrictions on the number of samples: the original one $m \in (0, n \log q)$ and $m \in (0, 2n)$. We see marginal improvements between 1 to 3 bits in Table 4.

For the second option, we transform the samples with $\mathbf{s}$ uniform in $\mathbb{Z}_q^n$ to normal-form ones at a loss of n samples. The advantage of this method is that as the secret is small, we can guess more entries than the previous option. Similarly, we present the estimations in Table 5. We see improvements across all parameter sets. Notice an anomaly from Regev1024: it occurs when there isn't sufficient number of samples. The advantage of hybrid embedded dual attack over embedded dual attack is surprisingly large when number of samples is (extremely) limited.

Table 4 and 5 show that hybrid dual attack always outperforms dual attack for uniform secrets, regardless the number of samples. In addition, the advantage of hybrid dual attacks increases (sometimes drastically) with the increase of n, when the number of samples is limited ($m \in (0, 2n)$).

Table 4. (Hybrid) original dual attack

Regev	$m \in (0, 2n)$		$m \in (0, n \log q)$	
n	Dual	Hybrid	Dual	Hybrid
256	63	62	56	56
512	150	149	135	134
1024	343	340	306	305

Table 5. (Hybrid) embedded dual attack

Regev	$m \in (0, n)$		$m \in (0, n \log q - n)$	
n	Dual	Hybrid	Dual	Hybrid
256	63	61	56	55
512	151	147	134	133
1024	631	570	306	303

5 Hybrid dual attack with optimal pruning

5.1 Guess with pruning

In this section, we show how to choose the *optimal* subset of secret candidates for different secret distributions when the hybrid dual attack becomes too expansive or unfeasible to guess all candidates. In this scenario, since our guess time need to approximate the cost of BKZ (similarly to Hybrid 1), we can only guess a limited number of candidates. To optimize the success probability p_c, we need to find a collection of certain number of candidates such that its success probability is as large as possible, i.e. we want to maximize the success probability when the number of candidates is limited. This can be formally stated as $\max_{|C|<c} p(C)$, where C is a collection of guessed candidates, c is the upper limit of $|C|$, and $p(C) = \Pr[\mathbf{s}_1 \in C]$ is the probability that the correct $\mathbf{s}_1$ is in C.

Note that the optimal parameters that minimize the target $\big(N \cdot T_{\text{BKZ}} + T_{\text{guess}}\big)/p_c$ may result in $p_c < \frac{1}{2}$. To boost the success probability p_c, we can repeat the attack by guessing different parts (r dimensions) of the secret. We can repeat the attack for at least $\lfloor \frac{n}{r} \rfloor$ times. Since the optimal guess strategy may ignore some candidates with low probability, it could happen that for some instances the attack fails for all $\lfloor \frac{n}{r} \rfloor$ times. However, the probability for this to happen is very low as long as p_c is not too small. For all LWE-related proposals we test in Section 7, we have $p_c \geq 2^{-0.97}$, $\lfloor \frac{n}{r} \rfloor \geq 7$, and the probability that the attack fails after repeat is at most 2^{-19} under the optimal parameters. Therefore, the attack is valid from a practical point of view.

In the rest of the section, we will look into three different distributions.

Pruning for $\boldsymbol{\mathcal{B}_h^+}$. Let $\mathbf{s} \in \mathcal{B}_h^+$ be a binary secret vector with hamming weight h. Denote S the set of all the candidates of $\mathbf{s}_1 \in \{0,1\}^r$. Let k_{min} and k_{max} be the lower and upper bound of the hamming weight of candidates in S. It is easy to see that $k_{min} = \max\big\{0, h + r - n\big\}$ and $k_{max} = \min\big\{h, r\big\}$.

Our goal is to greedily form the set C with candidates of high(est) success rate from S. To this end, we first partition the set S into several subsets according to the hamming weight. For each integer $k \in [k_{min}, k_{max}]$, let S_k be the set of candidates from S with hamming weight k. Then $S = \bigcup_{k\in[k_{min},k_{max}]} S_k$. Next, we can compute the order of S_k, denoted by $N(k)$, and the probability that S_k contains the correct $\mathbf{s}_1$, denoted by $p(k)$ for each $k \in [k_{min}, k_{max}]$ as follows:

$$N(k) = \binom{r}{k} \text{ and } p(k) = \frac{\binom{r}{k}\binom{n-r}{h-k}}{\binom{n}{h}}.$$

Since candidates in the same set S_k have the same probability to be the correct $\mathbf{s}_1$, the probability for each candidate in S_k to be $\mathbf{s}_1$ is $\overline{p}(k) = \frac{p(k)}{N(k)} = \frac{\binom{n-r}{h-k}}{\binom{n}{h}}$. Finally, based on $\overline{p}(k)$, we can greedily choose candidates in S_k with the highest $\overline{p}(k)$ to C till $|C| \approx c$. It is easy to see that this method achieve the optimal success probability as every time when we put a vector into C, it is the one with the highest success probability $\overline{p}(k)$ in $S\backslash C$.

Note 2. If $n > r + 2h$, then it holds that $(n-r)/2 > h - k$, and hence $\overline{p}(k)$ decreases as k increases. Therefore, in this case, we should always start guessing candidates from S_k with the lowest hamming weight. Accordingly, the guessing time and success probability are

$$T_{\text{guess}} = M \cdot \sum_{i=0}^{h^*} N(i) \cdot i, \quad \text{and} \quad p_c = \sum_{i=1}^{h^*} p(i),$$

where h^* satisfies $\sum_{i=1}^{h^*} N(i) < c$ and $\sum_{i=1}^{h^*+1} N(i) > c$.

Pruning for $\boldsymbol{\mathcal{B}_h^-}$. Let $\mathbf{s} \in \mathcal{B}_h^-$ be a ternary secret vector with h number of 1 and h number of -1. Similar to the case of binary secret vector, let $S_{(k^+,k^-)}$ be a subset of S where k^+ and k^- denote the number of 1 and -1, respectively. The order of $S_{(k^+,k^-)}$ (denoted by $N(k^+,k^-)$) and the probability that $S_{(k^+,k^-)}$ contains the correct $\mathbf{s}_1$ (denoted by $p(k^+,k^-)$) are calculated as

$$N(k^+,k^-) = \binom{r}{k^+}\binom{r-k^+}{k^-}, \quad p(k^+,k^-) = \frac{\binom{r}{k^+}\binom{r-k^+}{k^-}\binom{n-r}{h-k^+}\binom{n-r-h+k^+}{h-k^-}}{\binom{n}{h}\binom{n-h}{h}}.$$

Also, the probability for each candidate in $S_{(k^+,k^-)}$ to be the correct $\mathbf{s}_1$ is

$$\overline{p}(k^+,k^-) = \frac{p(k^+,k^-)}{N(k^+,k^-)} = \frac{\binom{n-r}{h-k^+}\binom{n-r-h+k^+}{h-k^-}}{\binom{n}{h}\binom{n-h}{h}}.$$

Based on $\overline{p}(k^+,k^-)$, we choose the candidates in $S_{(k^+,k^-)}$ with the highest $\overline{p}(k^+,k^-)$ to C till $C \approx c$. Accordingly, the guessing time and success probability are

$$T_{\text{guess}} = M \cdot \sum_{S_{(i^+,i^-)}\in C} N(i^+,i^-) \cdot (i^+ + i^-) \text{ and } p_c = \sum_{S_{(i^+,i^-)}\in C} p(i^+,i^-).$$

Note 3. If $n > r + 3h$, then $\overline{p}(k^+, k^-)$ decreases when $k^+ + k^-$ increases. Moreover, for a fixed $k^+ + k^-$, $\overline{p}(k^+, k^-)$ decreases as $|k^+ - k^-|$ increases. Therefore, in this case, we should choose the candidates following two rules: $k^+ + k^-$ is minimized, and $|k^+ - k^-|$ is minimized.

