Hybrid Dual and Meet-LWE Attack

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

[1] KLOIS, Institute of Information Engineering, CAS, Beijing 100093, China
{bilei,luxianhui,wangkunpeng}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
[3] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[4] Beijing Jiaotong University, Beijing 100044, China
jjluo1@bjtu.edu.cn

Abstract. The Learning with Errors (LWE) problem is one of the most prominent problems in lattice-based cryptography. Many practical LWE-based schemes, including Fully Homomorphic encryption (FHE), use sparse ternary secret for the sake of efficiency. Several (hybrid) attacks have been proposed that benefit from such sparseness, thus researchers believe the security of the schemes with sparse ternary secrets is not well-understood yet. Recently, May [Crypto 2021] proposed an efficient meet-in-the-middle attack named Meet-LWE for LWE with ternary secret, which significantly improves Odlyzko's algorithm. In this work, we generalize May's Meet-LWE and then introduce a new hybrid attack which combines Meet-LWE with lattice dual attack. We implement our algorithm to FHE-type parameters of LWE problem and compare it with the previous hybrid dual attacks. The result shows that our attack outperforms other attacks in a large range of parameters. We note that our attack has no impact on the LWE-based schemes in the PQC Standardization held by NIST as their secrets are not sparse and/or ternary.

Keywords: LWE · Meet-in-the-Middle · Dual Attack · Hybrid Attack.

1 Introduction

For decades, the Learning with Errors (LWE) problem [28] has brought large number of cryptographic applications in lattice-based cryptography, from public-key encryptions [6, 17] and digital signatures [5, 18] to homomorphic encryptions (HE) [22, 29, 15]. Informally, for a fixed secret $\mathbf{s}$ sampled from some fixed distribution over $\mathbb{Z}_q^n$, a set of LWE instances is defined as $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$, where $\mathbf{A}$ is uniformly sampled from $\mathbb{Z}_q^{m\times n}$ and $\mathbf{e}$ is a short error vector sampled from a small discrete Gaussian distribution. The search-version LWE is to recover $\mathbf{s}$ given the instances above and the decision-version LWE asks to distinguish LWE instances from uniform ones.

* Work done while with Nanyang Technological University.

The secret in originally proposed LWE-based schemes is uniform over $\mathbb{Z}_q^n$, while recently many practical constructions diverted the choice of secret distribution for the sake of efficiency. For instance, as one of the most popular implementation of LWE, most HE schemes including HElib [22], SEAL [29] and HEAAN [15] use ternary secret. Even more, the fully HE (FHE) schemes [21, 12, 14, 11, 23] use sparse ternary secrets as it depends on a key technique named bootstrapping which needs the sparsity of the secret. Another well-known lattice-based hard problem NTRU [9] also uses small/sparse[5] secrets.

The concrete security of these LWE-based (NTRU-based) schemes with small/sparse secrets is still not well-understood [1]. Many works [2, 4, 16] show that they are less secure than those with non-small/sparse secrets, but it is still inconclusive whether they are unsafe.

Recently, May [27] introduced a new combinatorial attack, named Meet-LWE, on ternary LWE that significantly improves over Odlyzko's Meet-in-the-Middle (MitM) attack [24]. Compared with Odlyzko's algorithm of runtime $S^{0.5}$, Meet-LWE runs in time roughly $S^{0.25}$, where S is the size of the search-space.

The main open problem proposed by May [27] is whether Meet-LWE can improve lattice hybrid attacks. We remark that the "lattice hybrid attack" in [27] means specifically *hybrid decoding attack* that combines the Nearest Plane (NP) algorithm [7] used in decoding attack and exhaustive-search, which is initialed by Howgrave-Graham [25] against NTRU. From the point of view of attacks against NTRU, hybrid decoding attack is presumably the best-known attack.

For FHE schemes, which are based on LWE with sparse ternary secret, there are two types of hybrid attacks that are usually better than hybrid decoding attack: hybrid primal attack [31, 20, 34, 33, 30] and hybrid dual attack [2, 32, 16, 19, 10]. We note that hybrid primal attack is essentially the same as hybrid decoding attack as it also needs NP algorithm to solve a decoding problem (in a different lattice), while hybrid dual attack is different from them.

Therefore, except for the open problem proposed by May [27], another problem follows from [27] is whether Meet-LWE can be used to improve *hybrid dual attack* for LWE with sparse ternary secret. We study this problem in this paper.

1.1 Related Work

Hybrid dual attack is an efficient attack against LWE, especially LWE with small/sparse secrets. Albrecht [2] introduced the first hybrid dual attack on LWE with small/sparse secret, which is a combination of dual attack and exhaustive-search. Accordingly, the hybrid dual attack consists of two phases, which we name them as the *lattice-phase* and the *guess-phase*, where the first phase uses dual attack to construct a new LWE instance and the second phase uses exhaustive-search to solve the new instance.

[5] In this paper, when we refer to "small", we mean that the secret is binary/ternary and has no fixed Hamming weight, i.e., uniform in $\{0, 1\}$ or $\{0, \pm 1\}$. While for "sparse", we mean that the secret is binary/ternary with a small fixed Hamming weight w.

The works following [2] improve the attack by accelerating the guess-phase. Espitau-Joux-Kharchenko [19] proposed an efficient matrix multiplication method to accelerate exhaustive-search. Bi-Lu-Luo-Wang-Zhang [10] generalized Albrecht's hybrid dual attack to arbitrary secret and error by using both optimal pruning and generalized efficient matrix multiplication. Cheon-Hhan-Hong-Son [16] replaced the exhaustive-search with MitM technique in [25] and showed that the resulting hybrid attack outperforms other attacks for sparse ternary LWE with large modulus. Our attack follows a similar strategy as in [16] but we replace the exhaustive-search with the more efficient algorithm Meet-LWE.

