Bi *et al. Cybersecurity* (2022) 5:15
https://doi.org/10.1186/s42400-022-00115-y

Cybersecurity

RESEARCH Open Access

Hybrid dual attack on LWE with arbitrary secrets

Lei Bi[1,2*], Xianhui Lu[1,2,5], Junjie Luo[3], Kunpeng Wang[1,2] and Zhenfei Zhang[4]

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 for most cryptographic use cases a hybrid dual attack outperforms 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 2-15 bits, under the BKZ-core-SVP model.

Keywords: Learning with errors, Lattice-based cryptography, Cryptanalysis, Dual attack, Hybrid attack, NIST PQC

Introduction

The learning with errors (LWE) problem, introduced by Regev (2009) 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 (NIST-round-3 2020 of National Institute of Standards and Technology's post-quantum cryptography standardization process (NIST-PQC), namely Dilithium (Ducas et al. 2018), Kyber (Bos et al. 2018b), Saber (D'Anvers et al. 2018), Frodo (Bos et al. 2018a) and NTRULPrime (Bernstein et al. 2017). This process has sparked a long list of cryptanalytic advancements (Albrecht 2017; Albrecht et al. 2015a, 2017, 2018; Buchmann et al. 2016; Cheon et al. 2019; Dachman-Soled et al. 2020; Espitau et al. 2020; Son and Cheon 2019), 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 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 (Albrecht et al. 2015a), 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) (Arora and Ge 2011; Albrecht et al. 2015b) 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) (Lindner and Peikert 2011) and the primal attack (Albrecht et al. 2017).
- The last strategy solves decisional LWE by reducing it to a Short Integer Solutions (SIS) problem. There are

*Correspondence: bilei@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
Full list of author information is available at the end of the article

also two subsequent attacks: the combinatorial attack (i.e., using BKW algorithm) (Albrecht et al. 2014) and the dual attack (Albrecht 2017).

In a later paper, Albrecht et al. (2018) 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 (Lindner and Peikert 2011) 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 Chen and Nguyen 2011). 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:

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

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 (Alkim et al. 2016).

One may additionally combine the above attacks with guessing. This method is known as the *hybrid attacks* in the literature (Albrecht 2017; Buchmann et al. 2016; Cheon et al. 2019; Espitau et al. 2020; Hoffstein et al. 2017; Howgrave-Graham 2007; Son and Cheon 2019; Wunderer 2018, 2019). Informally, a hybrid attack guesses part of the secret and performs some attack on the remaining part. As guessing 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 the 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 needs to solve a CVP (a decoding problem) rather than a uSVP (a primal attack) after guessing (see Son and Cheon 2019 for more details about the reduction). As a rule of thumb, a decoding attack requires a 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.

Table 1 Hybrid attacks on LWE

	Lattice	Guessing	Secret
Buchmann et al. (2016)	Decoding	MITM	Small
Son and Cheon (2019)	Decoding + Primal	MITM	Small + sparse
Albrecht (2017)	Dual	Pruning	Small + sparse
Cheon et al. (2019)	Dual	MITM	Small + sparse
Espitau et al. (2020)	Dual	Matrix Mul.	Small
This paper	Dual	Opt. Pruning + Mat. Mul.	Arbitrary

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 (2007) to analyze NTRU (Hoffstein et al. 1998). In the recent years, hybrid attacks have been extensively studied for LWE with sparse and/or small secrets. We summarize those results in Table 1. The first work of hybrid attack on LWE (Buchmann et al. 2016) combined decoding attack with meet-in-the-middle (MITM) technique. Then a similar approach was conducted on primal lattices (Son and Cheon 2019). Albrecht (2017) proposed the framework of hybrid dual attack and applied it over LWE with sparse and binary/ternary secrets. Cheon et al. (2019)

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. (2020) 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.

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:

For most cryptographic use cases, hybrid dual attacks out-perform dual attacks, regardless of the secret distribution.

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 theorem.

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 the same level of success rate will be reduced.*

We will provide our intuition shortly. The proof will be present in "The advantage of the hybrid dual attack" section. For LWE with *short* secrets, it is straightforward to see that the observation is implied by Theorem 1. For LWE with *large* secrets, when enough LWE samples are given, we normalize it and invoke Theorem 1. The only remaining case is LWE with large secrets and limited samples, for which we study separately in "Hybrid attack on uniform secrets" section.

To quantify the decreasing speed of β as r increases, we make an additional Heuristic 3 with justification in "Predicting improvement of Hybrid 1" section. Based on this heuristic, we give a prediction of the improvement of hybrid dual attack over dual attack in Theorem 2.

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 (Code for this paper 2019) For example, one may apply our estimator to other LWE based schemes, such as FHE (Gentry 2009; Brakerski et al. 2012; Gentry et al. 2013) or lattice-based ZK proofs (Bootle et al. 2019, 2020; Esgin et al. 2019).

From the practical side, we re-evaluate *all* LWE-related candidates of NIST-PQC round 3 (NIST-round-3 2020), namely, Dilithium (Ducas et al. 2018), Kyber (Bos et al. 2018b), Saber (D'Anvers et al. 2018), Frodo (Bos et al. 2018a) and NTRULPrime (Bernstein et al. 2017), and compare our results with the most prominent primal attack and standalone dual attack. An important issue when comparing primal attack and (hybrid) dual attack is the assumption about the short vectors produced by the BKZ algorithm. The optimistic assumption (Alkim et al. 2016), which we call Assumption 1, assumes that when using sieving as the SVP oracle, BKZ algorithm with blocksize β provides $2^{0.2075\beta}$ short vectors that are almost as short as the shortest one. However, this assumption has been criticized to be too optimistic on the attacker's ability (see the supporting document for Kyber, Frodo, and Dilithium). A more realistic assumption (Ducas 2018), which we call Assumption 2, assumes most of these $2^{0.2075\beta}$ vectors are $\sqrt{\frac{4}{3}}$ longer than the shortest one. We compare our results under both assumptions.

Our results under the classical core-SVP model (Alkim et al. 2016; Albrecht et al. 2015a, 2018) are summarized in Table 2. We will give more details on the estimations in "Security estimations" section. Compared with standalone dual attack, we improve the results by 2–13 bits under Assumption 1 and 2–15 bits under Assumption 2. Compared with the state-of-the-art cryptanalytic results, which is usually given by primal attack, we improve the results by 2–15 bits under Assumption 1.[1] Even under Assumption 2, for NTRULPrime we improve the results by 1–7 bits, and for other candidates, our results is close to that of primal attack and the difference is within 2 bits. We believe that hybrid dual attacks should be considered for cryptanalysis on any future practical lattice-based cryptosystem.

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.

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, for most cryptographic use cases, Hybrid 1 can outperform the dual attack, *regardless the secret distribution*. We further assert that optimal blocksize of the BKZ

[1] 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	Sec. level	Claim	Assumption 1				Assumption 2			
			Dual	Ours	Δ_{dual}	Δ_{claim}	Dual	Ours	Δ_{dual}	Δ_{claim}
Kyber512	1	118	117	114	−3	−4	122	119	−3	+1
Kyber768	3	182	181	175	−6	−7	188	182	−6	0
Kyber1024	5	256	253	245	−8	−11	263	254	−9	−2
Saber512	1	118	117	114	−3	−4	122	119	−3	+1
Saber768	3	189	189	184	−5	−5	196	191	−5	+2
Saber1024	5	260	258	250	−8	−10	268	260	-8	0
Dilithium1024	2	123	123	121	−2	−2	126	124	−2	+1
Dilithium1280	3	182	181	179	−2	−3	186	183	−3	+1
Dilithium1792	5	252	251	246	−5	−6	257	252	−5	0
Frodo640	1	150	141	139	−2	–	147	145	−2	–
Frodo976	3	215	205	202	−3	–	212	209	−3	–
Frodo1344	5	280	270	264	−6	–	278	272	−6	–
NTRULPrime653	1	130	130	125	−5	−5	135	129	−6	−1
NTRULPrime761	2	155	155	148	−7	−8	161	153	−8	−2
NTRULPrime857	2	176	176	168	−8	−6	183	174	−9	−2
NTRULPrime953	3	197	195	187	−8	−10	202	193	−9	−4
NTRULPrime1013	4	210	209	200	−9	−10	217	207	−10	−3
NTRULPrime1277	5	271	269	256	−13	−15	279	264	−15	−7

* Data for "Ours" uses Hybrid 2m estimator. Δ_{dual} is the improvement over dual attack. Δ_{claim} is the improvement over the claimed results.