Pruning for central discrete distribution. For a general central discrete distribution with a support $S := \{0, \pm 1, \cdots, \pm t\}$, we partition all candidates in S into subsets according to the appearance of each value in S. Denote $S_{(k_0,k_1,\cdots,k_t)}$ the subset of candidates with k_i entries being $\pm i$ for $i \in [0, t]$. For each subset, its order, denoted by $N(k_0, k_1, \cdots, k_t)$, and the probability of *each* candidate to be the correct guess, denoted by $\overline{p}(k_0, k_1, \cdots, k_t)$, can be calculated as

$$N(k_0, k_1, \cdots, k_t) = \binom{r}{k_0}\binom{r-k_0}{k_1}\cdots\binom{r-k_0-\cdots-k_{t-1}}{k_t}\cdot 2^{r-k_0},$$

$$\overline{p}(k_0, k_1, \cdots, k_t) = p_0^{k_0} p_1^{k_1} \cdots k_t^{k_t}.$$

Based on $\overline{p}(k_0, k_1, \cdots, k_t)$, we choose the candidates in $S_{(k_0,k_1,\cdots,k_t)}$ with the highest $\overline{p}(k_0, k_1, \cdots, k_t)$ to C till $C \approx c$. Accordingly, the guessing time and success probability are

$$T_{\text{guess}} = M \cdot \sum_{S_{(i_0,\cdots,i_t)} \in C} N(i_0, \cdots, i_t) \cdot (i_1 + \cdots + i_t), p_c = \sum_{S_{(i_0,\cdots,i_t)} \in C} p(i_0, \cdots, i_t).$$

5.2 The advantage of optimal guess

Now we are ready to analyze the advantage of Hybrid 2 over Hybrid 1. Similar to the previous comparison in Section 3.3, it is safe to ignore p_s as it is close to 1 for both algorithms. Recall that we have

$$T_{\text{Hybrid 1}} = N \cdot T_{\text{BKZ-h1}} + T_{\text{guess-h1}}$$
$$T_{\text{Hybrid 2}} = \big(N \cdot T_{\text{BKZ-h2}} + T_{\text{guess-h2}}\big)/p_c$$

Intuitively, in Hybrid 2, our guess dimension r will be larger. This decreases blocksize β, and therefore, the cost for a single attack is reduced. So long as the advantage one gains via Hybrid 2 makes it up to the loss in success probability (p_c), pruning will improve the overall cost. The detailed analysis comes as follows.

We first analyze the relation between the cost $T_{\text{Hybrid 2}}$ and the parameters r, β and L, which is shown in Figure 3. Note that the influence of r and β on the cost $T_{\text{Hybrid 2}}$ is almost the same as in Hybrid 1. The only difference is that in Hybrid 1 the number of candidates L is directly determined by r since we guess all candidates, while in Hybrid 2, L is a free parameter that the attacker can choose. This introduces a success probability p_c, i.e., the optimal probability we can achieve via optimal pruning in Section 5.1. It's easy to see that increasing r or decreasing L will decrease p_c.

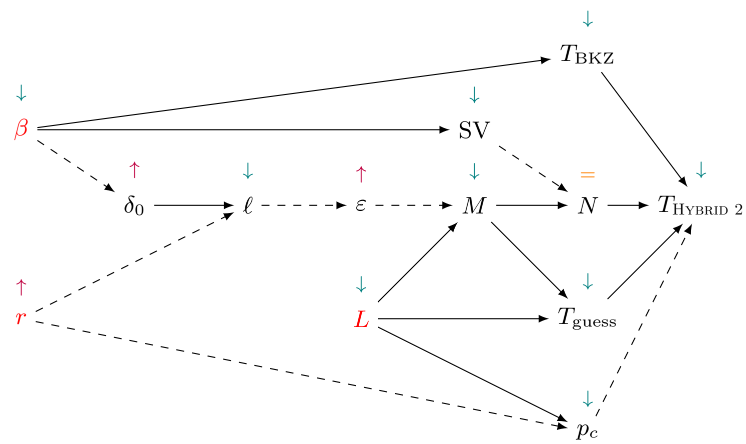


Fig. 3. Parameter relations in Hybrid 2 and value changes from Hybrid 1 to Hybrid 2. The attack can choose the value for parameters in red color (i.e., β, r and L), which then determine the value for other parameters.

A natural next step is to adjust the parameters r, β, L in Hybrid 2 to get a lower cost $T_{\text{Hybrid 2}}$ than that of Hybrid 1. Recall that in Hybrid 1, we fix $N = 1$, and gradually increase r from 0 (and decrease β accordingly) till $T_{\text{BKZ}} = T_{\text{guess}}$. We follow a similar strategy in Hybrid 2 by fixing $N = 1$ and gradually increase r. Once a balance between T_{BKZ} and T_{guess} is reached, we gradually decrease L (this do not change the condition that $T_{\text{BKZ}} = T_{\text{guess}}$) and compute the corresponding success probability p_c. We search for the point where the overall cost is minimal. Note that a deciding factor on whether there exists a minimal point (other than the starting point of L), in other words, whether Hybrid 2 can outperform Hybrid 1, is the concentration of the secret distribution.

Concentration level. As we will see in Section 7 the improvement of Hybrid 2 depends largely on the individual secret distribution. For example, for secret distributions that are more centralized, the success probability p_c are higher. To capture this quantity, we formally define a *concentration level* as a metric to indicate the effectiveness of our optimal pruning.

Definition 2. *Let $g(r, L)$ be a function of r and L, which is the optimal success probability when* Hybrid 2 *guesses L candidates for a secret of dimension r and distribution χ, i.e., $g(r, L) = \max_{C \subseteq D(r), |C| \leq L} p(C)$, where $D(r)$ is the set of all candidates for the secret and $p(C)$ is the probability that the correct secret is in C. We say $g(r, L)$ is χ's concentration level.*

As per definition, $g(r, L)$ characterizes how *centralized* a distribution is, or how hard it is to achieve a high success probability when guessing r dimensions and L candidates. For example, for two distributions χ_A and χ_B, if we guess a same r and L and we get $g_{\chi_A}(r, L) > g_{\chi_B}(r, L)$, then we can claim that χ_A is more centralized, or easier to guess. The metric $g(r, L)$ will be used in Theorem 4.

Note that concentration level is different from entropy. Surprisingly, a distribution with higher entropy could have a higher concentration level, which means easier to guess. For example, for two distributions χ_A and χ_B with the same support set $\{0, 1, 2\}$ and $p_A = (0.6, 0.2, 0.2)$ and $p_B = (0.5, 0.5, 0)$, the entropy of χ_A is higher than that of χ_B, but when guessing only one dimension ($r = 1$) and one candidate ($L = 1$), the success probability for χ_A is higher than that of χ_B, i.e., $g_{\chi_A}(1, 1) = 0.6 > g_{\chi_B}(1, 1) = 0.5$.

Example. To show how the concentration level influences Hybrid 2, let us consider two typical examples:

- LAC192 with a secret distribution $\mathcal{B}_h^+$ for $n = 1024$ and $h = 128$;
- Dilithium768 whose secret is from uniform distribution.

Our simulations show that Hybrid 2 can reduce the bit complexity of LAC192 by 12 bits compared with Hybrid 1, but there is no difference between Hybrid 2 and Hybrid 1 for Dilithium768.

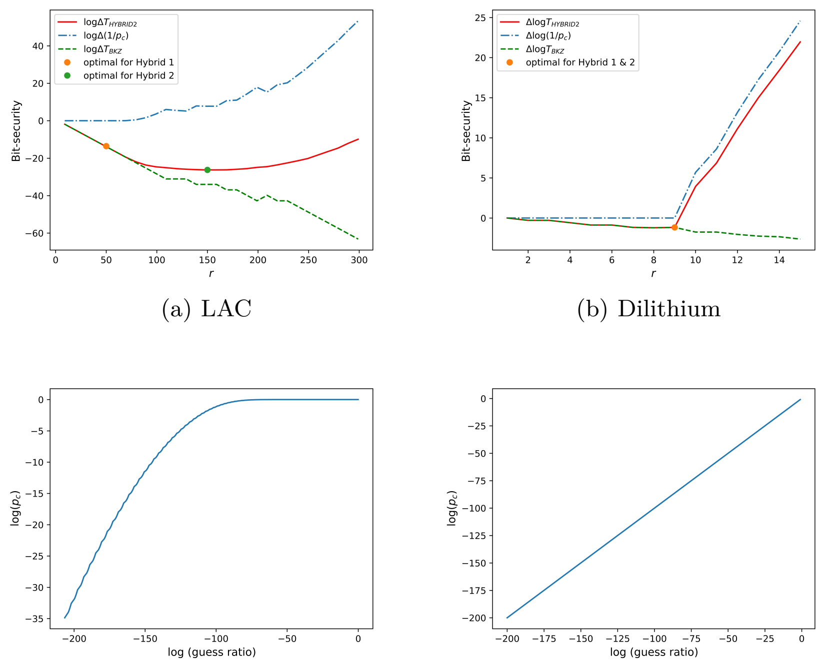
For each r, we should choose an appropriate β such that $N = 1$ and then choose L such that $T_{\text{guess}} = T_{\text{BKZ}}$. Then, for a secret distribution, the bit complexity and the optimal success probability $p_c = g(r, L)$ can be expressed as functions of r. We plot this function in the Figure 4. Specifically, 4(a) and 4(b) show the progression of $T_{\text{Hybrid 2}}$, T_{BKZ}, and p_c as functions of r, and 4(c) and 4(d) show the centralization function $g(r, L)$ for the two different secret distributions. For better visualization, in 4(a) and 4(b), we present the following quantities:

- $\Delta \log T_{\text{Hybrid 2}}(r) = \log T_{\text{Hybrid 2}}(r) - \log T_{\text{Hybrid 2}}(0)$,
- $\Delta \log T_{\text{BKZ}}(r) = \log T_{\text{BKZ}}(r) - \log T_{\text{BKZ}}(0)$,
- $\Delta \log(1/p_c(r)) = \log(1/p_c(r)) - \log(1/p_c(0))$.