1.2 Contributions

In this paper, we combine Meet-LWE with dual attack and introduce a new hybrid dual attack, which we call hybrid dual Meet-LWE attack. The idea is to replace the exhaustive-search for sub-secret in hybrid dual attack by Meet-LWE. One key step in Meet-LWE is to guess k coordinates of error $\mathbf{e}$ such that we can get k LWE equations without error. These equations will then be used to decrease the size of the candidate set of secrets in the MitM step. The main difficulty in replacing the exhaustive-search by Meet-LWE in *hybrid decoding attacks* (and also hybrid primal attacks) is that we cannot use the k error-free LWE equations on the projected sub-secret anymore [27].

log q \ w	64	128	192	64	128	192	64	128	192	64	128	192
log n	10			11			12			13		
200										103	120	131
160		Ours								117	146	161
120		Hybrid1								136	182	202
100		Hybrid2					102	124	133	147	199	230
80							119	146	162	162	225	276
60				90	106	111	134	179	207			
50				101	126	133	146	202	241			
40	76	83	86	115	153	166	163	234	287			
30	95	110	115	138	193	217						
20	128	161	175									
15	153	205	230									

Fig. 1: Comparison of our attack, Hybrid1 [10], Hybrid2 [16] for different LWE parameters settings $(\log n, \log q, w)$. For each case, the color indicates the best attack and its bit security.

However, this is not a problem for *hybrid dual attacks*, since we can view the lattice-phase of hybrid dual attacks as a dimension-error trade-off, as observed

by Albrecht [2]. More precisely, in hybrid dual attacks, the lattice-phase produces a new LWE instance that has a smaller dimension but a larger error. Our attack solves the new LWE instance by a generalized version of Meet-LWE. One feature of this generalization is that the secret of the new LWE instance follows an atypical ternary distribution while the original Meet-LWE is performed on ternary secret with exactly $w/2$ entries of 1 and $w/2$ entries of -1. In addition, the large error of the new LWE instance makes the analysis of Meet-LWE different from the original setting. We generalize Meet-LWE for the new setting in hybrid dual attacks and give a rigorous analysis for it.

We also compare our attack with previous hybrid dual attacks on sparse ternary LWE problems with FHE-type parameters and find out that our attack outperforms those attacks in a large range of parameters, especially when the Hamming weight of secret is small and the modulus q is not too large. See Fig.1 for an overview of the comparison. The main advantage of our attack is its high efficiency in the guess-phase due to Meet-LWE.

We remark that our result does not invalidate the security claims of the schemes in PQC Standardization held by NIST since their secrets are not sparse/ ternary or they use large enough Hamming weight.

1.3 Roadmap

In Section 2, we give some notations and a brief introduction of lattice reduction algorithms and LWE problem. We recall May's Meet-LWE in Section 3 and recall previous hybrid dual attacks in Section 4. Our new hybrid dual Meet-LWE attack is given in Section 5. In Section 6, we compare the complexity of our algorithm with previous hybrid dual attacks on LWE problem with FHE-type parameters. Finally, in Section 7 we present the conclusion of this paper.

2 Preliminaries

2.1 Notations

Denote log short for $\log_2$ and denote ln for the natural logarithm. Denote vectors in bold, e.g. $\mathbf{v}$. The Euclidean norm of $\mathbf{v}$ is $||\mathbf{v}||$. Denote $\langle \cdot, \cdot \rangle$ the product of two vectors. Matrices are denoted in upper-case bold, e.g. $\mathbf{A}$. Denote $hm(\cdot)$ the Hamming weight of a vector. For a compact set $S \in \mathbb{R}^n$, denote $\mathcal{U}(S)$ the uniform distribution over S. Denote $\mathcal{G}_{\mathbf{c},s}$ the Gaussian distribution of center $\mathbf{c}$ and deviation s, and denote $\mathcal{G}_s$ short for $\mathcal{G}_{\mathbf{0},s}$. Denote the combinatorial number $\binom{M}{N_1}\binom{M-N_1}{N_2}$ as $\binom{M}{N_1,N_2}$.

2.2 Lattice and Lattice Reduction

Lattice. A lattice of *dimension* m is a discrete additive subgroup of $\mathbb{R}^m$ for some $m \in \mathbb{N}$. A basis $\mathbf{B}$ of a lattice Λ is a set of n linearly independent vectors

$\{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^m$ satisfies

$$\Lambda = \Lambda(\mathbf{B}) = \mathbf{B} \cdot \mathbb{Z}^m = \left\{ \sum_{i \in [n]} z_i \cdot \mathbf{b}_i : z_i \in \mathbb{Z} \right\}.$$

We call n the *rank* of the lattice. If $n = m$, Λ is called a *full-rank lattice*. Denote $\det(\Lambda) = \sqrt{\det(\mathbf{B}^T\mathbf{B})}$ the determinant of $\Lambda = \Lambda(\mathbf{B})$. The *shortest vector* of Λ is a non-zero vector in a lattice Λ that has the minimum norm. Denote $\lambda_1(\Lambda)$ the norm of the shortest non-zero vector, i.e., $\lambda_1(\Lambda) = \min_{\mathbf{v} \in \Lambda, \mathbf{v} \neq 0} ||\mathbf{v}||$.

Lattice reduction algorithm. Given a basis of a lattice as input, the lattice reduction algorithm outputs a new basis of the lattice that consists of relatively shorter and relatively pairwise not so skew vectors. The quality of basis outputted by a lattice reduction algorithm is characterized by the *root-Hermite factor* δ_0 which satisfies $\delta_0^m = \frac{||\mathbf{b}_1||}{\det(\Lambda)^{\frac{1}{m}}}$, where $\mathbf{b}_1$ is the first and shortest vector in the output basis.