* For a fair comparison, data for "Dual" also comes from our estimator.

* The claimed results of Frodo use a different cost model, thus we do not compare our results with the claimed results

decreases linearly as the guess dimension increases, i.e., Heuristic 3, 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, Heuristic 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 attacks 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 this, 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 non-trivial generalization of the algorithm in Espitau et al. (2020). 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^r, where Q is a set of ℓ numbers, Espitau

et al. (2020)'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 and the assumption on the length of short vectors produced by BKZ. For example, Hybrid 1 will always out-perform dual attack, for core-SVP model or Practical model; the actual gain will vary depending on the cost model and the assumption, nonetheless. For consistency and a fair comparison, we will adopt the core-SVP model and Assumption 1 throughout the rest of the paper, unless otherwise stated.

Organization

We begin with some preliminaries in "Preliminaries" section. In "Hybrid attack on arbitrary secrets" section we present the hybrid attack on arbitrary secrets (Hybrid 1) and show its advantage over the standalone dual attack. In "Hybrid dual attack with optimal pruning" section we present the method of optimal pruning in the guessing phase for Hybrid 2 and analyze the advantage of Hybrid 2 over Hybrid 1. We give an additional efficient matrix multiplication in "An additional optimization" section. In "Security estimations" section, we conclude our paper with estimations for 5 NIST-PQC candidates.

Preliminaries

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. $\mathbf{A}$. 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.

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

$$\mathcal{P}(\mathbf{B}) = \mathbf{B} \cdot \left[-\frac{1}{2}, \frac{1}{2} \right)^n = \left\{ \sum_{i \in [n]} c_i \cdot \mathbf{b}_i : c_i \in [-\frac{1}{2}, \frac{1}{2}) \right\}.$$

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 (Chen and Nguyen 2011) 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 Chen (2013).

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 (Albrecht et al. 2018) for more details.

Let us first list relevant cost models in this paper. As mentioned earlier, we will be focusing on the core-SVP model with sieving (Alkim et al. 2016).

$$T_{\mathrm{BKZ}}(m,\beta)=\begin{cases}2^{0.292\beta}, \text{ classical}\\ 2^{0.257\beta}, \text{ quantum}\end{cases}$$

We will also briefly compare with the practical model, used by, for example (Albrecht 2017), where the number of calls is $8m$ rather than 1.

$$T_{\mathrm{BKZ}}(m,\beta)=\begin{cases}8m\cdot 2^{0.292\beta+16.4}, \text{ classical}\\ 8m\cdot 2^{0.257\beta+16.4}, \text{ quantum}\end{cases}$$

Note that the classical and quantum complexity of sieving are from Becker et al. (2016) and Chailloux and Loyer (2021),respectively.

In addition, when using sieving as the SVP oracle, Alkim et al. (2016) made an assumption on the output short vectors from BKZ. Alkim et al. (2016) 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.

Assumption 1 (Alkim et al. 2016) When using sieving as the SVP oracle, the 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.

This assumption has been adopted by many LWE related proposals in round 3 finalists of NIST [38]: see Section 5.1.3 of the supporting documentation for Kyber, Section 5.2.3 for Frodo, Dilithium follows (Alkim et al. 2016); also see Section 6.1 of D'Anvers et al. (2018) and Section 2.3 of Espitau et al. (2020). To give a fair comparison, we follow this line of work and adopt this assumption when analyzing the schemes in "Security estimations" section. Nonetheless, we note that Assumption 1 is very optimistic on the attacker's capability. In practice, most of the output vectors from sieving could be $\sqrt{\frac{4}{3}}$ longer than the shortest one.

Assumption 2 (Ducas 2018) When using sieving as the SVP oracle, the BKZ algorithm with blocksize β provides $2^{0.2075\beta}$ short vectors in one run, and most of them are $\sqrt{\frac{4}{3}}$ longer than the shortest one produced by BKZ algorithm.

For consistency, we will focus on Assumption 1 throughout the rest of the paper, except for Section 3.5.

We emphasize that Assumptions 1 and 2 marginally affect the quality of our improvement. They do not change the fact that hybrid dual attacks are better than dual attacks. More concretely, under Assumption 1, the improvement of Hybrid 2m over dual attack will be 2-14 bits; this changes to 2-15 bits under Assumption 2. See "Security estimations" section for more details.

For completeness, in "Advantage under different cost models andassumptions" section, we will compare our advantage under three assumptions, namely, Assumptions 1 and 2 and the amortized cost method (Albrecht 2017), where the large number of short vectors are provided by using LLL instead of sieving. The advantage of Hybrid 2m over dual attack under different cost models and assumptions is given in Table 12.

The learning with errors problem

The learning with errors (LWE) problem, introduced by Regev (2009), 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 $\mathrm{LWE}_{n,q,\mathbf{s},\chi}$ 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 $\mathrm{LWE}_{n,q,\mathbf{s},\chi}$ distribution, we define two 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 $\mathbf{s}\leftarrow\mathcal{S},\mathbf{A}\xleftarrow{\$}\mathbb{Z}_q^{m\times n},\mathbf{e}\xleftarrow{\$}\chi^m,\mathbf{b}\in\mathbb{Z}_q^m$.

A 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 (Applebaum et al. 2009) *Given the instances* $(\mathbf{a},b=\langle\mathbf{a},\mathbf{s}\rangle+e)$ *sampled from* $\mathrm{LWE}_{n,q,\mathbf{s},\chi}$ *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.*

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 (2009), 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 $\text{LWE}_{\mathbf{s},\sigma}$, then, $\mathbf{b} = \mathbf{As} + \mathbf{e} \bmod q$. In this case, given a short vector $\mathbf{w}$, we have

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

which will follow a modular Gaussian distribution. Otherwise, $\langle \mathbf{w}, \mathbf{b} \rangle \bmod q$ 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. (2016) presented an improved dual attack on normal-form LWE, which tries to solve an inhomogeneous SIS problem, and works over the *embedded lattice*:

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

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

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

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

Assumption 3 (Chillotti et al. 2020; Espitau et al. 2020) 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 $t := \langle \mathbf{v}, \mathbf{s} \rangle + \langle \mathbf{w}, \mathbf{e} \rangle$ can be viewed as a Gaussian distribution $\mathcal{G}_\rho$ with mean 0 and standard deviation $\rho = \ell\sigma$ (Alkim et al. 2016). Then the maximal variance distance between modular Gaussian distribution $t \bmod q$ and uniform distribution $\mathcal{U}(-\frac{q}{2}, \frac{q}{2})$ is bounded by $\varepsilon = 4\exp(-2\pi^2\tau^2)$, where $\tau = \ell\sigma/q$ (Alkim et al. 2016). According to these, the advantage of the attack is summarized in the following heuristic.

Heuristic 2 (Alkim et al. 2016) *Given m normal-form LWE instances $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e} \bmod q)$ characterized by n, σ, q, and a vector $(\mathbf{w}, \mathbf{v}) \in \Lambda_{\text{dual}}^{E}$ of length ℓ, the dual*

attack solves the decision-LWE with advantage $\varepsilon = 4\exp(-2\pi^2\tau^2)$ *where* $\tau = \frac{\ell\sigma}{q}$. *The success probability of the attack can be amplified to be at least* $\frac{1}{2}$ *by using about* $1/\varepsilon^2$ *many such vectors* $(\mathbf{w}, \mathbf{v}) \in \Lambda^E_{\text{dual}}$ *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.

This attack (Alkim et al. 2016) was initially designed for normal-form LWE. When the secret does not match the error distribution, the attack also works via the scaling technique (Albrecht 2017). For the remaining part of this paper, we will also adopt this technique.

Note that the (Micciancio and Regev 2009) dual attack (referred to as *original dual attack*) works for arbitrary secrets; while the (Alkim et al. 2016) 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.

Hybrid attack on arbitrary 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 first give the framework of our hybrid dual attack (which is the same as that in Albrecht 2017) and its analysis in "The framework" and "Analysis" sections. In "The advantage of the hybrid dual attack" section, we conduct an extensive analysis of the advantage of Hybrid 1 over a standalone dual attack, which is our main contribution in this section. We further derive a formula to predict the improvement of Hybrid 1 in "Predicting improvement of Hybrid 1" section and compare the improvement under different cost models and assumptions in "Advantage under different cost models andassumptions" section. Finally, we study Hybrid 1 on LWE with uniform secrets in Section 3.6.