For LAC192, when $0 \le r \le 50$, $T_{\text{BKZ}}(r)$ decreases; $1/p_c(r) = 1/p_c(0) = 1$. As a result, $T_{\text{Hybrid 2}}(r)$ and $T_{\text{BKZ}}(r)$ behaves similarly. Indeed, during this stage, we have $T_{\text{guess}}(r) < T_{\text{BKZ}}(r)$. This means we have been under-guessing for Hybrid 2: we can afford to guess all candidates. The optimal r for Hybrid 1 is $r = 50$ when $T_{\text{guess}}(r) = T_{\text{BKZ}}(r)$.

On the other hand, when $50 < r \le 150$, $T_{\text{BKZ}}(r)$ decreases and $1/p_c(r)$ increases. The overall cost, $T_{\text{Hybrid 2}}(r)$ drops since the gain in doing less BKZ overtakes the loss of success probability. The above gain and loss balance out at $r = 150$, at which point, Hybrid 2 becomes optimal.

For Dilithium768, $0 \le r \le 9$ is also the under-guessing phase where Hybrid 1 $\approx$ Hybrid 2. Beyond $r = 9$, $1/p_c(r)$ increases much faster due to its low concentration level, there is not a point where the gain in BKZ cost can catch up the loss in success probability. Therefore, for Dilithium768, pruning does not improve the hybrid attack.



(a) LAC (b) Dilithium

(c) LAC (d) Dilithium

Fig. 4. Comparison between LAC192 and Dilithium768. Figure (a) and (b) plot $T_{\text{HYBRID 2}}$, T_{BKZ}, and p_c in function of r; Figure (c) and (d) visualize the impact of centralization level over p_c.

Figure 4(c) and 4(d) visualize the concentration level for a fixed $r = 150$. Here, observe that for LAC192 a small ratio of guessed candidates is enough to achieve a high success probability, while for Dilithium768 with uniform secrets, the success probability is proportional to the guessed candidates. For example, with a guess ratio of 2^{-50}, the success probability is close to 1 for LAC192, and remains 2^{-50} for Dilithium768.

5.3 Predicting improvement of Hybrid 2

In this section, we present a predictor for Hybrid 2's advantage. In our simulator, we observe that, similar to Hybrid 1, the optimal parameters for Hybrid 2 also satisfy that $N = 1$ and $T_{\text{BKZ}} = T_{\text{guess}}$. This leads to the predictor in Theorem 4. We defer the proof to supplementary material C.3.

Theorem 4. *Assuming Heuristic 2 and that the optimal parameters of* Hybrid 2 *satisfy* $N = 1$ *and* $T_{BKZ} = T_{guess}$*, let* b_1 *the optimal* β *for the dual attack, then the optimal cost of* Hybrid 2 *when guessing* r *entries of the secret* $\mathbf{s}$ *is*

$f(r) = \frac{2^{0.292 \cdot b(r)+1}}{g(r, 2^{0.0845 \cdot b(r)})}$, *where* $b(r) = b_1 + \alpha r$ *is the optimal* β *corresponding to* r, $g(r, L)$ *is the centralization function, and* α *is the slope. The optimal cost of* Hybrid 2 *is* $\min_{r \geq 0} f(r)$.

Remark 4. As an additional sanity check, we show that Theorem 3 and 4 converge when guessing all candidates is indeed the optimal strategy. In this case we have $g(r^*, 2^{0.0845 \cdot b(r^*)}) = 1$ for some optimal point r^*. Note that $2^{0.0845 \cdot b(r^*)} = R^{r^*}$, where R is the size of the support for each entry of the secret. Combined with $b(r^*) = b_1 + \alpha r^*$, we achieve Theorem 3, that is, $b_2 \approx b_1 \frac{\log R}{\log R - 0.0845\alpha}$.

6 An additional optimization

Recall that in the guessing stage, for each $\tilde{\mathbf{s}}_1 \in C$, we use M short vectors $(\mathbf{w}, \mathbf{v}) \in \Lambda^E_{\text{dual}}(\mathbf{A}_2)$ to check the distribution of $\tilde{e} = \hat{b} - \langle \hat{\mathbf{a}}, \tilde{\mathbf{s}}_1 \rangle \bmod q$ corresponding to the guesses $\tilde{\mathbf{s}}_1$ (line 9 in Algorithm 1). For all the M short vectors and all the L guessed $\tilde{\mathbf{s}}_1$, we rewrite their combinations into the matrix form as $\tilde{\mathbf{E}} = \hat{\mathbf{B}} - \hat{\mathbf{A}}\mathbf{S} \bmod q$, where $\tilde{\mathbf{E}}, \hat{\mathbf{B}} \in \mathbb{Z}_q^{M \times L}$, $\hat{\mathbf{A}} \in \mathbb{Z}_q^{M \times r}$ and $\mathbf{S} \in \mathbb{Z}^{r \times L}$. Each column of $\tilde{\mathbf{E}}$ denotes all the $\tilde{e}$'s to be tested of a guessed $\tilde{\mathbf{s}}_1 \in C$. Therefore, the overall cost of the guessing stage has two main parts: (1), computing the multiplication of $\hat{\mathbf{A}}$ and $\mathbf{S}$ and (2), checking the distributions of all the L columns of $\tilde{\mathbf{E}}$. It is obvious that the multiplication cost dominants, and is therefore, the focus of optimization.

6.1 An efficient algorithm from [25]

A school book multiplication for $\mathbf{A} \in \mathbb{Z}_q^{M \times r}$ and $\mathbf{S} \in \mathbb{Z}^{r \times L}$ takes $O(M \cdot r \cdot L)$, assuming integer multiplications take unit time. [25] improves the cost by a factor of r, when the matrix $\mathbf{S}$ has a special form.

Lemma 6 ([25]). *The product of a matrix* $\mathbf{A} \in \mathbb{Z}^{M \times r}$ *and a matrix* $\mathbf{S}$ *of size* $r \times \ell^r$ *which consists of all vectors from* $\{t_1, \dots, t_\ell\}^r$ *in lexicographic order can be calculated in* $\mathcal{O}(M \cdot \ell^r)$ *time.*

The idea of the algorithm from [25] is as follows. For any $i \in \mathbb{N}$, denote $\mathbf{S}_{(i)}$ of size $i \times \ell^i$ the matrix consisting of all vectors from $\{t_1, \cdots, t_\ell\}^i$ in lexicographic order. These matrices can be constructed recursively. For $i = 1, \mathbf{S}_{(1)} = \begin{pmatrix} t_1 \cdots t_\ell \end{pmatrix}$ and for $\forall i > 1$, $\mathbf{S}_{(i)} = \begin{pmatrix} \mathbf{S}_1 & \cdots & \mathbf{S}_\ell \\ \mathbf{S}_{(i-1)} & \cdots & \mathbf{S}_{(i-1)} \end{pmatrix}$. Denote $\mathbf{a} = (a_r, \cdots, a_1)$ a d-dimensional vector and $\mathbf{a}_{(i)} = (a_i, \cdots, a_1)$. Then the scalar products of $\mathbf{a}_{(i)}$ and $\mathbf{S}_{(i)}$ can be calculated recursively as follows:

$$\begin{aligned} \mathbf{a}_{(i)} \cdot \mathbf{S}_{(i)} &= \begin{pmatrix} a_i \ \mathbf{a}_{(i-1)} \end{pmatrix} \begin{pmatrix} \mathbf{t}_1 & \cdots & \mathbf{t}_\ell \\ \mathbf{S}_{(i-1)} & \cdots & \mathbf{S}_{(i-1)} \end{pmatrix} \\ &= \begin{pmatrix} a_i \cdot \mathbf{t}_1 + \mathbf{a}_{(i-1)}\mathbf{S}_{(i-1)} \cdots a_i \cdot \mathbf{t}_\ell + \mathbf{a}_{(i-1)}\mathbf{S}_{(i-1)} \end{pmatrix}. \end{aligned}$$

Hence, given $\mathbf{a}_{(i-1)}\mathbf{S}_{(i-1)}$, we can compute $\mathbf{a}_{(i)} \cdot \mathbf{S}(i)$ in $\mathcal{O}(\ell^i)$ time. Therefore, the product of $\mathbf{A}$ and $\mathbf{S}$ can be calculated in time $\mathcal{O}(M \cdot \ell^i)$. We state this result formally in the following lemma.

However, Lemma 6 only works for HYBRID 1, and does not work after pruning. For example, for a central discrete distribution with a support set $\{0, \pm 1, \pm 2\}$ and $p_0 = 0.7, p_1 = 0.2, p_2 = 0.1$, an optimal guess set C for dimension 3 may contain $(0,0,1)$ and $(0,0,2)$, but not $(1,1,1)$, since $(0,0,1)$ and $(0,0,2)$ have higher success probabilities than $(1,1,1)$. Now there is no set in the form required by Lemma 6 (except for the whole set $\{0, \pm 1, \pm 2\}^3$) that contains $(0,0,1)$ and $(0,0,2)$ but not $(1,1,1)$. In the next section, we present an improved algorithm.