The BKZ algorithm [13], which is a successful generalization of the famous LLL algorithm, is now a commonly used lattice reduction algorithm. The most important parameter for BKZ is the blocksize β, whose relation with δ_0 is given in the following heuristic.

Heuristic 1 *BKZ with blocksize β yields a basis with 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 by Chen [13].

2.3 The Learning with Errors Problem

Definition 1 (LWE [28]). *Let $n, q \in \mathbb{N}$. $\mathcal{S}$ is the secret distribution over $\mathbb{Z}_q^n$ and χ is a small error distribution over $\mathbb{Z}$. For a secret $\mathbf{s} \leftarrow \mathcal{S}$, denote $LWE_{n,q,\mathbf{s},\chi}$ the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by sampling $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly random, sampling $e \xleftarrow{\$} \chi$ and returning $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. Given access to the outputs from $LWE_{n,q,\mathbf{s},\chi}$, we define two versions of LWE problem:*

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

The LWE instances can be rewrite in matrix form as follows:

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

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

In this paper, we focus on LWE with sparse ternary secrets. We consider three different distributions of sparse ternary $\mathbf{s}$, where the last type of distribution characterizes the error of the new LWE instance in the guess-phase of hybrid dual attacks.

- Ternary-0 : $\mathcal{T}_0^n(w) = \left\{\mathbf{s} \in \{0, \pm 1\}^n : \mathbf{s} \text{ has } \frac{w}{2}(\pm 1)\text{-entries each}\right\}$
- Ternary-1 : $\mathcal{T}_1^n(w) = \{\mathbf{s} \in \{0, \pm 1\}^n : \mathbf{s} \text{ has } w \text{ non-zero entries}\}$
- Ternary-2 : $\mathcal{T}_2^n(w) = \sum_{h=0}^{w} p_s(h)\mathcal{T}_1^n(h)$, where $\sum_{h=0}^{w} p_s(h) = 1$, i.e., $\mathcal{T}_2^n(w)$ is a mixture distribution of $\mathcal{T}_1^n(h)$ with weight $p_s(h)$ for $h \leq w$.

2.4 Lemma

Lemma 1 ([8]). *For any real $s > 0$ and $C > 0$, and any $\mathbf{x} \in \mathbb{R}^d$, we have*

$$\Pr\left[|\langle \mathbf{x}, \mathcal{G}_s \rangle| \geq C \cdot s||\mathbf{x}||\right] < 2 \cdot \exp\left(-\frac{C^2}{2}\right).$$

3 May's Meet-LWE Attack

In this section, we review May's Meet-LWE [27] attack on LWE with $\mathbf{s} \in \mathcal{T}_0^n(w)$ and $\mathbf{e} \in \{0, \pm 1\}^m$, and show that it can be straightforwardly generalized to the case with $\mathbf{s} \in \mathcal{T}_1^n(w)$.

3.1 Ternary-0

We recall May's Meet-LWE in its simplest form (Rep-0 in [27]). Given LWE instance $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e} \bmod q) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, where $\mathbf{s} \in \mathcal{T}_0^n(w)$ and $\mathbf{e} \in \{0, \pm 1\}^m$, a typical MitM works by splitting the secret into $\mathbf{s} = \mathbf{s}_1 + \mathbf{s}_2$, rewriting the LWE equation as

$$\mathbf{As}_1 = \mathbf{b} - \mathbf{As}_2 - \mathbf{e} \bmod q,$$

and hashing $\mathbf{As}_1$ and $\mathbf{b} - \mathbf{As}_2$ for all enumerated $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{T}_0^n(w/2)$. Then for each pair of $\mathbf{s}_1$ and $\mathbf{s}_2$ with colliding hash values, we check whether

$$\mathbf{b} - \mathbf{A}(\mathbf{s}_1 + \mathbf{s}_2) \bmod q \in \{0 \pm 1\}^m.$$

In order to reduce the number of doing hash, which is the main runtime of the process, Meet-LWE chooses only a subset of $\mathcal{T}_0^n(w/2)$ for $\mathbf{s}_1$ and $\mathbf{s}_2$ as follows.

Notice that the number of *representations* $\mathbf{s} = \mathbf{s}_1 + \mathbf{s}_2$ is $R = \binom{w/2}{w/4}^2$. We define the mapping

$$\pi_k^m : \mathbb{Z}_q^m \to \mathbb{Z}_q^k, \mathbf{x} = (x_1, \cdots, x_m) \to (x_1, \cdots, x_k)$$

and fix a random target $\mathbf{t} \in \mathbb{Z}_q^k$ and then look for $\mathbf{s}_1$ and $\mathbf{s}_2$ satisfying

$$\pi_k^m(\mathbf{As}_1 + \mathbf{e}_1) = \mathbf{t} \bmod q \text{ and } \pi_k^m(\mathbf{b} - \mathbf{As}_2 + \mathbf{e}_2) = \mathbf{t} \bmod q, \tag{1}$$