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:

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

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 $\mathbf{A}_2$:

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

$\Lambda^E_{\text{dual}}(\mathbf{A}_2)$ has a dimension of $d = m + n - r$ and a volume of q^{n-r} with high probability. Then, we assume that with lattice reduction algorithms we will obtain some short vector(s) $(\mathbf{w}, \mathbf{v}) \in \Lambda^E_{\text{dual}}$ 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 modular 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 "Analysis" section. 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 $\mathbf{P} \xleftarrow{\$}$ 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 4.1
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 modular Gaussian distribution* **then**
10 **return** *LWE distribution*;
11 **return** *Uniform;*

Note 2 We note that the framework is not efficient for LWE with large or uniform secrets, as the number of samples we need will be very large. However, this inefficiency is note caused by the hybrid framework, but by embedded dual attack itself. We will provide methods to deal with this problem in "Hybrid attack on uniform secrets" *section*.

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 "Guess with pruning" section 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 the 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. 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 $(\mathbf{w}_i, \mathbf{v}_i) \in \Lambda^E_{\text{dual}}(\mathbf{A}_2)$ 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 $\Lambda^E_{\text{dual}}(\mathbf{A}_2)$,
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 "An additional optimization" section.

In summary, under Assumption 1 and Heuristic 2 for dual attacks, we have the results for hybrid dual attacks as follows.

Lemma 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* $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e}) \bmod q$ *characterized by* n, σ, q *or they are uniformly random. The success probability* $p = p_c \cdot p_s$, *where* p_c *is presented in*"*Guess with pruning*" *section and* $p_s = 1 - \frac{1}{2^\kappa}$, *where* κ *is a security parameter. The cost of dual attack is calculated as*

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

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* $T_{\text{guess}} = M \cdot L \cdot r$ (*see*"*An additional optimization*" *section for an improvement of* T_{guess}).

Remark 1

We will take κ *as an arbitrary number from [0, 10] for the rest of the paper. In "Security estimations" section when we estimate schemes we set* $\kappa = 7$ *such that* $p_s > 0.99$, *the*

same as Albrecht et al. (2018). Notice that the value of κ makes little difference for the final estimations.

The advantage of the hybrid dual attack

We analyze the advantage of the hybrid dual attack by comparing the dual attack and Hybrid 1 (Algorithm 1 with exhaustive search). Since we always set the probability $p_s = 1 - \frac{1}{2^\kappa}$, it is safe to ignore p_s. Then we just need to compare the running time.

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 *If $\beta \geq 50$, for a fixed r such that $T_{\text{guess}} \leq 2^{50} \cdot T_{\text{BKZ}}$,[2] the optimal β that minimizes $N \cdot T_{\text{BKZ}}$ will satisfy $N = 1$ and $\frac{SV}{2^{0.2075}} \leq M \leq SV$.*

Proof (Proof sketch) The full proof is deferred to Appendix A.3.1. 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:

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

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^*. □

Next, we study the influence of the guessing dimension r on the number of required short vectors $M = \frac{\kappa + \ln L}{\varepsilon^2}$. In Hybrid 1 when we guess r dimensions, the benefit is that the advantage ε will be increased, which will decrease M. On the other hand, the number L of guessing candidates increases with r, which will increase M. The key problem is how does M change when r increases. Our estimator shows that for all 5 schemes tested in "Security estimations" section M decreases when r increases. This can be intuitively explained by the fact that $\ln L = r \ln R$, where R is the size of the support for each entry of the secret, increases linearly in r while ε^2 increases exponentially in r (from $2^{-\mathcal{O}(n)}$ when $r = 0$ to $\mathcal{O}(1)$ when $r = n$).

We assume for now that M is decreasing in r and use this to explain why Hybrid 1 outperforms dual attack. Then, we will show that this condition, M is decreasing in r, is satisfied by most cryptosystems.

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

Proof To ease the 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 $M = SV$,[3] 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 outperforms the dual attack. For dual attack we have $T_{\text{dual}} = T_{\text{BKZ}-\text{d}}$ and for Hybrid 1 we have $T_{\text{Hybrid 1}} = T_{\text{BKZ}-\text{h}} + T_{\text{guess}}$. Note that we can take dual attack as a special case of Hybrid 1 with $r = 0$ and $T_{\text{guess}} = 0$. According to Lemma 3 and Lemma 4, 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 $T_{\text{guess}} = T_{\text{BKZ}}$) and decrease β such that the overall running time $T_{\text{Hybrid 1}} = T_{\text{BKZ}-\text{h}} + T_{\text{guess}}$ 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.

[2] 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 "Security estimations" section have $T_{\text{guess}} \leq T_{\text{BKZ}}$ under the optimal parameters. So it is safe to assume that $T_{\text{guess}} \leq 2^{50} \cdot T_{\text{BKZ}}$.*

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

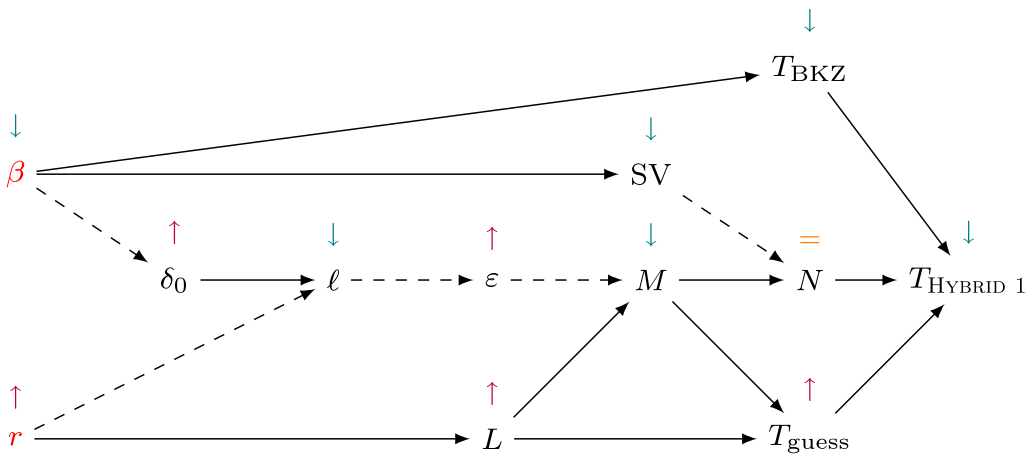

Fig. 1 Parameter relations and their value changes from dual attack to HYBRID 1. An arrow "→" (respectively, "- - →") from node *A* to node *B* means that increasing *A* will increase (respectively, decrease) *B*. "↑" and "↓" shows the direction that the values change from the dual attack to HYBRID 1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

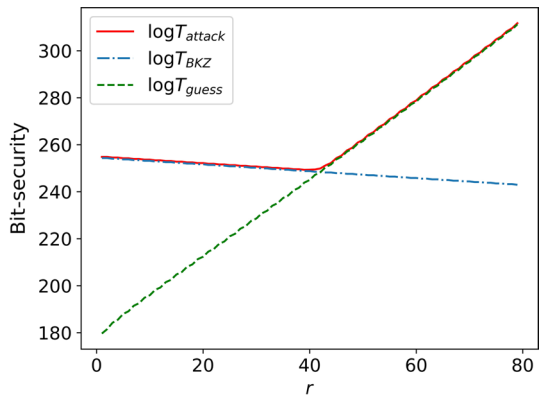

Fig. 2 Example: $T_{\mathrm{Hybrid\ 1}}$, T_{BKZ} and T_{guess} for Kyber1024

Example

To give a more intuitive explanation, we take Kyber1024 as an example and use a figure to show how T_{BKZ}, T_{guess}, and $T_{\mathrm{Hybrid\ 1}}$ change as r increases. In fact, T_{guess} and $T_{\mathrm{Hybrid\ 1}}$ depend on both r and β. However, since we need to guarantee $N = 1$ (according to Lemma 3), the value of β can be determined once the value of r is chosen. This allows us to estimate T_{BKZ}, T_{guess}, and $T_{\mathrm{Hybrid\ 1}}$ as functions of r. The results are shown in Fig. 2. As expected, as r increases (and β decreases), T_{guess} increases and T_{BKZ} decreases. Hence, as r increases, $T_{\mathrm{Hybrid\ 1}}$ first decreases and then increases, and the optimal $T_{\mathrm{Hybrid\ 1}}$ is achieved when the two lines cross. From Fig. 2, we can see that the cross point ($T_{\mathrm{Hybrid\ 1}}$) is smaller than the starting point, which has $r = 0$ and represents a standalone dual attack.