6.2 An improved algorithm

Warm up. Let us begin with our intuition. Let $\mathbf{a} = (a_1, a_2, \ldots, a_r)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_r)$ be two vectors of dimension r. Compute $\langle \mathbf{a}, \mathbf{b} \rangle$ requires $O(r)$ time. However, if we already have the result of $\langle \mathbf{a}, \mathbf{b}' \rangle$, where $\mathbf{b}'_j = 0$ for some $j \in [r]$ and $\mathbf{b}'_i = \mathbf{b}_i$ for all other $i \neq j$, then $\langle \mathbf{a}, \mathbf{b} \rangle = \langle \mathbf{a}, \mathbf{b}' \rangle + \mathbf{a}_j \mathbf{b}_j$ can be computed in constant time based on the result of $\langle \mathbf{a}, \mathbf{b}' \rangle$. To compute the product of a vector $\mathbf{a}$ and a matrix $\mathbf{S}$, we need to compute the inner product of $\mathbf{a}$ with each column of $\mathbf{S}$. If all columns of the matrix $\mathbf{S}$ have an order such that the inner product for one column can be computed recursively based on the inner product for another column, then we can drop the dimension r out in the running time.

Concrete algorithm. We start with a few new definitions. Let $D \subseteq \mathbb{Z}$ be a set of integers including 0. For two vectors $\mathbf{v}, \mathbf{v}' \in D^r$, we say $\mathbf{v}'$ *precedes* $\mathbf{v}$, denoted as $\mathbf{v}' \prec \mathbf{v}$, if there exists $j \in [r]$ such that $\mathbf{v}'_j = 0$ and $\mathbf{v}'_i = \mathbf{v}_i$ for all $i \neq j$. Slightly abusing the notation, we use $\mathbf{S}$ as the set of column vectors of $\mathbf{S}$ and we write $\mathbf{v} \in \mathbf{S}$ if $\mathbf{v}$ is a column of $\mathbf{S}$. Finally we can formally define the *closed* matrices.

Definition 3 (Closed Matrix). *For a matrix $\mathbf{S} \in D^{r \times L}$, we say $\mathbf{S}$ is* closed *if for any $\mathbf{v} \in \mathbf{S}$, we have $\mathbf{v}' \in \mathbf{S}$ for all $\mathbf{v}' \prec \mathbf{v}$.*

The main result of this section is stated in the following theorem.

Theorem 5. *The product of a matrix $\mathbf{A} \in \mathbb{Z}^{M \times r}$ and a closed matrix $\mathbf{S} \in D^{r \times L}$, where $D \subseteq \mathbb{Z}$ is a set of integers including 0, can be computed in $\mathcal{O}(M \cdot L)$ time.*

We defer to supplementary material C.4 for the proof.

Next, we show that all the optimal subsets of candidates discussed in Section 5.1 are closed, and hence Theorem 5 can be applied. The proof of Corollary 1 is deferred to supplementary material C.5.

Corollary 1. *If the guessing part $\mathbf{s}_1$ has dimension r and the secret distribution of the LWE problem is from one the following distributions: $\mathcal{B}_h^+$ with $n - r \geq 2h$, $\mathcal{B}_h^-$ with $n - r \geq 3h$, or a central discrete distribution, then the candidate* *subset C^* for $\mathbf{s}_1$ satisfying that $C^* = \arg\max_{|C|<c} p(C)$ is closed. Hence, the multiplication of the matrix $\hat{\mathbf{A}} \in \mathbb{Z}_q^{M\times r}$ and the corresponding optimal candidate matrix $\mathbf{S}^* \in \mathbb{Z}^{r\times L}$ can be computed in $\mathcal{O}(M \cdot L)$ time.*

7 Security estimations

We conclude our paper with new estimations for 5 NIST-PQC candidates. Their parameters are given in Table 7. The highlight are presented in Table 2, and a full comparison is given in Table 11. Again, our base line for comparison is the dual attack. Then we compare it with the most optimized one, Hybrid 2m, taking into account the optimal pruning and our additional optimization. Our results are in both the core-SVP model and the practical model. We skipped the Frodo model, since its estimations will always lie in between those of core-SVP and practical models.

The number of samples allowed from each scheme is shown in Table 7. We observe that the optimal number of samples is smaller than the allowed one in our simulation, with an exception of Frodo. For Frodo, we use the optimal number of samples under the restriction of allowed samples. Nevertheless, the influence of this restriction is at most one bit. We set target $p_s = 1 - \frac{1}{2^\kappa}$ with $\kappa = 128$.

In addition, we note that for the schemes whose distributions of secret $\mathbf{s}$ and error $\mathbf{e}$ are different, we use the "modulus switching" technique [1] (which balances the weight of $\mathbf{s}$ and $\mathbf{e}$) to improve the estimation results. Among the 5 schemes we considered, we use this technique for Saber and NTRULPrime.

For all cases, Hybrid 2m is more efficient than dual attacks, regardless of the model. Although, we remark that the gain becomes more significant, if we assume a higher complexity of BKZ (i.e., the practical model). Our method reports an overall improvement between 1 to 9 bits; the actual improvement varies, depending on scheme/parameter sets, as well as the security model. Our algorithm works best on NTRULPrime857 and Kyber1024 under the core-SVP model, which records an improvement of 9 bits with classical computation.

We want to emphasis that, the new estimations for Kyber, Dilithium and NTRULPrime are indeed lower than the corresponding security level. As a final takeaway, we believe that hybrid dual attacks (with pruning) should be considered for cryptanalysis on any future practical lattice-based cryptosystem.

References

1. Martin R Albrecht. On dual lattice attacks against small-secret lwe and parameter choices in helib and seal. In *Eurocrypt*, pages 103–129. Springer, 2017.
2. Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Algebraic algorithms for LWE problems. *ACM Commun. Comput. Algebra*, 49(2):62, 2015.
3. Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the {LWE, NTRU} schemes! In *SCN 2018*.

4. Martin R. Albrecht, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Lazy modulus switching for the BKW algorithm on LWE. In *PKC 2014*, Lecture Notes in Computer Science, 2014.
5. Martin R Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving usvp and applications to lwe. In *Asiacrypt*, pages 297–322. Springer, 2017.
6. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *J. Mathematical Cryptology*, 9(3):169–203, 2015.
7. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange—a new hope. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 327–343, 2016.
8. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO 2009*, 2009.
9. Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP 2011*, 2011.
10. Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU prime: Reducing attack surface at low cost. In *SAC 2017*.
11. Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A non-pcp approach to succinct quantum-safe zero-knowledge. In *CRYPTO 2020*.
12. Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *CRYPTO 2019*.
13. Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In *ACM CCS, 2016*.
14. Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In *EuroS&P 2018*.
15. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS 2012*.
16. Johannes A. Buchmann, Florian Göpfert, Rachel Player, and Thomas Wunderer. On the hardness of LWE with binary error: Revisiting the hybrid lattice-reduction and meet-in-the-middle attack. In *AFRICACRYPT 2016*, 2016.
17. Yuanmi Chen. *Réduction de réseau et sécurité concrete du chiffrement completement homomorphe*. PhD thesis, Paris 7, 2013.
18. Yuanmi Chen and Phong Q Nguyen. Bkz 2.0: Better lattice security estimates. In *Asiacrypt 2011*, pages 1–20. Springer, 2011.
19. Jung Hee Cheon, Minki Hhan, Seungwan Hong, and Yongha Son. A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret LWE. *IEEE Access*, 7:89497–89506, 2019.
20. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: fast fully homomorphic encryption over the torus. *J. Cryptology*, 33(1):34–91, 2020.
21. Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In *CRYPTO 2020*.
22. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure KEM. In *AFRICACRYPT 2018*.
23. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *TCHES*, 2018(1):238–268, 2018.

24. Muhammed F. Esgin, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In *CRYPTO 2019*.
25. Thomas Espitau, Antoine Joux, and Natalia 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 *INDOCRYPT 2020*.
26. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC 2009*.
27. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO 2013*.
28. Jeffrey Hoffstein, Jill Pipher, John M. Schanck, Joseph H. Silverman, William Whyte, and Zhenfei Zhang. Choosing parameters for ntruencrypt. In *CT-RSA 2017*.
29. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS-III, 1998*.
30. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 150–169. Springer, 2007.
31. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against ntru. In *Crypto*. Springer, 2007.
32. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for lwe-based encryption. In *CT-RSA 2011*.
33. Daniele Micciancio and Oded Regev. Lattice-based cryptography in post quantum cryptography, bernstein dj, buchmann j., dahmen e, 2009.
34. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.
35. Yongha Son and Jung Hee Cheon. Revisiting the hybrid attack on sparse and ternary secret LWE. *IACR Cryptol. ePrint Arch.*, 2019:1019, 2019.
36. Damien Stehlé. An overview of lattice reduction algorithms. Invited talk at ICISC, 2013.
37. Thomas Wunderer. *On the Security of Lattice-Based Cryptography Against Lattice Reduction and Hybrid Attacks*. PhD thesis, Darmstadt University of Technology, Germany, 2018.
38. Thomas Wunderer. A detailed analysis of the hybrid lattice-reduction and meet-in-the-middle attack. *J. Mathematical Cryptology*, 13(1):1–26, 2019.