Algorithm 1 Meet-LWE on LWE with Ternary-0 Secret

Require: $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e} \bmod q) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ with $\mathbf{s} \in \mathcal{T}_0^n(w)$ and $\mathbf{e} \in \{0, \pm 1\}^m$
Ensure: $\mathbf{s} \in \mathcal{T}_0^n(w)$ satisfying $\mathbf{As} - \mathbf{b} \bmod q \in \{0, \pm 1\}^m$
1: compute the number R of representations of $\mathbf{s} = \mathbf{s}_1 + \mathbf{s}_2$ where $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{T}_0^n(w/2)$
2: compute $k = \lfloor \log_q(R) \rfloor$
3: sample a random $\mathbf{t} \in \mathbb{Z}_q^k$
4: **for** all $\pi_k^m(\mathbf{e}_1) \in \{0, \pm 1\}^{k/2} \times 0^{k/2}$ **do**
5: construct $L_1^{(1)} = \{(\mathbf{s}_1, \varphi(\mathbf{As}_1)) : \pi_k^m(\mathbf{As}_1 + \mathbf{e}_1) = \mathbf{t} \bmod q\}$ via a standard MitM on $\mathbf{u}_1 \in \mathcal{T}_0^{n/2}(w/4) \times 0^{n/2}$ and $\mathbf{u}_2 \in 0^{n/2} \times \mathcal{T}_0^{n/2}(w/4)$.
6: **for** all $\pi_k^m(\mathbf{e}_2) \in 0^{k/2} \times \{0, \pm 1\}^{k/2}$ **do**
7: construct $L_2^{(1)} = \{(\mathbf{s}_2, \varphi(\mathbf{b} - \mathbf{As}_2)) : \pi_k^m(\mathbf{b} - \mathbf{As}_2 + \mathbf{e}_2) = \mathbf{t} \bmod q\}$ analogously
8: **for** all matched of $(\mathbf{s}_1, \cdot)$ and $(\mathbf{s}_2, \cdot)$ in the second component of $L_1^{(1)}$ and $L_2^{(2)}$ **do**
9: **if** $\mathbf{s} = \mathbf{s}_1 + \mathbf{s}_2 \in \{\pm 1, 0\}^n$ has weight w and $\mathbf{As} - \mathbf{b} \bmod q \in \{0, \pm 1\}^m$ **then**
10: **return** $\mathbf{s}$

where $\mathbf{e}_1, \mathbf{e}_2 \in \{0, \pm 1\}^m$ satisfies $\mathbf{e}_1 - \mathbf{e}_2 = \mathbf{e}$. To ensure that there is at least one couple of $\mathbf{s}_1$ and $\mathbf{s}_2$ satisfying Eq.(1), we choose k such that $k = \lfloor \log_q R \rfloor$ and therefore we have $q^k \leq R$. Note that the probability that Eq.(1) holds for at least one representation of $\mathbf{s}_1 + \mathbf{s}_2$ is $p_\pi = \left(1 - \frac{1}{q^k}\right)^R \approx \frac{1}{e}$. In order to find such $\mathbf{s}_1$ and $\mathbf{s}_2$, we make up two lists

$$L_1^{(1)} = \{(\mathbf{s}_1, \varphi(\mathbf{As}_1)) : \pi_k^m(\mathbf{As}_1 + \mathbf{e}_1) = \mathbf{t} \bmod q\},$$
$$L_2^{(1)} = \{(\mathbf{s}_2, \varphi(\mathbf{b} - \mathbf{As}_2)) : \pi_k^m(\mathbf{b} - \mathbf{As}_2 + \mathbf{e}_2) = \mathbf{t} \bmod q\},$$

where the hash function $\varphi : \mathbb{Z}_q^m \to \{0, 1\}^m$ is defined as

$$\varphi(\mathbf{x})_i = \begin{cases} 0 & \text{if } x_i \in \left[-\frac{q}{2}, -1\right) \\ 1 & \text{if } x_i \in \left[0, \frac{q}{2} - 1\right) \\ 0, 1 & \text{if } x_i \in [-1, 0) \cup \left[\frac{q}{2} - 1, \frac{q}{2}\right) \end{cases}.$$

Notice that for entries in the two border ranges $[-1, 0)$ and $\left[\frac{q}{2} - 1, \frac{q}{2}\right)$, we assign both 0 and 1 to them. The lists $L_1^{(1)}, L_2^{(1)}$ can be constructed in a standard MitM manner, i.e., enumerate $\mathbf{s}_1$ as the sum of

$$\mathbf{u}_1 \in \mathcal{T}_0^{n/2}(w/4) \times 0^{n/2} \text{ and } \mathbf{u}_2 \in 0^{n/2} \times \mathcal{T}_0^{n/2}(w/4).$$

Analogously, we proceed with $\mathbf{s}_2 = \mathbf{u}_3 + \mathbf{u}_4$.

To summarize, we first compute a number k based on the number of representations R. Next for each enumeration of the first k coordinates of $\mathbf{e}$ (via some standard MitM approach as $\mathbf{e} = \mathbf{e}_1 + \mathbf{e}_2$), we construct lists $L_1^{(1)}$ and $L_2^{(1)}$ and then search for a representation $\mathbf{s}_1 + \mathbf{s}_2$ of $\mathbf{s}$ based on the second component of $L_1^{(1)}$ and $L_2^{(1)}$. The full algorithm is listed in Algorithm 1.

Analysis. The size of lists $L_1^{(1)}$ and $L_2^{(1)}$ is

$$L^{(1)} = \frac{S^{(1)}}{q^k} \approx \frac{S^{(1)}}{R} = \binom{n}{w/4, w/4}\binom{w/2}{w/4}^{-2},$$

where $S^{(1)} = \binom{n}{w/4,w/4}$ is the search-space of $\mathbf{s}_1$ and $\mathbf{s}_2$. Notice that $L^{(1)}$ is much smaller than $S^{(1)}$ and this is the main advantage of Meet-LWE. We remark that here we only count one in $L^{(1)}$ for each element in the lists $L_1^{(1)}$ and $L_2^{(1)}$, and omit possible multiple labels for elements, since the expected number of labels for each element is $\frac{2}{q} \cdot m = \Theta(1)$. However, in Section 5 when we study our hybrid attack we cannot omit this as the error becomes much larger. We will discuss this in more detail in Section 5 . The size of the four lists for $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ is

$$L^{(2)} = S^{(2)} = \binom{n/2}{w/8, w/8},$$

where $S^{(2)}$ is the search-space of $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$.