M is decreasing in r

Recall that $M = \frac{\kappa + \ln L}{\varepsilon^2}$ and the intuition for the decreasing is that $\ln L = r \ln R$ increases linearly in r while ε^2 increases exponentially in r. However, consider the extreme case when r increases from 0 to 1, if we set $\kappa = 1$, then $\kappa + \ln L$ is increased from 1 to $1 + \ln R$, which is a vary large increase. Therefore, when r is very small, M could be actually increasing in r, but in the long run, M is decreasing in r. In this part, we show that when $r \geq 2$ M is decreasing in r, under two minor assumptions. First, we assume $\beta \geq 150$, which implies that the cost of the BKZ is larger than 44 bits. This covers most cryptographic use cases, specifically, all 5 schemes tested in "Security estimations" section, whose optimal β is larger than 300. The second assumption is that the optimal number m of equations for the dual attack is at least $\frac{n}{2}$, which is again satisfied by most cryptographic use cases, and for all 5 schemes tested in "Security estimations" section, the optimal m is close to n. We state this formally in the following assumption.

Assumption 4 Assume $\beta \geq 150$ and the optimal number m of equations for the dual attack satisfies that $m \geq \frac{n}{2}$.

Now we can show that M decreases when r increases.

Lemma 5 *Under Assumption 4, the number of short vectors required to achieve the success probability p_s, denoted by M, is decreasing in the guessing dimension r for any $r \geq 2$.*

The proof is deferred to Appendix A.3.2. The idea is to firstly upper bound $\frac{M(r+1)}{M(r)}$ by a function that only depends on β. Then it becomes easy to derive the condition on β that ensures M decreases with r.

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

Theorem 1(Formal) For Hybrid 1 under the core-SVP model, for any LWE instance with arbitrary secrets, under Assumption 4, the optimal BKZ blocksize β that minimizes $N \cdot T_{\mathrm{BKZ}}$ and maintains the same level of success rate is decreasing in the guessing dimension r when $r \geq 2$.

Predicting improvement of Hybrid 1

We now proceed to a predictor that estimates the advantage of Hybrid 1 over dual attacks under the aforementioned core-SVP model. We give our theoretical results in Theorem 2. 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.

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. Intuitively, the slope should be close to $\frac{\beta^*}{n}$, where β^* is the optimal β for dual attack, as the optimal β decreases from β^* to 0 if we increase r from 0 to n. 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 3 *Fix $N = 1$. The optimal β decreases linearly as r increases. The slope, denoted by α, for 5 schemes are shown in Table 10 in Appendix A.2.*

Next, our simulations show that the optimal r and β for Hybrid 1 will satisfy $T_{\mathrm{BKZ}} \approx T_{\mathrm{guess}}$, 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 $T_{\mathrm{BKZ}} = T_{\mathrm{guess}}$ and take parameters r and β as real numbers in our predictor. Since $N = 1$ (Lemma 3), we have $T_{\mathrm{Hybrid\,1}} = 2T_{\mathrm{BKZ}} = 2T_{\mathrm{guess}}$. Note that this approximation differs from the optimal $T_{\mathrm{Hybrid\,1}}$ 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 2.

Theorem 2 *Let R be the size of the support for each entry of the secret and let b_1 be the optimal β for the dual attack. Using Heuristic 3 and assuming $T_{\mathrm{BKZ}} = T_{\mathrm{guess}}$ for Hybrid 1, then the cost of Hybrid 1 is $T_{\mathrm{Hybrid\,1}} = 2^{0.292b_2+1}$, 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 $M = SV = 2^{0.2075b_2}$ (when β is taken as a real number). Using $T_{\mathrm{guess}} = T_{\mathrm{BKZ}} = 2^{0.292b_2}$, we get

$$L = \frac{T_{\mathrm{guess}}}{M} = \frac{T_{\mathrm{BKZ}}}{SV} = 2^{0.0845b_2}.$$

Since $L = R^r$, we get

$$r = \frac{0.0845b_2}{\log R}.$$

According to Heuristic 3,

$$b_2 - b_1 = \alpha r \Rightarrow r = \frac{b_2 - b_1}{\alpha}.$$

Combining $r = \frac{0.0845b_2}{\log R}$ and $r = \frac{b_2 - b_1}{\alpha}$, we get

$$b_2 = b_1 \cdot \frac{\log R}{\log R - 0.0845\alpha}.$$

□

We use the result of Theorem 2 to predict the bit-security of all 5 schemes. 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. The results can be found in Table 10 in Appendix A.2.

Advantage under different cost models and assumptions

In this section, we take Kyber1024 as an example to compare the improvement of Hybrid 1 over dual attack under different cost models, the core-SVP model and the practical model, and different assumptions, Assumption 1, Assumption 2, and Amortising cost method (Albrecht 2017) (see "Lattices and lattice reductions" section).

In the proof of Theorem 2, we have $L = \frac{T_{\mathrm{BKZ}}}{SV} = 2^{0.0845b_2}$. This means the guessing space (L) is determined by the gap between the running time of BKZ (T_{BKZ}) and the number of short vectors produced by sieving (SV). In addition, Theorem 2 shows $b_2 = b_1 \cdot \frac{\log R}{\log R - 0.0845\alpha}$, then $b_1 - b_2 = b_1 \cdot \frac{-0.0845\alpha}{\log R - 0.0845\alpha}$ (recall that $\alpha < 0$). If we switch to Assumption 2, then b_1 becomes larger, and hence the improvement of Hybrid 1 over dual attack ($b_1 - b_2$) will be larger.

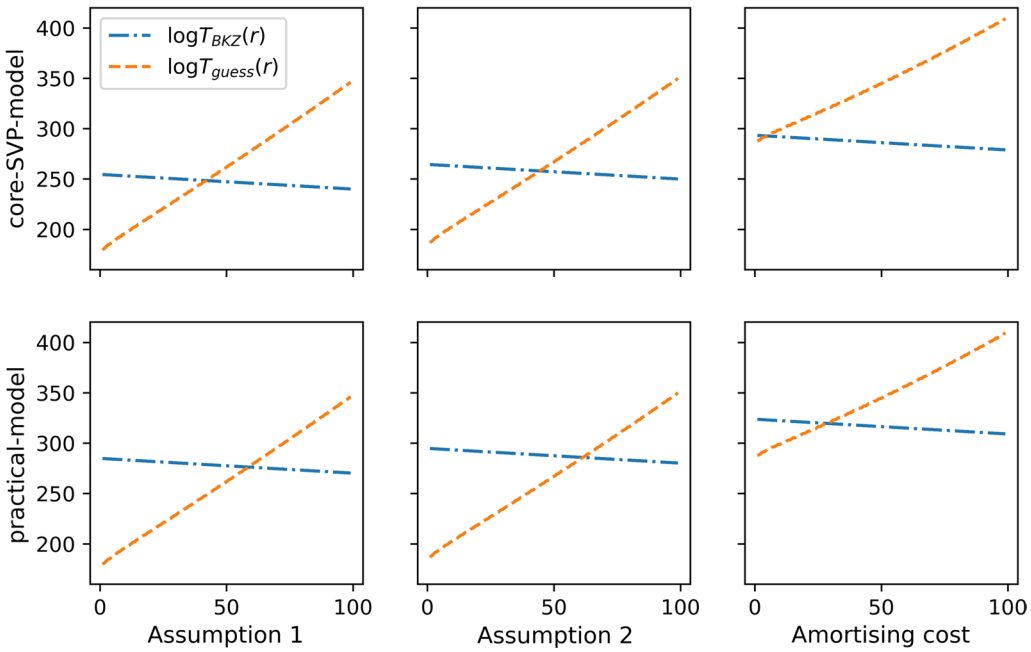

Fig. 3 Example: T_{BKZ} and T_{guess} as a function of r for Kyber1024 under different cost models and assumptions

Taking Kyber1024 as an example, Fig. 3 shows T_{BKZ} and T_{guess} as a function of the guessing dimension r under different cost models and assumptions/method. For Assumptions 1 and 2, the advantage of Hybrid 1 is apparent under both cost models. The advantage is slightly larger under the practical model since here the gap between T_{BKZ} and SV is $8d \cdot 2^{0.0845\beta+16.4}$, where d is the dimension of the dual lattice, which is greater than the gap $2^{0.0845\beta}$ under the core-SVP model.