A Parameters for various cryptosystems

Parameters for various cryptosystems considered in this paper are listed in Table 6, 7, 8, 9 and 10.

Table 6. Parameters of LAC192

Name	n	q	σ	Secret distribution	Hamming weight
LAC192	1024	251	$1/\sqrt{2}$	$\#(-1,0,1) = (128,768,128)$	256

B A concrete treatment of embedded dual attacks

The analysis of Alkim et al. [7] about the cost and success probability of embedded dual attack, as shown in Section 2.4, does not specify how to distinguish the Gaussian distribution from the uniform to get the desired advantage (which equals to the statistical distance between the two distributions) and how to amplify this advantage. In this section, we present concrete algorithms to make the conclusion given by Alkim et al. [7] complete.

B.1 With a single short vector

Given instances $(\mathbf{A}, \mathbf{b})$ from $\mathrm{LWE}_{\mathbf{s},\sigma}$ or uniform distribution $\mathcal{U}(\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q)$ and a short vector $(\mathbf{w}, \mathbf{v}) \in \Lambda_{\mathrm{dual}}^E$ of length ℓ, denote $t = \langle \mathbf{w}, \mathbf{b}\rangle$, which is from $\mathcal{G}_\rho$ or $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$ accordingly. Let ε be the maximum variance distance between $\mathcal{G}_\rho$ and $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$. In the following, we present an algorithm to distinguish the distribution of t with an advantage ε.

Denote $g(x)$ and $G(x)$ the probability density function and the cumulative distribution function of $\mathcal{G}_\rho$, respectively. Similarly, denote $f(x)$ and $F(x)$ those of $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$. Let I_g be the subset of $(-\frac{q}{2}, \frac{q}{2})$ where $g(x) \geq f(x)$, and accordingly, let I_f be the subset of $(-\frac{q}{2}, \frac{q}{2})$ where $f(x) > g(x)$. Denote

$$p_g \coloneqq \Pr\left[t \in I_g | t \leftarrow \mathcal{G}_\rho\right], \text{ and}$$

$$p_f \coloneqq \Pr\left[t \in I_g | t \leftarrow \mathcal{U}(-\frac{q}{2}, \frac{q}{2})\right].$$

By definition, we can compute the maximal variance distance between $\mathcal{G}_\rho$ and $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$ by p_g and p_f as follows.

Lemma 7. $\varepsilon = p_g - p_f$.

Table 7. Parameters for NIST-PQC round 3 LWE-based schemes

Name	Parameters					Security			
	n	k	q	σ	m^*	Secret dist.	Level	Claim	
								Classical	Quantum
Kyber	256	2	3329	1	768	see Table 8	1	111	100
	256	3	3329	1	1024		3	181	164
	256	4	3329	1	1280		5	254	230
Saber	256	2	2^{13}	2.29	768	see Table 9	1	118	107
	256	3	2^{13}	2.29	1024		3	189	172
	256	4	2^{13}	2.29	1280		5	260	236
Dilithium	256	3	8380417	3.74	1024	uniform in $[-6, 6]$	1	100	91
	256	4	8380417	3.16	1280	uniform in $[-5, 5]$	2	141	128
	256	5	8380417	2	1536	uniform in $[-3, 3]$	3	174	158
Frodo	640	-	2^{15}	2.8	640	see Table 10	1	148	108
	976	-	2^{16}	2.3	976		3	214	154
	1344	-	2^{16}	1.4	1344		5	279	201
NTRULPrime	653	-	4621	$\sqrt{2/3}$	909	$\#(\pm 1) = 252$	1	130	118
	761	-	4591	$\sqrt{2/3}$	1017	$\#(\pm 1) = 250$	2	155	140
	857	-	5167	$\sqrt{2/3}$	1113	$\#(\pm 1) = 281$	3	176	160

* The parameters are the secret dimension n, MLWE rank k, modulo q, standard deviation of the error σ and the distribution of secret vector $\mathbf{s}$.

* m^* is the maximum number of allowed samples for each scheme.

* The claimed bit-security are derived from the dual attack for all schemes except for NTRULPrime. NTRULPrime did not consider dual attacks; we use their best claimed result.

* Frodo uses the Frodo model; all the rest schemes use core-SVP model.

* As the parameters of the schemes in round 3 haven't been published until the paper is submitted, we use the parameters proposed in round 2. We will update our results in the final version of this paper, if it is accepted.

Table 8. Kyber's secret distribution

Name	n	k	Probability of		
			0	± 1	± 2
Kyber512	256	2	$\frac{3}{8}$	$\frac{1}{4}$	$\frac{1}{16}$
Kyber768	256	3	$\frac{3}{8}$	$\frac{1}{4}$	$\frac{1}{16}$
Kyber1024	256	4	$\frac{3}{8}$	$\frac{1}{4}$	$\frac{1}{16}$

Table 9. Saber's secret distribution

Name	n	k	Probability of 0	± 1	± 2	± 3	± 4	± 5
Saber512	256	2	0.2460	0.2051	0.1172	0.0439	0.0098	0.0010
Saber768	256	3	0.2734	0.2187	0.1094	0.0313	0.0039	
Saber1024	256	4	0.3124	0.2344	0.0938	0.0156		

Table 10. Frodo's secret distribution

Name	n	Probability of (in multiples of 2^{-16}) 0	± 1	± 2	± 3	± 4	± 5	± 6	± 7	± 8	± 9	± 10	± 11	± 12
Frodo640	640	9288	8720	7216	5264	3384	1918	958	422	164	56	17	4	1
Frodo976	976	11278	10277	7774	4882	2545	1101	396	118	29	6	1		
Frodo1344	1344	18286	14320	6876	2023	364	40	2						

Proof. By definition, we have

$$\begin{aligned}\varepsilon &= \frac{1}{2}\int_{-q/2}^{q/2} |g(x) - f(x)|\, dx \\ &= \frac{1}{2}\left(\int_{I_g} g(x) - f(x)\, dx + \int_{I_f} f(x) - g(x)\, dx\right) \\ &= \int_{I_g} g(x) - f(x)\, dx \qquad \left(\text{as} \int_{-q/2}^{q/2} \big(g(x) - f(x)\big)\, dx = 0\right) \\ &= p_g - p_f.\end{aligned}$$

□

Therefore, we can first compute the subset I_g. Then, for a given target instance t, when $t \in I_g$, we label $t \leftarrow \mathcal{G}_\rho$, otherwise $t \leftarrow \mathcal{U}(-\frac{q}{2}, \frac{q}{2})$. The concrete algorithm is shown in Algorithm 2. Its success probability is shown in Lemma 8.

Lemma 8. *The success probability of Algorithm 2 is* $\frac{1}{2} + \frac{1}{2}\varepsilon$.

Proof. Assume that the target instance t is sampled from either $\mathcal{G}_\rho$ or $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$ with equal probability, i.e., $\frac{1}{2}$. Denote Pr[Gaussian|Gaussian] the probability that the algorithm outputs 'Gaussian' when the input instance is from $\mathcal{G}_\rho$, and Pr[Uniform|Uniform] the probability that the algorithm outputs 'Uniform' when the input instance is from $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$. Then, according to Lemma 7, the success probability can be calculated as follows:

$$\begin{aligned}\Pr[\text{success}] &= \frac{1}{2}\big(\Pr[\text{Gaussian}|\text{Gaussian}] + \Pr[\text{Uniform}|\text{Uniform}]\big)\\ &= \frac{1}{2}\big(p_g + 1 - p_f\big)\\ &= \frac{1}{2}\big(1+\varepsilon\big)\\ &= \frac{1}{2} + \frac{1}{2}\varepsilon\end{aligned}$$

Therefore, the success probability of Algorithm 2 is $\frac{1}{2} + \frac{1}{2}\varepsilon$. □

Algorithm 2: Distinguish $\mathcal{G}_\rho$ from $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$ given one instance

Input: $(\mathbf{A}, \mathbf{b})$, $(\mathbf{w}, \mathbf{v}) \in \Lambda^E_{\text{dual}}(\mathbf{A})$ of length ℓ
Output: Gaussian or Uniform

```
Compute I_g;
Compute t := ⟨w, b⟩ mod q;
if t ∈ I_g then return Gaussian;
else return Uniform;
```

B.2 Extend to M short vectors

Recall that $p_g = \Pr\big[t \in I_g | t \leftarrow \mathcal{G}_\rho\big]$ and $p_f = \Pr\big[t \in I_g | t \leftarrow \mathcal{U}(-\frac{q}{2}, \frac{q}{2})\big]$. If the input distribution is $\mathcal{G}_\rho$, then $p_g M$ of instances are expected to be labeled as from Gaussian distribution; otherwise, $p_f M$. This allows us to setup a threshold $\frac{1}{2}(p_g + p_f)M$. The distinguisher will output $\mathcal{G}_\rho$ if the threshold is reached; $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$ otherwise. The algorithm that captures the above logic is shown in Algorithm 3, and the success probability is captured via Lemma 9.