The time $T^{(1)}$ to construct list $L_1^{(1)}$ (respectively $L_2^{(1)}$) is

$$T^{(1)} = \max\left\{L^{(1)}, L^{(2)}\right\}.$$

Finding a representation $\mathbf{s}_1 + \mathbf{s}_2$ from $L_1^{(1)}$ and $L_2^{(1)}$ can be realized via Odlyzko's hash function on the $m-k$ coordinates in time

$$T^{(0)} = \max\left\{L^{(1)}, 2^{-(m-k)}\left(L^{(1)}\right)^2\right\} = L^{(1)}.$$

Here we assume that $L^{(1)} \leq 2^{m-k}$, otherwise we can modify Odlyzko's hash function by assigning more than two labels to ensure this.

Then the time complexity of list construction is

$$T_s = \max\{T^{(1)}, T^{(0)}\} = \max\{L^{(1)}, L^{(2)}\}.$$

In addition, the time of enumerating

$$\pi_k^m(\mathbf{e}_1) \in \{0, \pm 1\}^{k/2} \times 0^{k/2} \text{ and } \pi_k^m(\mathbf{e}_2) \in 0^{k/2} \times \{0, \pm 1\}^{k/2}$$

is $T_e = 3^{k/2}$. We summarize these results in lemma 2.

Lemma 2. *The runtime of Meet-LWE attack on LWE with Ternary-0 secret shown in Algorithm 1 is computed as*

$$T_{MitM\text{-}0} = T_s \cdot T_e = \max\left\{\binom{n}{w/4, w/4}\binom{w/2}{w/4}^{-2}, \binom{n/2}{w/8, w/8}\right\} \cdot 3^{k/2},$$

and the success probability is $p_{MitM\text{-}0} = p_\pi = \frac{1}{e}$.

3.2 Ternary-1

Recall that $\mathcal{T}_1^n(w)$ contains $\mathbf{s} \in \{0, \pm 1\}^n$ with w non-zero entries. This type of secret is very similar to Ternary-0. Given LWE instance with secret $\mathbf{s} \in \mathcal{T}_1^n(w)$, we can split $\mathbf{s}$ into $\mathbf{s}_1 + \mathbf{s}_2$ with $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{T}_1^n(w/2)$. The number of representations is $R = \binom{w}{w/2}$. The two levels of lists is constructed similarly as Section 3.1. Accordingly, we can compute the size of the lists as

$$L^{(1)} = \frac{S^{(1)}}{R} = \binom{n}{w/2} \cdot 2^{w/2} / \binom{w}{w/2}$$
$$L^{(2)} = S^{(2)} = \binom{n/2}{w/4} \cdot 2^{w/4}.$$

The total runtime is

$$T_{\text{MitM-1}} = T_s \cdot T_e = \max\left\{ \binom{n}{w/2} \cdot 2^{w/2} / \binom{w}{w/2}, \binom{n/2}{w/4} \cdot 2^{w/4} \right\} \cdot 3^{k/2},$$

where $k = \lfloor \log_q R \rfloor$, and the success probability is also $p_{\text{MitM-1}} = p_\pi = \frac{1}{e}$.

4 Hybrid Dual Attacks

In this section, we review previous hybrid dual attacks [2, 10, 16, 19]. Hybrid dual attacks have two phases: the lattice-phase and the guess-phase. The lattice-phase is the same for all hybrid dual attacks and we can view it as a dimension-error trade-off, i.e., after the first phase, we get a new decision-LWE instance with a smaller dimension but a larger error. In the guess-phase, there are two different approaches to solve the new instance. A detailed description follows.

Lattice-phase. In order to distinguish whether the given instance $(\mathbf{A}, \mathbf{b})$ is sampled from $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$ or $\text{LWE}_{n,q,\mathbf{s},\mathcal{G}_\sigma}$ with $\mathbf{s} \in \mathcal{T}_1^n(w)$, we divide $\mathbf{A}$ into two parts:

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

Accordingly, we also divide $\mathbf{s}$ into two parts: $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2) \in \{0, \pm 1\}^r \times \{0, \pm 1\}^{n-r}$. In this phase, the attack constructs the dual lattice over $\mathbf{A}_2$:

$$\Lambda(\mathbf{A}_2) = \left\{ (\mathbf{w}, \mathbf{v}) \in \mathbb{Z}^m \times \left(\frac{1}{c} \cdot \mathbb{Z}\right)^{n-r} : \mathbf{w} \cdot \mathbf{A}_2 = c \cdot \mathbf{v} \bmod q \right\}$$

with scale factor $c = \sigma \cdot \sqrt{\frac{m}{w_{n-r}}}$, where w_{n-r} is the expected hamming weight of $\mathbf{s}_2$. If the given instance follows $\text{LWE}_{n,q,\mathbf{s},\mathcal{G}_\sigma}$, by obtaining short vector $(\mathbf{w}, \mathbf{v})$ from $\Lambda(\mathbf{A}_2)$, we compute $\langle \mathbf{w}, \mathbf{b} \rangle \bmod q$ as

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

This can be viewed as a new LWE instance $\left(\bar{\mathbf{a}}, \bar{b} = \langle \bar{\mathbf{a}}, \mathbf{s}_1 \rangle + \bar{e} \bmod q\right)$, where

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

Denote M the number of short vectors $(\mathbf{w}, \mathbf{v}) \in \Lambda(\mathbf{A}_2)$. We write Eq.(2) in the matrix form as $\bar{\mathbf{b}} = \bar{\mathbf{A}}\mathbf{s}_1 + \bar{\mathbf{e}} \bmod q$, where $\bar{\mathbf{b}}, \bar{\mathbf{e}} \in \mathbb{Z}_q^M$ and $\bar{\mathbf{A}} \in \mathbb{Z}_q^{M \times r}$. This instance follows distribution $\mathrm{LWE}_{r,q,\mathbf{s}_1,\mathcal{G}_\rho}$ with $\rho = \ell\sigma$ where $\ell = ||(\mathbf{w}, \mathbf{v})||$ [6]. If the given instance is from $\mathcal{U}\left(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m\right)$, then the new instance $\left(\bar{\mathbf{A}}, \bar{\mathbf{b}}\right)$ is also uniform over $\mathbb{Z}_q^{M \times r} \times \mathbb{Z}_q^M$. So next we are going to solve this new decision-LWE instance in the second phase.