For amortized cost method, we first run BKZ once and then re-randomize the basis and run LLL M times to produce M short vectors (see Albrecht et al. 2017 for details). The optimal blocksize will balance the cost of BKZ and the cost of repeated LLL. Assume these two costs are equal, then the overall cost to produce M short vectors is close to the cost of repeating LLL M times. Then the gap between the overall cost and M is essentially the cost of running LLL once. Consequently, under the core-SVP model, the advantage of Hybrid 1 is very small as the additional cost from LLL under this cost model is only 0.584 bit; while under the practical model, the advantage of Hybrid 1 is larger as the cost of LLL under this cost model becomes larger.

Hybrid attack on uniform secrets

The framework in "The framework" section is not efficient for LWE with large or uniform secrets, as the number of samples we need will be large. 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 "Hybrid attack on arbitrary secrets" section. 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 "Hybrid attack on arbitrary secrets" section, 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 2, via setting $R = q$, and α to a value close to -1 for simplicity (Table 10 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 2, we have

Table 3 (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 4 (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

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

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 (Regev 2009) 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 and 3 bits in Table 3.

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 4. 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.

Tables 3 and 4 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 ($m \in (0, 2n)$).

Hybrid dual attack with optimal pruning

We proceed with our hybrid dual attack combined with optimal pruning. We name this strategy Hybrid 2. After presenting the method of optimal pruning for different secret distributions in "Hybrid dual attack with optimal pruning" section, we analyze the advantage of Hybrid 2 in "Advantage under different cost models andassumptions" section and give a prediction for the improvement of Hybrid 2 in "Predicting improvement of Hybrid 2" section.

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 expensive 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 $\left(N \cdot T_{\text{BKZ}} + T_{\text{guess}}\right)/p_c$ 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 "Security estimations" section, the probability that the attack fails after repeat is at most 2^{-19} under the optimal parameters, with an exception of NTRULPrime1277, for which the fail probability is 0.01. 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 $\mathcal{B}_h^+$

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 $k_{min} = \max\{0, h + r - n\}$ and $k_{max} = \min\{h, r\}$.

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 3 *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 $\mathcal{B}_h^-$

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

$$N(k^+,k^-) = \binom{r}{k^+}\binom{r-k^+}{k^-}$$

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

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

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

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

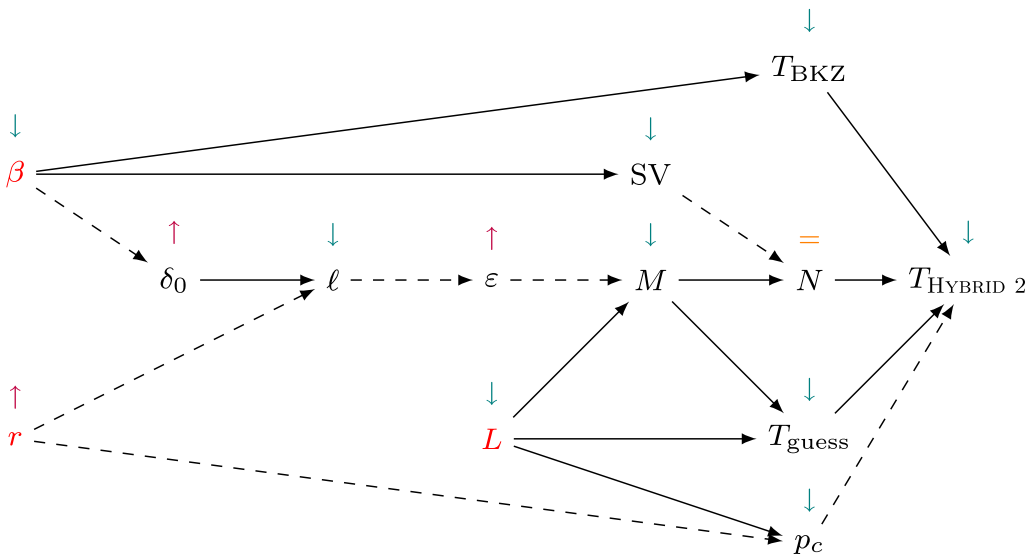

Fig. 4 Parameter relations in HYBRID 2 and value changes from HYBRID 1 to HYBRID 2. The attack can choose the value for parameters in red color (i.e., β, r and L), which then determine the value for other parameters

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

$$T_{\mathrm{guess}} = M \cdot \sum_{S_{(i_0,\ldots,i_t)} \in C} N(i_0, \ldots, i_t) \cdot (i_1 + \ldots + i_t)$$

$$p_c = \sum_{S_{(i_0,\ldots,i_t)} \in C} p(i_0, \ldots, i_t).$$

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 "The advantage of the hybrid dual attack" section, it is safe to ignore p_s as it is close to 1 for both algorithms. Recall that we have $T_{\mathrm{Hybrid\ 1}} = N \cdot T_{\mathrm{BKZ-h1}} + T_{\mathrm{guess-h1}}$ and $T_{\mathrm{Hybrid\ 2}} = \left(N \cdot T_{\mathrm{BKZ-h2}} + T_{\mathrm{guess-h2}}\right)/p_c$

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 $T_{\mathrm{Hybrid\ 2}}$ and the parameters r, β and L, which is shown in Fig. 4. Note that the influence of r and β on the cost $T_{\mathrm{Hybrid\ 2}}$ 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 "Guess with pruning" section. It's easy to see that increasing r or decreasing L will decrease p_c.

A natural next step is to adjust the parameters r, β, L in HYBRID 2 to get a lower cost $T_{\mathrm{Hybrid\ 2}}$ 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 $T_{\mathrm{BKZ}} = T_{\mathrm{guess}}$. 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 $T_{\mathrm{BKZ}} = T_{\mathrm{guess}}$) 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 "Security estimations" section 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 the prediction of the improvement of Hybrid 2 in Theorem 3. The influence of concentration level on Hybrid 2 could be a useful reference when designing schemes based on LWE with special secret distributions.

Note that concentration level is different from entropy. Surprisingly, a distribution with higher entropy could have a higher concentration level, which means it will be 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 ($1.37 > 1$), 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.

Hybrid 2 can improve the state-of-the-art cryptanalytic result by 27 bits for LAC192. In particular, our estimator show that Hybrid 2 can reduce the bit complexity of LAC192 by 13 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 Fig. 5. Specifically, the first row shows the progression of $T_{\text{Hybrid 2}}$, T_{BKZ}, and p_c as functions of r, and the second row shows the centralization function $g(r, L)$ for the two different secret distributions. For better visualization, in Fig. 5, 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 $0 \leq r \leq 50$, $T_{\text{BKZ}}(r)$ 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 $50 < r \leq 150$, $T_{\text{BKZ}}(r)$ 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, $0 \leq r \leq 9$ 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.

Figure 5 (the second row) visualizes 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.

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 3.

Theorem 3 *Assuming Heuristic 3 and that the optimal parameters of* Hybrid 2 *satisfy* $N = 1$ *and* $T_{\text{BKZ}} = T_{\text{guess}}$, *let* b_1 *the optimal* β *for the dual attack, then the optimal cost of* Hybrid 2 *when guessing* r *entries of the secret* **s** *is*