Lemma 9. *Given M instances $t_{i\in[M]}$ sampled from $\mathcal{G}_\rho$ or $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$, the success probability of Algorithm 3 is at least* $1 - \exp(-\frac{\varepsilon^2}{2}M)$.

To prove the success probability, we will use an additional lemma of Hoeffding's inequality.

Lemma 10 (Hoeffding's inequality.). *Let $Z_1, \ldots, Z_n$ be independent bounded random variables with $Z_i \in [a, b]$ for all i, where $-\infty < a \leq b < \infty$. Then*

$$\Pr\big(\frac{1}{n}\sum_{i=1}^{n}(Z_i - E[Z_i]) \geq t\big) \leq \exp(-\frac{2nt^2}{(b-a)^2})$$

and

$$\Pr\big(\frac{1}{n}\sum_{i=1}^{n}(Z_i - E[Z_i]) \le -t\big) \le \exp(-\frac{2nt^2}{(b-a)^2})$$

for all $t \ge 0$.

Now we prove Lemma 9.

Proof. We assume that the input instances $(\mathbf{A}, \mathbf{b})$ are all from Gaussian or uniform distribution with probability $\frac{1}{2}$ respectively. Each y_i in Algorithm 3 is a indicator variable and let $Y = \sum_{i=1}^{M} y_i$. We consider the probability that the algorithm outputs the wrong distribution in the following two cases:

- Algorithm 3 outputs "Uniform" when the input instances are from Gaussian distribution. Since $(\mathbf{A}, \mathbf{b})$ are from $\mathcal{G}_\rho$, we have $\Pr[y_i = 1] = p_g$ and $E(Y) = p_g M$. Since Algorithm 3 outputs "Uniform", $Y < \frac{1}{2}(p_g + p_f)M$. Using Hoeffding's inequality, we get:

$$\begin{aligned}\Pr[Y < \frac{p_g + p_f}{2}M] &= \Pr[Y - p_g M < -\frac{\varepsilon}{2}M] \\ &\le \exp(-\frac{\varepsilon^2}{2}M).\end{aligned}$$

- Algorithm 3 outputs "Gaussian" when the input instances are from Uniform distribution. Since $(\mathbf{A}, \mathbf{b})$ are from $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$, we have $\Pr[y_i = 1] = p_f$ and $E(Y) = p_f M$. Since Algorithm 3 outputs "Gaussian", $Y \ge \frac{1}{2}(p_g + p_f)M$. Using Hoeffding's inequality, we get:

$$\begin{aligned}\Pr[Y > \frac{p_g + p_f}{2}M] &= \Pr[Y - p_f M > \frac{\varepsilon}{2}M] \\ &\le \exp(-\frac{\varepsilon^2}{2}M)\end{aligned}$$

Therefore the success probability of Algorithm 3 is at least $1 - \exp(-\frac{\varepsilon^2}{2}M)$. □

According to Lemma 9, if we set the target success probability of the dual attack to $1 - \frac{1}{2^\kappa}$, where κ is the security parameter, then, the required number of short vectors can be derived via $1 - exp(-\frac{\varepsilon^2}{2}M) = 1 - \frac{1}{2^\kappa}$. That is, when we use $M \approx \frac{\kappa}{\varepsilon^2}$ short vectors in $\Lambda^E_{\text{dual}}(\mathbf{A})$ of length ℓ, we can distinguish $\mathcal{G}_\rho$ from random with success probability $1 - \frac{1}{2^\kappa}$.

Finally, based on Lemma 7, 8, 9, assuming Heuristic 1 and Assumption 1, we draw the following conclusion for dual attack.

Theorem 6. *Given m normal-form LWE instances* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$ *characterized by* n, σ, q. *Using BKZ with blocksize* β *to obtain a vector* $(\mathbf{w}, \mathbf{v})$ *of length* ℓ *in the dual lattice* $\Lambda^E_{dual}(\mathbf{A}) = \{(\mathbf{w}, \mathbf{v}) \in \mathbb{Z}^m \times \mathbb{Z}^n : \mathbf{w} \cdot \mathbf{A} \equiv \mathbf{v} \bmod q\}$, *the success probability of distinguishing* $t = \langle \mathbf{v}, \mathbf{s} \rangle + \langle \mathbf{w}, \mathbf{e} \rangle$ *from random is at*

Algorithm 3: Distinguish between Gaussian and Uniform Distribution

Input: $(\mathbf{A}, \mathbf{b})$, M short vectors $(\mathbf{w}, \mathbf{v})_i \in \Lambda^E_{\text{dual}}(\mathbf{A})$ for $i \in [M]$, I_g, p_g, p_f
Output: Gaussian or Uniform

```
for i ∈ [M] do
    t_i = ⟨w_i, b⟩ mod q;
    if t_i ∈ I_g then y_i = 1;
    else y_i = 0;
Calculate Y = ∑_i^M y_i;
if Y ≥ (1/2)(p_g + p_f)M then return Gaussian;
else return Uniform;
```

least $p = \frac{1}{2} + 2e^{-2\pi^2\tau^2}$, *where* $\tau = \frac{\ell\sigma}{q}$ *and* $\ell = \delta_0^m q^{\frac{n}{m}}$*. The success probability can be amplified to* $1 - \frac{1}{2^\kappa}$ *by using* $M = \frac{\kappa}{\varepsilon^2}$ *vectors of the same length* ℓ*, where* κ *is a security parameter. Accordingly, the BKZ algorithm should be repeated* $N = \lceil \frac{\kappa}{2^{0.2075\beta} \cdot \varepsilon^2} \rceil$ *times. Then the cost of dual attack is calculated as* $T = N \cdot T_{BKZ}$*.*

C Additional proofs

C.1 Proof for Lemma 3

Proof. We first show that $N = 1$ and $\frac{SV}{2^{0.2075}} \leq M \leq SV$. Note that β is an integer. In the following analysis, we will assume β is a real number and show that the optimal β will satisfy $M(\beta) = SV(\beta)$ and hence $N = 1$. Then when β has to be an integer, we have that the optimal β satisfies $N = 1$ and $\frac{SV}{2^{0.2075}} \leq M \leq SV$, as claimed.

Let β^* be the real number such that $M(\beta^*) = SV(\beta^*)$. We consider two cases when $\beta \leq \beta^*$ and $\beta \geq \beta^*$, and show that in both cases the optimal β is β^*. Since $M(\beta)$ is decreasing in β and $SV(\beta)$ is increasing in β, $\frac{M(\beta)}{SV(\beta)}$ is decreasing in β. Then $\beta \leq \beta^* \Leftrightarrow \frac{M(\beta)}{SV(\beta)} \geq 1$ and $\beta \geq \beta^* \Leftrightarrow \frac{M(\beta)}{SV(\beta)} \leq 1$.

When $\beta \geq \beta^*$ and $\frac{M(\beta)}{SV(\beta)} \leq 1$, we have that $N = \lceil \frac{M(\beta)}{SV(\beta)} \rceil = 1$ and $N \cdot T_{\text{BKZ}} = T_{\text{BKZ}}$ is increasing in β. Then in this case the optimal β minimizing $N \cdot T_{\text{BKZ}} = T_{\text{BKZ}}$ is the minimum β such that $\beta \geq \beta^*$, i.e., the optimal β is β^*.

When $\beta \leq \beta^*$ and $\frac{M(\beta)}{SV(\beta)} \geq 1$, we consider the continuous function $f(\beta)$ corresponding to $N \cdot T_{\text{BKZ}}$ defined as follows:

$$f(\beta) \coloneqq \frac{M(\beta)}{SV(\beta)} \cdot T_{\text{BKZ}(\beta)} = \frac{M(\beta)}{2^{0.2075\beta}} \cdot 2^{0.292\beta} = M(\beta) \cdot 2^{0.0845\beta}.$$

We will show that $f(\beta)$ is decreasing in β. Then in this case the optimal β minimizing $N \cdot T_{\text{BKZ}}$ is the maximum β such that $\beta \leq \beta^*$, i.e., the optimal β is β^*.