Guess-phase. The difference between different hybrid dual attacks is the method of solving the new instance in this phase.

The first method works by checking the distribution of $\bar{\mathbf{b}} - \bar{\mathbf{A}}\tilde{\mathbf{s}}_1 \bmod q$, where $\tilde{\mathbf{s}}_1$ is some guessed candidate of $\mathbf{s}_1$. By enumerating $\tilde{\mathbf{s}}_1$ in some way, we can compute $\bar{\mathbf{b}} - \bar{\mathbf{A}}\tilde{\mathbf{s}}_1 \bmod q$. It equals to a Gaussian error $\bar{\mathbf{e}}$ if $(\bar{\mathbf{A}}, \bar{\mathbf{b}}) \sim \mathrm{LWE}_{r,q,\mathbf{s}_1,\mathcal{G}_\rho}$, otherwise it is uniform over $\mathbb{Z}_q^M$. We can compute the statistical distance to distinguish $\mathrm{LWE}_{n,q,\mathbf{s}_1,\mathcal{G}_\rho}$ from $\mathcal{U}\left(\mathbb{Z}_q^{M \times r} \times \mathbb{Z}_q^M\right)$. This method is used in the first hybrid dual attack [2] and also in [10, 19]. Note that [19] defined the distance by themselves instead of using statistical distance while the results are similar.

The second method is to check whether all entries of $\bar{\mathbf{b}} - \bar{\mathbf{A}}\tilde{\mathbf{s}}_1 \bmod q$ are in some range $[-B, B]$ for a chosen B. If this holds for one enumerated $\tilde{\mathbf{s}}_1$, we decide the original instance is from $\mathrm{LWE}_{n,q,\mathbf{s},\mathcal{G}_\sigma}$. The hybrid dual attack in [16] uses this method and it additionally accelerates the guessing of $\mathbf{s}_1$ by a MitM approach. Notice that compared with the first method, the second method has a stricter requirement on the error size of the new LWE instance, and thus shorter vectors from the dual lattice are required in the lattice-phase.

5 Combine Meet-LWE with Dual Attack

We are now ready to present our hybrid dual Meet-LWE attack. The idea is to replace the exhaustive-search in the guess-phase of hybrid dual attacks by Meet-LWE. That is, in the guess-phase, we use generalized Meet-LWE to solve the new instance $\left(\bar{\mathbf{A}}, \bar{\mathbf{b}}\right) \in \mathbb{Z}_q^{M \times r} \times \mathbb{Z}_q^M$.

There are two problems we need to overcome when applying Meet-LWE to the new setting. The first one is that the secret of the new LWE instance has a different distribution, which will influence the choices of k and the analysis of success probability. The second one is that the error of the new LWE instance becomes large. For this we need to re-analyze the runtime of Meet-LWE as some constant omitted in the original setting with ternary error now becomes too large to be omitted. We solve these two problems in Section 5.1 and Section 5.2 respectively, and then present the complete algorithm and analysis in Section 5.3.

5.1 Meet-LWE on Ternary-2 LWE

We first consider the secret distribution of the new instance

$$\left(\bar{\mathbf{A}}, \bar{\mathbf{b}} = \bar{\mathbf{A}}\mathbf{s}_1 + \bar{\mathbf{e}} \bmod q\right) \in \mathbb{Z}_q^{M\times r} \times \mathbb{Z}_q^M.$$

Note that in this subsection, we follow May [27] to set the error ternary and defer the discussion of the Gaussian error to the next subsection.

As the secret $\mathbf{s}_1 \in \{0, \pm 1\}^r$ is part of the original secret $\mathbf{s}$, we have $hm(\mathbf{s}_1) \leq w_r \coloneqq \min(r, w)$. Thus $\mathbf{s}_1 \in \mathcal{T}_2^r(w_r)$ and for each $h \leq w_r$ the probability $p_s(h)$ for $\mathbf{s}_1$ to have weight h is

$$p_s(h) = \frac{\binom{w}{h}\binom{n-w}{r-h}}{\binom{n}{r}}.$$

To apply Meet-LWE attack to this type of secret, we first need to choose a weight parameter $\hat{w} \leq w_r$ and use $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{T}_1^r(\hat{w}/2)$ to form $\mathbf{s}_1$. Then the search-space of $\mathbf{u}_1, \mathbf{u}_2$ is

$$S^{(1)} = \binom{r}{\hat{w}/2} \cdot 2^{\hat{w}/2}.$$

Notice that for a fixed parameter $\hat{w}$, we can only form the secrets in

$$\bigcup_{h=0}^{\hat{w}/2} \mathcal{T}_1^r(2h),$$

and hence the success probability is at most $\sum_{h=0}^{\hat{w}/2} p_s(2h)$.