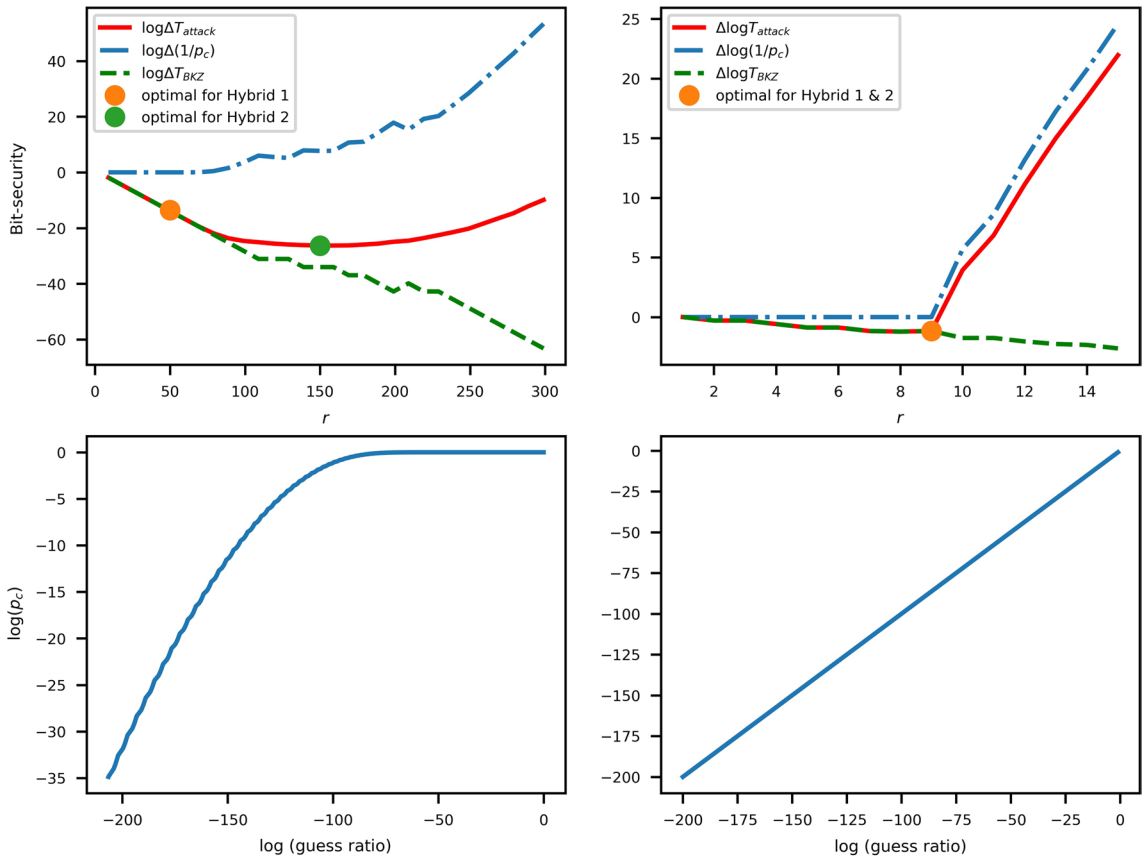

Fig. 5 Comparison between LAC192 (left) and Dilithium768 (right). Figures in the first row plot $T_{\text{Hybrid 2}}$, T_{BKZ}, and p_c in function of r; Figures in the second row visualize the impact of centralization level over p_c

$f(r) = \frac{2^{0.292\cdot b(r)+1}}{g(r,2^{0.0845\cdot b(r)})}$, *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* $\min_{r\geq 0} f(r)$.

Proof According to Heuristic 3 and $T_{\text{guess}} = T_{\text{BKZ}}$, we have that the optimal β and r satisfies that $b(r) = b_1 + \alpha r$, and

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

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

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

□

Remark 2

As an additional sanity check, we show that Theorems 2 and 3 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 2, that is,* $b_2 \approx b_1 \frac{\log R}{\log R - 0.0845\alpha}$.

An additional optimization

In this section we give an efficient algorithm for the matrix multiplication in the guessing stage, which can further decrease T_{guess}. This algorithm can be used for both Hybrid 1 and Hybrid 2 and we refer to the attacks with this additional optimization by Hybrid 1m and Hybrid 2m.

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.

An efficient algorithm from Espitau et al. (2020)

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. Espitau et al. (2020) improves the cost by a factor of r, when the matrix $\mathbf{S}$ has a special form.

Lemma 6 (Espitau et al. 2020) *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 $\{t_1, \ldots, t_\ell\}^r$ in lexicographic order can be calculated in $\mathcal{O}(M \cdot \ell^r)$ time.*

However, Lemma 6 relies on the property that the second matrix $\mathbf{S}$ of size $r \times \ell^r$ consists of all vectors from $\{t_1, \ldots, t_\ell\}^r$. As a result, 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.

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 4 *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.*

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 3
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 I_g for each column of $\mathbf{AS}$. 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.

Next, we show that all the optimal subsets of candidates discussed in "Guess with pruning" section are closed, and hence Theorem 4 can be applied.

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.*

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

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

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 4, the multiplication of $\hat{\mathbf{A}}$ and $\mathbf{S}^*$ can be done in $\mathcal{O}(M \cdot L)$ time. □

Security estimations

We conclude our paper with new estimations for 5 NIST-PQC candidates. Their parameters are given in Table 6 in Appendix A.1. The highlight is presented in Table 2. A full comparison under Assumption 1 for both classical and quantum models is given in Table 11 in Appendix A.2. The improvements under different assumptions discussed in "Lattices and lattice reductions" section are presented in Table 12 in Appendix A.2. 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.

The number of samples allowed from each scheme is shown in Table 6. 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.

Table 5 Parameters of LAC192

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

In addition, we note that for the schemes whose distributions of secret $\mathbf{s}$ and error $\mathbf{e}$ are different, we use the "lattice scaling" technique (Albrecht 2017) (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 and the assumption. Although, we remark that the gain becomes more significant, if we assume a higher complexity of BKZ (i.e., the practical model). Compared with the claimed results (by primal attack), our method reports an overall improvement between 2 to 15 bits under Assumption 1; the actual improvement varies, depending on scheme/parameter sets, as well as the security model. Even under Assumption 2, our method achieves a speedup of up to 7 bits on NTRULPrime. Our algorithm works best on NTRULPrime1277 under the classical core-SVP model, which records an improvement

Table 6 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	1.2	768	see Table 7	1	118	107
	256	3	3329	1	1024		3	182	165
	256	4	3329	1	1280		5	256	232
Saber	256	2	2^{13}	2.29	768	see Table 8	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	4	8380417	$\sqrt{2}$	1280	Uniform in $[-2, 2]$	2	123	112
	256	5	8380417	$\sqrt{20/3}$	1536	Uniform in $[-4, 4]$	3	182	165
	256	7	8380417	$\sqrt{2}$	2048	Uniform in $[-2, 2]$	5	252	229
Frodo	640	–	2^{15}	2.8	640	see Table 9	1	150	137
	976	–	2^{16}	2.3	976		3	215	196
	1344	–	2^{16}	1.4	1344		5	280	255
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	$\#(\pm 1) = 281$	2	176	160
	953	–	6343	$\sqrt{2/3}$	1209	$\#(\pm 1) = 345$	3	197	178
	1013	–	7177	$\sqrt{2/3}$	1269	$\#(\pm 1) = 392$	4	210	190
	1277	–	7879	$\sqrt{2/3}$	1533	$\#(\pm 1) = 429$	5	271	245

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

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

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

Table 7 Kyber's secret distribution

Name	n	k	Probability of			
			0	± 1	± 2	± 3
Kyber512	256	2	$\frac{5}{16}$	$\frac{15}{64}$	$\frac{3}{32}$	$\frac{1}{64}$
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 8 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 9 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						

of 15 bits under Assumption 1 and 7 bits under Assumption 2.

We want to emphasis that, under Assumption 1 the new estimations for Kyber, Saber, 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.

Appendix A: Appendix

Parameters for various cryptosystems

Parameters for various cryptosystems considered in this paper are listed in Table 5, 6, 7, 8 and 9 .

Full results

The full results are shown in Tables 10, 11 and 12.

Additional proofs

Proof of 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 $N \cdot T_{\mathrm{BKZ}} = T_{\mathrm{BKZ}}$ is increasing in β. Then in this case the optimal β minimizing $N \cdot T_{\mathrm{BKZ}} = T_{\mathrm{BKZ}}$ 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 $N \cdot T_{\mathrm{BKZ}}$ defined as follows:

$$\begin{aligned} f(\beta) :&= \frac{M(\beta)}{SV(\beta)} \cdot T_{\mathrm{BKZ}(\beta)} \\ &= \frac{M(\beta)}{2^{0.2075\beta}} \cdot 2^{0.292\beta} \\ &= M(\beta) \cdot 2^{0.0845\beta}. \end{aligned}$$

We will show that $f(\beta)$ is decreasing in β. Then in this case the optimal β minimizing $N \cdot T_{\mathrm{BKZ}}$ 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 approximation $\delta_0 = 2^{\frac{1}{\beta}}$ (Stehlé 2013). Let m_1 and m_2 be the optimal number of equations to use for β and $\beta + 1$ respectively, we have

Table 10 Comparison of Hybrid 1 and the Predictor

Name	Parameters			Dual	Hybrid 1	Predictor
	b_1	R	α			
Kyber512	404	5	−0.98	117	115	115
Kyber768	622	5	−0.96	181	176	177
Kyber1024	870	5	−0.99	253	246	246
Saber512	402	11	−0.96	117	115	116
Saber768	648	9	−0.99	189	185	185
Saber1024	885	7	−1	258	252	252
Dilithium1024	424	13	−0.56	123	122	123
Dilithium1280	623	11	−0.62	181	179	180
Dilithium1792	861	7	−0.59	251	247	248
Frodo640	486	25	−0.99	141	140	140
Frodo976	705	21	−0.89	205	203	203
Frodo1344	927	13	−0.83	270	266	267
NTRULPrime653	447	3	−0.86	130	126	126
NTRULPrime761	532	3	−0.84	155	150	150
NTRULPrime857	605	3	−0.85	176	170	170
NTRULPrime953	671	3	−0.85	195	189	189
NTRULPrime1013	719	3	−0.86	209	202	202
NTRULPrime1277	925	3	−0.86	269	260	260