Now we show that $\frac{f(\beta+1)}{f(\beta)} \leq 1$. To ease the analysis, we use the following approximation for δ_0:

$$\delta_0 = 2^{\frac{1}{\beta}}.[36]$$

Let m_1 and m_2 be the optimal number of equations to use for β and $\beta+1$ respectively, we have

$$\begin{aligned}
f(\beta+1) &= \frac{\kappa + \ln L}{\varepsilon^2(\beta+1)} \cdot 2^{0.0845(\beta+1)} \\
&= \frac{\kappa + \ln L}{4e^{\frac{-4\pi^2\sigma^2 q^{\frac{2n}{m_2+n}}}{q^2} 2^{\frac{2(m_2+n)}{\beta+1}}}} \cdot 2^{0.0845(\beta+1)} \\
&\leq \frac{\kappa + \ln L}{4e^{\frac{-4\pi^2\sigma^2 q^{\frac{2n}{m_1+n}}}{q^2} 2^{\frac{2(m_1+n)}{\beta+1}}}} \cdot 2^{0.0845(\beta+1)} \\
&= \frac{\kappa + \ln L}{\left(\varepsilon^2(\beta)\right)^{2^{-\frac{2(m_1+n)}{\beta(\beta+1)}}}} \cdot 2^{0.0845(\beta+1)} \\
&= f(\beta) \cdot \left(\varepsilon^2(\beta)\right)^{1-2^{-\frac{2(m_1+n)}{\beta(\beta+1)}}} \cdot 2^{0.0845} \\
&\leq f(\beta) \cdot \left(\varepsilon^2(\beta)\right)^{1-2^{-\frac{2}{\beta}}} \cdot 2^{0.0845}.
\end{aligned}$$

The first inequality holds since m_2 is the optimal number to minimize $\varepsilon(\beta+1)$. The last inequality holds since the BKZ blocksize β should be smaller than the dimension m_1+n of the dual lattice.

Then our goal is to show that

$$g(\beta) \coloneqq \left(\varepsilon^2(\beta)\right)^{1-2^{-\frac{2}{\beta}}} \cdot 2^{0.0845} \leq 1$$

when $\beta > 50$ and $\frac{M(\beta)}{SV(\beta)} \geq 1$. To this end, we give an upper bound for $\varepsilon^2(\beta)$. According to $\frac{M(\beta)}{SV(\beta)} \geq 1$, we have that $M(\beta) = \frac{\kappa+\ln L}{\varepsilon^2(\beta)} \geq SV(\beta) = 2^{0.2075\beta}$, then

$$\varepsilon^2(\beta) \leq 2^{-0.2075\beta}(128 + \ln L). \tag{3}$$

According to $\frac{M(\beta)}{SV(\beta)} \geq 1$ and $T_{\text{guess}} \leq 2^{50} \cdot T_{\text{BKZ}}$, we can upper bound L by that

$$L = \frac{T_{\text{guess}}(\beta)}{M(\beta)} \leq 2^{50} \cdot \frac{T_{\text{BKZ}}(\beta)}{SV(\beta)} = 2^{50+0.0845\beta}.$$

Then it is easy to verify that for any $\beta > 50$,

$$2^{-0.0075\beta}(128 + \ln L) \leq 2^7. \tag{4}$$

Incorporating Equation 4 to Equation 3, we get the upper bound for $\varepsilon^2(\beta)$:

$$\varepsilon^2(\beta) \leq 2^{-0.2\beta+7}. \tag{5}$$

Incorporating Equation 5 to $g(\beta)$ we get

$$g(\beta) \leq 2^{(-0.2\beta+7)(1-2^{-\frac{2}{\beta}})+0.0845}.$$

It is easy to verify that the right side is decreasing in β and for any $\beta > 50$,

$$g(\beta) < 1.$$

This finish the proof for that the optimal β will satisfy $M(\beta) = SV(\beta)$ and $N = 1$. For $\varepsilon^2(\beta) \leq 2^{-0.2\beta+7}$, note that we have already shown this bound in the above when assuming $\frac{M(\beta)}{SV(\beta)} \geq 1$. And we have just showed that the optimal β will satisfy $M(\beta) = SV(\beta)$, so $\varepsilon^2(\beta) \leq 2^{-0.2\beta+7}$ is satisfied by the optimal β. □

C.2 Proof for Lemma 5

Proof. For a fixed r, we can find the corresponding optimal β. Then the advantage is $\varepsilon(r) = 2e^{-2\pi^2\tau^2}$, where $\tau = \frac{\ell\sigma}{q}$ and $\ell = \delta_0^{m+n-r} q^{\frac{n-r}{m+n-r}}$. Once r and β are fixed, it is easy to verify that the optimal number m of equations to use is given by

$$m = \sqrt{\frac{(n-r)\log q}{\log \delta_0}} - (n-r)[33][10],$$

then $\ell = (\delta_0^2)^{\sqrt{\frac{(n-r)\log q}{\log \delta_0}}}$. So the number of samples we need is

$$F(r) := M(r) = \frac{\kappa + \ln L(r)}{\varepsilon^2(r)} = \frac{\kappa + \ln L(r)}{4e^{\frac{-4\pi^2\sigma^2(\delta_0^4)^{\sqrt{\frac{(n-r)\log q}{\log \delta_0}}}}{q^2}}}.$$

To ease the notation, let $X(r) = (\delta_0^4)^{\sqrt{\frac{(n-r)\log q}{\log \delta_0}}}$. Notice that

$$X(r+1) = X(r)^{\sqrt{\frac{n-r-1}{n-r}}},$$

and

$$\varepsilon^2(r+1) = 4e^{\frac{-4\pi^2\sigma^2 X(r+1)}{q^2}} = (\varepsilon^2(r))^{X(r)^{\sqrt{\frac{n-r-1}{n-r}}-1}}.$$

Now

$$\begin{aligned} \frac{F(r+1)}{F(r)} &= \frac{\kappa + \ln L(r+1)}{\kappa + \ln L(r)} \frac{\varepsilon^2(r)}{\varepsilon^2(r+1)} \\ &= \frac{\kappa + \ln L(r+1)}{\kappa + \ln L(r)} \frac{\varepsilon^2(r)}{\left(\varepsilon^2(r)\right)^{X(r)^{\sqrt{\frac{n-r-1}{n-r}}-1}}} \\ &= \frac{\kappa + \ln L(r+1)}{\kappa + \ln L(r)} \left(\varepsilon^2(r)\right)^{1-X(r)^{\sqrt{\frac{n-r-1}{n-r}}-1}}. \end{aligned} \qquad (6)$$

[10] The formular in [33] is $\sqrt{\frac{n\log q}{\log \delta_0}}$ since [33] considers the original dual attack.

Our goal is to show that $F(r)$ decreases when r increases. It suffices to show that $\frac{F(r+1)}{F(r)} < 1$ for any $r \geq 0$. We will upper bound $\frac{\kappa+\ln L(r+1)}{\kappa+\ln L(r)}$ and $\varepsilon^2(r)$, and lower bound $1 - X(r)^{\sqrt{\frac{n-r-1}{n-r}}-1}$ in Equation 6 by functions that only depend on β, and then using these upper bounds to show that $\frac{F(r+1)}{F(r)} < 1$ for any $\beta > 50$.

1. Let $R \leq q$ be the size of the support for each entry of the secret, then

$$\begin{aligned} \frac{\kappa + \ln L(r+1)}{\kappa + \ln L(r)} &= \frac{\kappa + (r+1)\ln R}{\kappa + r \ln R} \\ &\leq \frac{\kappa + \ln R}{\kappa} \\ &\leq \frac{\kappa + \ln 2^{0.0845\beta + 0.2075}}{\kappa} \text{(Assumption 3)} \\ &= \frac{128 + (0.0845\beta + 0.207)\ln 2}{128} \end{aligned} \tag{7}$$

2. According to Lemma 3, we can upper bound $\varepsilon^2(r)$ by that

$$\varepsilon^2(r) \leq 2^{-0.2\beta+7} \tag{8}$$

3. Since $m = \sqrt{\frac{(n-r)\log q}{\log \delta_0}} - (n-r) \geq 0$, $\sqrt{\frac{(n-r)\log q}{\log \delta_0}} \geq (n-r)$. In addition, $\sqrt{\frac{n-r-1}{n-r}} - 1 \leq -\frac{1}{2(n-r)}$. Combining these two inequalities, we get

$$\sqrt{\frac{(n-r)\log q}{\log \delta_0}}(\sqrt{\frac{n-r-1}{n-r}} - 1) \leq -\frac{1}{2}.$$

Then

$$\begin{aligned} 1 - X(r)^{\sqrt{\frac{n-r-1}{n-r}}-1} &= 1 - (\delta_0^4)^{\sqrt{\frac{(n-r)\log q}{\log \delta_0}}(\sqrt{\frac{n-r-1}{n-r}}-1)} \\ &\geq 1 - \delta_0^{-2}. \end{aligned} \tag{9}$$

Note that δ_0 is a function of β.

Now incorporating Equations 7 8 9 into Equation 6, we can upper bound $\frac{F(r+1)}{F(r)}$ by a function of β:

$$\frac{F(r+1)}{F(r)} \leq f(\beta) \coloneqq \frac{128 + (0.0845\beta + 0.2075)\ln 2}{128}(\frac{1}{2})^{(1-\delta_0^{-2})(0.2\beta-7)}.$$

It is easy to verify that for any $50 < \beta < 1990$, $f(\beta) < 1$. We plot the value of $f(\beta)$ for $20 \leq \beta \leq 2000$ in Figure 5.