The next step is to identify the dimension k of the random target $\mathbf{t} \in \mathbb{Z}_q^k$. Recall that in Section 3.1 and Section 3.2, we just set $k = \lfloor \log_q R \rfloor$ based on the number of representations R. However, for $\mathcal{T}_2^r(w_r)$ we cannot identify k directly as we have different number of representations for different cases of $\mathbf{s}_1$ with different weights. More precisely, for each $h \leq \hat{w}/2$, the number of representations of $\mathbf{u}_1 + \mathbf{u}_2$ for $\mathbf{s}_1$ with $hm(\mathbf{s}_1) = 2h$ is

$$R(h) = \binom{2h}{h}\binom{r-2h}{\hat{w}/2-h} \cdot 2^{\hat{w}/2-h}.$$

Notice that since $hm(\mathbf{s}_1) = 2h$ and $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{T}_1^r(\hat{w}/2)$, there are $\hat{w}/2 - h$ non-zero entries of $\mathbf{u}_1$ and $\hat{w}/2 - h$ non-zero entries of $\mathbf{u}_2$ that cancel each other out among the $r - 2h$ 0-entries of $\mathbf{s}_1$, and for each cancel out entry, we have two possibilities $1 + (-1) = 0$ or $(-1) + 1 = 0$. This gives us $\binom{r-2h}{\hat{w}/2-h} \cdot 2^{\hat{w}/2-h}$. For the $2h$ non-zero entries of $\mathbf{s}_1$, there are $\hat{w}/2 - (\hat{w}/2 - h) = h$ entries from $\mathbf{u}_1$ and $\mathbf{u}_2$ respectively, which gives us $\binom{2h}{h}$.

For each $R(h)$, let $k(h) = \lfloor \log_q(R(h)) \rfloor$. Thus we need to choose

$$k \in \left[\min_h(k(h)), \max_h(k(h))\right].$$

Algorithm 2 Generalized Meet-LWE on LWE with Ternary-2 Secret

Require: $\left(\bar{\mathbf{A}}, \bar{\mathbf{b}}\right) \in \mathbb{Z}_q^{M \times r} \times \mathbb{Z}_q^M, \hat{w}$
Ensure: $\mathbf{s}_1 \in \mathcal{T}_2^r(\hat{w})$ satisfying $\bar{\mathbf{A}}\mathbf{s}_1 - \bar{\mathbf{b}} \bmod q \in \{0, \pm 1\}^M$ or $\perp$
1: **for** each $h \in [0, \hat{w}/2]$ **do**
2: compute the number $R(h)$ of representations of $\mathbf{s}_1 = \mathbf{u}_1 + \mathbf{u}_2$ where $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{T}_1^r(\hat{w}/2)$
3: compute $k(h) = \lfloor \log_q(R) \rfloor$
4: choose a $k \in [\min_h(k(h), \max_h(k(h))]$ (we will brute-force all possible values for k and choose the optimal one in Section 6)
5: sample a random $\mathbf{t} \in \mathbb{Z}_q^k$
6: **for** all $\pi_k^M(\mathbf{e}_1) \in \{0, \pm 1\}^{k/2} \times 0^{k/2}$ **do**
7: construct $L_1^{(1)} = \left\{\left(\mathbf{u}_1, \varphi\left(\bar{\mathbf{A}}\mathbf{u}_1\right)\right) : \pi_k^M\left(\bar{\mathbf{A}}\mathbf{u}_1 + \mathbf{e}_1\right) = \mathbf{t} \bmod q\right\}$ via a standard MitM
8: **for** all $\pi_k^M(\mathbf{e}_2) \in 0^{k/2} \times \{0, \pm 1\}^{k/2}$ **do**
9: construct $L_2^{(1)} = \left\{\left(\mathbf{u}_2, \varphi\left(\bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{u}_2\right)\right) : \pi_k^M\left(\bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{u}_2 + \mathbf{e}_2\right) = \mathbf{t} \bmod q\right\}$ analogously
10: **for** all matched of $(\mathbf{u}_1, \cdot)$ and $(\mathbf{u}_2, \cdot)$ in the second component of $L_1^{(1)}$ and $L_2^{(1)}$ **do**
11: **if** $\mathbf{s}_1 = \mathbf{u}_1 + \mathbf{u}_2 \in \mathcal{T}_2^r(\hat{w})$ and $\bar{\mathbf{A}}\mathbf{s}_1 - \bar{\mathbf{b}} \bmod q \in \{0, \pm 1\}^M$ **then**
12: **return** $\mathbf{s}_1$
13: **return** $\perp$

For a fixed k, if $hm(\mathbf{s}_1) = 2h$, then

$$\pi_k^M\left(\bar{\mathbf{A}}\mathbf{u}_1 + \mathbf{e}_1\right) = \mathbf{t} \bmod q \text{ and } \pi_k^M\left(\bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{u}_2 + \mathbf{e}_2\right) = \mathbf{t} \bmod q$$

holds with probability $p_\pi(h) = 1 - \left(1 - \frac{1}{q^k}\right)^{R(h)}$, where $\bar{\mathbf{e}} = \mathbf{e}_1 - \mathbf{e}_2$. Then overall success probability is

$$\sum_{h=0}^{\hat{w}/2} p_s(2h) \cdot p_\pi(h) = \sum_{h=0}^{\hat{w}/2} p_s(2h) \cdot \left(1 - \left(1 - \frac{1}{q^k}\right)^{R(h)}\right). \tag{3}$$

The remaining part of the algorithm is the same as before. We give the pseudo-code of the generalized Meet-LWE on LWE with Ternary-2 Secret in Algorithm 2.