Table 11 Bit-security estimations under Assumption 1

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	117	114	13	103	102	7	146	140	26	132	128	20
Kyber768	181	175	24	159	157	13	211	201	38	189	183	28
Kyber1024	253	245	34	223	219	19	284	271	49	253	246	34
Saber512	117	114	11	103	102	6	146	141	22	132	129	17
Saber768	189	184	20	166	164	11	219	211	32	196	191	23
Saber1024	258	250	31	227	223	17	288	277	43	257	251	30
Dilithium1024	123	121	14	108	108	6	154	150	26	139	136	20
Dilithium1280	181	179	15	160	159	8	212	208	25	190	188	18
Dilithium1792	251	246	30	221	219	16	282	275	42	252	248	30
Frodo640	141	139	10	124	123	5	171	167	19	154	151	13
Frodo976	205	202	17	181	179	9	236	230	27	211	208	17
Frodo1344	270	264	28	238	235	17	301	292	41	268	263	28
NTRULPrime653	130	125	25	114	112	14	159	149	48	144	137	37
NTRULPrime761	155	148	38	136	133	22	184	172	61	166	158	45
NTRULPrime857	176	168	46	155	151	26	206	192	72	185	176	51
NTRULPrime953	195	187	44	172	168	24	225	211	68	202	193	50
NTRULPrime1013	209	200	45	184	180	25	239	225	67	214	205	48
NTRULPrime1277	269	256	90	237	231	56	300	281	112	268	256	83

Table 12 Bit-security estimations under different cost models and assumptions

Name	Core-SVP Model									Practical Model								
	Assumption 1			Assumption 2			Amortising cost			Assumption 1			Assumption 2			Amortising cost		
	Dual	H2M	Δ	Dual	H2M	Δ	Dual	H2M	Δ	Dual	H2M	Δ	Dual	H2M	Δ	Dual	H2M	Δ
Kyber512	117	114	3	122	119	3	138	138	0	146	140	6	152	145	7	167	164	3
Kyber768	181	175	6	188	182	6	210	210	0	211	201	10	218	208	10	240	236	4
Kyber1024	253	245	8	263	254	9	292	292	0	284	271	13	294	280	14	323	318	5
Saber512	117	114	3	122	119	3	137	137	0	146	141	5	151	146	5	167	164	3
Saber768	189	184	5	196	191	5	219	219	0	219	211	8	226	218	8	249	245	4
Saber1024	258	250	8	268	260	8	297	297	0	288	277	11	298	286	12	328	324	4
Dilithium1024	123	121	2	126	124	2	135	135	0	154	150	4	157	153	4	165	163	2
Dilithium1280	181	179	2	186	183	3	198	198	0	212	208	4	216	212	4	229	227	2
Dilithium1792	251	246	5	257	252	5	272	272	0	282	275	7	288	281	7	304	301	3
Frodo640	141	139	2	147	145	2	164	164	0	171	167	4	177	172	5	194	192	2
Frodo976	205	202	3	212	209	3	233	233	0	236	230	6	243	237	6	264	261	3
Frodo1344	270	264	6	278	272	6	302	302	0	301	292	9	309	300	9	333	330	3
NTRULPrime653	130	125	5	135	129	6	151	150	1	159	149	10	165	154	11	180	174	6
NTRULPrime761	155	148	7	161	153	8	179	177	2	184	172	12	191	177	14	209	199	10
NTRULPrime857	176	168	8	183	174	9	203	200	3	206	192	14	213	198	15	233	222	11
NTRULPrime953	195	187	8	202	193	9	224	221	3	225	211	14	233	218	15	254	245	9
NTRULPrime1013	209	200	9	217	207	10	239	236	3	239	225	14	247	232	15	270	260	10
NTRULPrime1277	269	256	13	279	264	15	306	299	7	300	281	19	309	289	20	337	323	14

*H2M means Hybrid 2m in this table

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

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

$$
g(\beta) := \left(\varepsilon^2(\beta)\right)^{1-2^{-\frac{2}{\beta}}} \cdot 2^{0.0845} \le 1
$$

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

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

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

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

Then it is easy to verify that for any $\beta \ge 50$ and any $\kappa \le 10$,

$$
2^{-0.0075\beta}(\kappa+\ln L) \le 2^6. \tag{4}
$$

Incorporating Eqs. 4–3, we get the upper bound for $\varepsilon^2(\beta)$:

$$
\varepsilon^2(\beta) \le 2^{-0.2\beta+6}. \tag{5}
$$

Incorporating Eq. 5 to $g(\beta)$ we get

$$
g(\beta) \le 2^{(-0.2\beta+6)(1-2^{-\frac{2}{\beta}})+0.0845}.
$$

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

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

This finish the proof for that the optimal β will satisfy $M(\beta) = SV(\beta)$ and $N = 1$. □

Proof of 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

$$m = \sqrt{\frac{(n-r)\log q}{\log \delta_0}} - (n-r)[40]$$

then $\ell = (\delta_0^2)^{\sqrt{\frac{(n-r)\log q}{\log \delta_0}}}$. So the number of samples we need is,[4]

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

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

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

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 $r \geq 2$. 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 Eq. 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 $\beta \geq 150$.

1. For any $r \geq 2$, we have that

$$\begin{aligned}\frac{\kappa + \ln L(r+1)}{\kappa + \ln L(r)} &= \frac{\kappa + (r+1)\ln R}{\kappa + r \ln R} \\ &\leq 1 + \frac{\ln R}{\kappa + r \ln R} \\ &\leq 1 + \frac{1}{r} \\ &\leq \frac{3}{2}\end{aligned} \tag{7}$$

2. According to Lemma 3, the optimal β satisfies $M = \frac{\kappa+\ln L}{\varepsilon^2} = SV = 2^{0.2075\beta}$. As long as $T_{\text{guess}} \leq T_{\text{BKZ}}$, we can upper bound L by that

$$L \leq \frac{T_{\text{guess}}}{M} \leq \frac{T_{\text{BKZ}}}{SV} = 2^{0.0845\beta}.$$

So we can upper bound $\varepsilon^2(r)$ by that

$$\varepsilon^2(r) \leq 2^{-0.2075\beta}(\kappa + \ln L) \leq 2^{-0.2075\beta + \log(10+0.06\beta)} \tag{8}$$

3. According to Assumption 4, we have $\sqrt{\frac{n\log q}{\log \delta_0}} - n \geq \frac{1}{2}n$, so $\sqrt{\frac{n\log q}{\log \delta_0}} \geq \frac{3}{2}n$ and then $\sqrt{\frac{(n-r)\log q}{\log \delta_0}} \geq \frac{3(n-r)}{2}$. In addition, $\sqrt{\frac{n-r-1}{n-r}} - 1 \leq -\frac{1}{2(n-r)}$. Combining these two inequalities, we get

$$\sqrt{\frac{(n-r)\log q}{\log \delta_0}}\left(\sqrt{\frac{n-r-1}{n-r}} - 1\right) \leq -\frac{3}{4}.$$

Then

$$\begin{aligned}1 - X(r)^{\sqrt{\frac{n-r-1}{n-r}}-1} &= 1 - (\delta_0^4)^{\sqrt{\frac{(n-r)\log q}{\log \delta_0}}\left(\sqrt{\frac{n-r-1}{n-r}}-1\right)} \\ &\geq 1 - \delta_0^{-3}.\end{aligned} \tag{9}$$

Note that δ_0 is a function of β.

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

$$\frac{F(r+1)}{F(r)} \leq f(\beta) := \frac{3}{2}\left(\frac{1}{2}\right)^{(1-\delta_0^{-3})(0.2075\beta - \log(10+0.06\beta))}.$$

It is easy to verify that for any $\beta \geq 150$, $f(\beta) < 1$.

□

[4] The formular in Micciancio and Regev (2009) is $\sqrt{\frac{n\log q}{\log \delta_0}}$ since Micciancio and Regev (2009) considers the original dual attack.

Acknowledgements
We would like to thank the anonymous reviewers and editors for detailed comments and useful feedback.

Authors' contributions
BL and LJJ completed the drafted manuscripts of the paper and the scripts of the estimator. LXH and ZZF participated in problem discussions and ZZF completed the final version of the paper. All authors read and approved the final manuscripts.