□

C.3 Proof for Theorem 4

Proof. According to Heuristic 2 and $T_{\text{guess}} = T_{\text{BKZ}}$, we have that the optimal β and r satisfies that $b(r) = b_1 + \alpha r$, and

$$T_{\text{guess}} = T_{\text{BKZ}} = 2^{0.292 \cdot b(r)}.$$

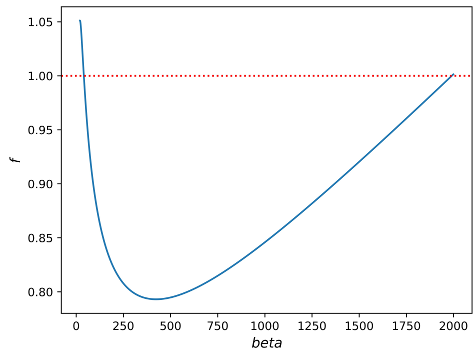


Fig. 5. The value of $f(\beta)$ for $20 \leq \beta \leq 2000$.

Since $N = 1$,

$$M = SV = 2^{0.2075 \cdot b(r)},$$

then

$$L = \frac{T_{\text{guess}}}{M} = 2^{0.0845 \cdot b(r)}.$$

The success probability

$$p_c = g(r, L) = g(r, 2^{0.0845 \cdot b(r)}).$$

Therefore, we get the cost of the attack

$$f(r) = \frac{T_{\text{BKZ}} \cdot N + T_{\text{guess}}}{p_c} = \frac{2T_{\text{BKZ}}}{p_c} = \frac{2^{0.292 \cdot b(r)+1}}{g(r, 2^{0.0845 \cdot b(r)})}.$$

□

C.4 Proof for Theorem 5

Proof. Let $\mathbf{A}_i$ be the i-th row vector of $\mathbf{A}$. We show that $\mathbf{A}_i \cdot \mathbf{S}$ runs in $\mathcal{O}(L)$ time. Then the claim of the theorem follows.

Denote h the maximum number of non-zero entries of all columns of $\mathbf{S}$. We can partition all columns of $\mathbf{S}$ into $h+1$ subsets $\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_h$, where $\mathbf{S}_k$ consists of all columns having k non-zero entries. Since $\mathbf{S}$ is closed, all these subsets are non-empty. Moreover, for any $\mathbf{v} \in \mathbf{S}_k$, there is a vector $\mathbf{v}' \in \mathbf{S}_{k-1}$ such that $\mathbf{v}' \prec \mathbf{v}$. Let $j \in [r]$ be the index such that $\mathbf{v}'_j = 0$, $\mathbf{v}_j \neq 0$, and $\mathbf{v}'_i = \mathbf{v}_i$ for all $i \neq j$. Then, the product of $\langle \mathbf{A}_i, \mathbf{v} \rangle$ can be easily computed based on the product of $\langle \mathbf{A}_i, \mathbf{v}' \rangle$ as follows:

$$\langle \mathbf{A}_i, \mathbf{v} \rangle = \langle \mathbf{A}_i, \mathbf{v}' \rangle + \mathbf{A}_{i,j} \mathbf{v}_j.$$

This can be done in constant time. Hence, when we compute the product of $\mathbf{A}_i$ and $\mathbf{S}$, we can compute the product of $\mathbf{A}_i$ and the columns of $\mathbf{S}$ in the order of increased number of non-zero entries. The result for each column in $\mathbf{S}_0$ can be done in constant time, and result for each columns in $\mathbf{S}_k$ can also be done in constant time given the results for all columns in $\mathbf{S}_{k-1}$. Therefore, the product of $\mathbf{A}_i$ and $\mathbf{S}$ can be done in $\mathcal{O}(L)$ time. □

Remark 5. Note that to ensure the recursive computation in the proof, we need to maintain a function which given a column $\mathbf{v} \in \mathbf{S}$ outputs a column $\mathbf{v}' \in \mathbf{S}$ with $\mathbf{v}' \prec \mathbf{v}$. We can do this once for all A_i in $\mathcal{O}(L^2)$ time. Since under the optimal parameters, we have $ML = T_{\text{guess}} = T_{\text{BKZ}} = 2^{0.292\beta}$ and $M = 2^{0.2075\beta}$, so $L = 2^{0.0845\beta} < M$. Therefore, this additional $\mathcal{O}(L^2)$ does not influence the claimed running time.

About the increased storage space, our algorithm need at most $\mathcal{O}(2^{0.0845\beta})$ bits (recall that $L = 2^{0.0845\beta}$). At first glance, it seems that our algorithm needs ML bits to store the resulting matrix $\mathbf{AS}$. However, it is actually not necessary to store the whole matrix since what we need is the number of entries that are in $(-H, H)$ for each column of $\mathbf{AS}$ (see Algorithm 2). Hence, during our algorithm, we keep a vector of length L to record this number for all columns. And at each step when computing $\mathbf{A}_i\mathbf{S}$, we need to remember at most L numbers to ensure the recursive approach. Therefore, the actual storage space is $\mathcal{O}(2^{0.0845\beta})$ bits, which is negligible compared with the exponential storage space ($\mathcal{O}(2^{0.2075\beta})$) needed for the sieving algorithm.

C.5 Proof for Corollary 1

Proof. For any non-zero candidate vector $\mathbf{v} \in C^*$ and any vector $\mathbf{v}' \prec \mathbf{v}$, we show that $\mathbf{v}' \in C^*$. According to the definition of C^*, it suffices to show that the probability that $\mathbf{v}$ or $\mathbf{v}'$ is the correct $\mathbf{s_1}$ satisfies that $p(\mathbf{v}') \geq p(\mathbf{v})$.

For $\mathcal{B}_h^+$ with $n - r \geq 2h$, assume that the hamming weight of $\mathbf{v}$ and $\mathbf{v}'$ are k and $k-1$, respectively. We have that

$$p(\mathbf{v}) = \frac{\binom{n-r}{h-k}}{\binom{n}{h}}, \text{ and } p(\mathbf{v}') = \frac{\binom{n-r}{h-k+1}}{\binom{n}{h}}.$$

Since $n - r \geq 2h$, we have $p(\mathbf{v}') \geq p(\mathbf{v})$.

For $\mathcal{B}_h^-$ with $n - r \geq 3h$, assume that $\mathbf{v}$ contains k^+ of 1 and k^- of -1. We have that

$$p(\mathbf{v}) = \frac{\binom{n-r}{h-k^+}\binom{n-r-h+k^+}{h-k^-}}{\binom{n}{h}\binom{n-h}{h}} = \frac{\binom{n-r}{h-k^-}\binom{n-r-h+k^-}{h-k^+}}{\binom{n}{h}\binom{n-h}{h}}.$$

Since $\mathbf{v}' \prec \mathbf{v}$, $\mathbf{v}'$ contains one less 1 or one less -1. It's easy to see that in both case we have $p(\mathbf{v})' \geq p(\mathbf{v})$.

For a central discrete distribution, assume that $\mathbf{v}$ contains k_i of $\pm i$ for $i \in [t]$. We have that

$$p(\mathbf{v}) = p_0^{k_0} p_1^{k_1} \cdots k_t^{k_t}.$$

Since $\mathbf{v}' \prec \mathbf{v}$, $\mathbf{v}'$ contains one less non-zero entry. Since $p_0 \geq p_i$ for all $i \in [t]$, we have that $p(\mathbf{v})' \geq p(\mathbf{v})$.

Therefore, for any one of these three distributions, C^* is closed, and according to Theorem 5, the multiplication of $\hat{\mathbf{A}}$ and $\mathbf{S}^*$ can be done in $\mathcal{O}(M \cdot L)$ time. □

D Full comparison results

Table 11. Bit-security estimations

Name	Core-SVP Model						Practical Model					
	Classical			Quantum			Classical			Quantum		
	Dual	Hybrid 2M		Dual	Hybrid 2M		Dual	Hybrid 2M		Dual	Hybrid 2M	
	λ	λ	r	λ	λ	r	λ	λ	r	λ	λ	r
Kyber512	112	108	14	101	99	9	141	134	29	131	125	23
Kyber768	182	176	24	165	161	16	212	202	38	195	188	31
Kyber1024	254	245	34	231	225	23	284	271	49	261	252	38
Saber512	117	115	11	107	105	8	147	141	22	136	132	18
Saber768	189	184	20	172	169	13	219	211	32	202	196	25
Saber1024	258	250	30	235	230	20	289	277	43	265	257	33
Dilithium768	100	99	6	91	90	3	130	128	15	121	119	11
Dilithium1024	142	140	10	129	128	7	172	169	18	159	157	15
Dilithium1280	175	172	16	158	157	11	205	201	26	189	186	21
Frodo640	142	139	11	129	127	6	172	167	19	159	155	15
Frodo976	206	202	17	187	185	10	236	230	27	217	213	19
Frodo1344	271	264	29	246	242	19	301	292	41	276	270	30
NTRULPrime653	131	125	26	118	115	16	160	149	48	148	140	39
NTRULPrime761	155	148	36	141	137	24	185	172	64	171	161	52
NTRULPrime857	177	168	44	160	155	30	207	192	77	190	180	63