Funding
This work is supported by National Natural Science Foundation of China (No. 61972391).

Availability of data and materials
Not applicable.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author details
[1]SKLOIS, Institute of Information Engineering, CAS, Beijing, China. [2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China. [3]Nanyang Technological University, Singapore, Singapore. [4]Ethereum Foundation, New York, USA. [5]State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China.

Received: 18 November 2021 Accepted: 10 February 2022
Published online: 01 August 2022

References

Albrecht MR (2017) On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In: EUROCRYPT, vol 10211, pp 103–129
Albrecht MR, Faugère J, Fitzpatrick R, Perret L (2014) Lazy modulus switching for the BKW algorithm on LWE. In: PKC, vol 8383, pp 429–445. https://doi.org/10.1007/978-3-642-54631-0_25
Albrecht MR, Player R, Scott S (2015a) On the concrete hardness of learning with errors. J Math Cryptol 9(3):169–203
Albrecht MR, Cid C, Faugère J, Fitzpatrick R, Perret L (2015b) Algebraic algorithms for LWE problems. ACM Commun Comput Algebra 49(2):62. https://doi.org/10.1145/2815111.2815158
Albrecht MR, Göpfert F, Virdia F, Wunderer T (2017) Revisiting the expected cost of solving uSVP and applications to LWE. In: ASIACRYPT, vol 10624, pp 297–322
Albrecht MR, Curtis BR, Deo A, Davidson A, Player R, Postlethwaite EW, Virdia F, Wunderer T (2018) Estimate all the LWE, NTRU schemes! In: SCN, vol 11035, pp 351–367. https://doi.org/10.1007/978-3-319-98113-0_19
Alkim E, Ducas L, Pöppelmann T, Schwabe P (2016) Post-quantum key exchange-a new hope. In: 25th USENIX security symposium, pp 327–343
Applebaum B, Cash D, Peikert C, Sahai A (2009) Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: CRYPTO, vol 5677, pp 595–618. https://doi.org/10.1007/978-3-642-03356-8_35
Arora S, Ge R (2011) New algorithms for learning in presence of errors. In: ICALP, vol 6755, pp 403–415. https://doi.org/10.1007/978-3-642-22006-7_34
Becker A, Ducas L, Gama N, Laarhoven T (2016) New directions in nearest neighbor searching with applications to lattice sieving. In: SODA, vol 2016, pp 10–24. https://doi.org/10.1137/1.9781611974331.ch2
Bernstein DJ, Chuengsatiansup C, Lange T, van Vredendaal C (2017) NTRU prime: reducing attack surface at low cost. In: SAC, vol 10719. Springer, pp 235–260. https://doi.org/10.1007/978-3-319-72565-9_12
Bootle J, Lyubashevsky V, Seiler G (2019) Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In: CRYPTO, vol 11692, pp 176–202. https://doi.org/10.1007/978-3-030-26948-7_7
Bootle J, Lyubashevsky V, Nguyen NK, Seiler G (2020) A non-PCP approach to succinct quantum-safe zero-knowledge. In: CRYPTO, vol 12171, pp 441–469. https://doi.org/10.1007/978-3-030-56880-1_16
Bos JW, Costello C, Ducas L, Mironov I, Naehrig M, Nikolaenko V, Raghunathan A, Stebila D (2018a) Frodo: take off the ring! practical, quantum-secure key exchange from LWE. In: ACM CCS, pp 1006–1018. https://doi.org/10.1145/2976749.2978425
Bos JW, Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schanck JM, Schwabe P, Seiler G, Stehlé D (2018b) CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM. In: EuroS&P, pp 353–367. https://doi.org/10.1109/EuroSP.2018.00032
Brakerski Z, Gentry C, Vaikuntanathan V (2012) (leveled) fully homomorphic encryption without bootstrapping. In: ITCS, pp 309–325. https://doi.org/10.1145/2090236.2090262
Buchmann JA, Göpfert F, Player R, Wunderer T (2016) On the hardness of LWE with binary error: revisiting the hybrid lattice-reduction and meet-in-the-middle attack. In: AFRICACRYPT, vol 9646, pp 24–43. https://doi.org/10.1007/978-3-319-31517-1_2
Chailloux A, Loyer J (2021) Lattice sieving via quantum random walks. In: ASIACRYPT 2021. Lecture notes in computer science, vol 13093, pp 63–91. https://doi.org/10.1007/978-3-030-92068-5_3
Chen Y (2013) Réduction de réseau et sécurité concrete du chiffrement completement homomorphe. Ph.D. Thesis, Paris 7
Chen Y, Nguyen PQ (2011) Bkz 2.0: Better lattice security estimates. In: ASIACRYPT, vol 7073, pp 1–20
Cheon JH, Hhan M, Hong S, Son Y (2019) A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret LWE. IEEE Access 7:89497–89506. https://doi.org/10.1109/ACCESS.2019.2925425
Chillotti I, Gama N, Georgieva M, Izabachène M (2020) TFHE: fast fully homomorphic encryption over the torus. J Cryptol 33(1):34–91. https://doi.org/10.1007/s00145-019-09319-x
Code for this paper (2021) https://github.com/BiLei121/hybrid-dual-estimator
Dachman-Soled D, Ducas L, Gong H, Rossi M (2020) LWE with side information: attacks and concrete security estimation. In: CRYPTO, vol 12171, pp 329–358. https://doi.org/10.1007/978-3-030-56880-1_12
D'Anvers J, Karmakar A, Roy SS, Vercauteren F (2018) Saber: module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. In: AFRICACRYPT, vol 10831, pp 282–305. https://doi.org/10.1007/978-3-319-89339-6_16
Ducas L (2018) Shortest vector from lattice sieving: a few dimensions for free. In: EUROCRYPT, vol 10820, pp 125–145. https://doi.org/10.1007/978-3-319-78381-9_5
Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schwabe P, Seiler G, Stehlé D (2018) CRYSTALS-Dilithium: a lattice-based digital signature scheme. TCHES 2018(1):238–268. https://doi.org/10.13154/tches.v2018.i1.238-268
Esgin MF, Steinfeld R, Liu JK, Liu D (2019) Lattice-based zero-knowledge proofs: new techniques for shorter and faster constructions and applications. In: CRYPTO, vol 11692, pp 115–146. https://doi.org/10.1007/978-3-030-26948-7_5
Espitau T, Joux A, Kharchenko N (2020) 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, vol 12578, pp 440–462. https://doi.org/10.1007/978-3-030-65277-7_20
Gentry C (2009) Fully homomorphic encryption using ideal lattices. In: STOC, pp 169–178. https://doi.org/10.1145/1536414.1536440
Gentry C, Sahai A, Waters B (2013) Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: CRYPTO, vol 8042, pp 75–92. https://doi.org/10.1007/978-3-642-40041-4_5
Hoffstein J, Pipher J, Silverman JH (1998) NTRU: A ring-based public key cryptosystem. In: ANTS, vol 1423, pp 267–288. https://doi.org/10.1007/BFb0054868
Hoffstein J, Pipher J, Schanck JM, Silverman JH, Whyte W, Zhang Z (2017) Choosing parameters for NTRUEncrypt. In: CT-RSA, vol 10159, pp 3–18. https://doi.org/10.1007/978-3-319-52153-4_1
Howgrave-Graham N (2007) A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In: CRYPTO, vol 4622, pp 150–169
Lindner R, Peikert C (2011) Better key sizes (and attacks) for LWE-based encryption. In: CT-RSA, vol 6558, pp 319–339. https://doi.org/10.1007/978-3-642-19074-2_21

Micciancio D, Regev O (2009). Lattice-based cryptography Springer. https://doi.org/10.1007/978-3-540-88702-7_5
NIST PQC Round 3 submissions (2020) https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions
Regev O (2009) On lattices, learning with errors, random linear codes, and cryptography. JACM 56:1–40
Son Y, Cheon JH (2019) Revisiting the hybrid attack on sparse and ternary secret LWE. In: IACR Cryptol. ePrint Arch., 1019
Stehlé D (2013) An overview of lattice reduction algorithms. Invited talk at ICISC
Wunderer T (2018) On the security of lattice-based cryptography against lattice reduction and hybrid attacks. Ph.D. Thesis, Darmstadt University of Technology, Germany http://tuprints.ulb.tu-darmstadt.de/8082/
Wunderer T (2019) A detailed analysis of the hybrid lattice-reduction and meet-in-the-middle attack. J Math Cryptol 13(1):1–26. https://doi.org/10.1515/jmc-2016-0044

Publisher's Note