Analysis. The runtime analysis is similarly as before. The sizes of the lists are $L^{(1)} = \frac{S^{(1)}}{q^k}$ and $L^{(2)} = S^{(2)} = \binom{r/2}{\hat{w}/4} \cdot 2^{\hat{w}/4}$. The time $T^{(1)}$ to construct list $L_1^{(1)}$, respectively $L_2^{(1)}$, is $T^{(1)} = \max\left\{L^{(1)}, L^{(2)}\right\}$, and the time $T^{(0)}$ of approximately matching on the $M - k$ coordinates via Odlyzko's hash function is $T^{(0)} = \max\left\{L^{(1)}, 2^{-(M-k)}\left(L^{(1)}\right)^2\right\} = L^{(1)}$. The time of list construction is

$$T_s = \max\left\{T^{(1)}, T^{(0)}\right\} = \max\left\{L^{(1)}, L^{(2)}\right\}.$$

The time of enumerating $\pi_k^M(\mathbf{e}_1)$ and $\pi_k^M(\mathbf{e}_2)$ is $T_e = 3^{k/2}$.

Combining the runtime and the success probability given in Eq.(3), we conclude with the following lemma.

Lemma 3. *The runtime of Meet-LWE algorithm in Algorithm 2 is*

$$T_{MitM\text{-}2} = T_s \cdot T_e = \max\left\{ \binom{r}{\hat{w}/2} \cdot 2^{\hat{w}/2}/q^k, \binom{r/2}{\hat{w}/4} \cdot 2^{\hat{w}/4} \right\} \cdot 3^{k/2},$$

and the success probability is $p_{MitM\text{-}2} = \sum_{h=0}^{\hat{w}/2} p_s(2h) \cdot p_\pi(h)$.

5.2 The larger error

When performing Meet-LWE in the guess-phase of hybrid dual attack, the error of the new LWE instance is Gaussian instead of in $\{0, \pm 1\}^M$. In this case, we need to reconsider the runtime of the attack. We first choose a boundary B to cover the new error with a high probability[6] as in the following lemma, which can be proved by using Lemma 1.

Lemma 4. *Error* $\bar{e}$ *of the new LWE instance given in Eq.(2) satisfies*

$$\Pr\left[|\bar{e}| < B\right] \geq 1 - 2 \cdot \exp(-4\pi),$$

where $B = (2\sqrt{2\pi} + 1) \cdot \sqrt{\frac{m}{m+n-r}} \cdot \ell\sigma$ *and* ℓ *is the length of* $(\mathbf{w}, \mathbf{v}) \in \Lambda(\mathbf{A}_2)$. *Thus, we have that*

$$p_M \coloneqq \Pr\left[\bar{\mathbf{e}} \in [-B, B]^M\right] \geq (1 - 2 \cdot \exp(-4\pi))^M.$$

Now we have to enumerate $\pi_k^M(\mathbf{e}_1)$ and $\pi_k^M(\mathbf{e}_2)$ in a larger range $[-B, B]$ using time $T_e = (2B+1)^{k/2}$.

Note that for the estimation in Section 6 we usually have $k = 1$ as q and B are large. In this case we can still use MitM for $\pi_1^M(\mathbf{e}_1)$ and $\pi_1^M(\mathbf{e}_2)$ in one dimension to get time T_e. We note that this case is not considered in [27] as the parameter k in [27] is large, which is different from ours.

Specifically, now we can use

$$\pi_1^M(\mathbf{e}_1) \in [0, c) \text{ and } \pi_1^M(\mathbf{e}_2) \in \{c \cdot i - B \mid i \in [0, c)\}$$

to form $\pi_1^M(\bar{\mathbf{e}}) \in [-B, B]$, where $c = \left\lceil \sqrt{2B+1} \right\rfloor$. For example, to enumerate $e \in [-40, 40]$ we can split it into $e = e_1 + e_2$ by taking $e_1 \in [0, 9)$ and $e_2 \in \{9 \cdot i - 40 \mid i \in [0, 9)\}$ where $9 = \left\lceil \sqrt{40 \times 2 + 1} \right\rfloor$. This method can also be used to deal with the situation when k is odd.

[6] In Lemma 4, we follow [16] to choose the value of B such that the probability for $|\bar{e}| < B$ is close to 1. Our experimental results show that the overall attack complexity is not sensitive on B and the current choice of B in Lemma 4 is almost optimal.

The second difference is that when constructing lists

$$L_1^{(1)} = \left\{ \left(\mathbf{u}_1, \varphi\left(\bar{\mathbf{A}}\mathbf{u}_1\right)\right) : \pi_k^M\left(\bar{\mathbf{A}}\mathbf{u}_1 + \mathbf{e}_1\right) = \mathbf{t} \bmod q \right\},$$
$$L_2^{(1)} = \left\{ \left(\mathbf{u}_2, \varphi\left(\bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{u}_2\right)\right) : \pi_k^M\left(\bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{u}_2 + \mathbf{e}_2\right) = \mathbf{t} \bmod q \right\},$$

we will use a different hash function $\varphi : \mathbb{Z}_q^M \to \{0,1\}^M$ defined as

$$\varphi(\mathbf{x})_i = \begin{cases} 0 & \text{if } x_i \in \left[-\frac{q}{2}, -B\right) \\ 1 & \text{if } x_i \in \left[0, \frac{q}{2} - B\right) \\ 0,1 & \text{if } x_i \in [-B, 0) \cup \left[\frac{q}{2} - B, \frac{q}{2}\right) \end{cases}.$$

Recall that in Section 3.1, when we compute the size $L^{(1)}$ of lists $L_1^{(1)}$ and $L_2^{(1)}$, we count each element once in $L^{(1)}$ as each element has only a constant number of labels in expectation. However, this does not hold for the current setting, since now the expected number of labels for each element is $\frac{2B+1}{q}M$, which is not small anymore if B is large.

To figure out this difference, we introduce a new notation $\overline{L^{(1)}}$ to represent the overall number of labels for all elements in lists $L_1^{(1)}$ and $L_2^{(1)}$, and we still use $L^{(1)}$ to represent the number of elements in the lists. For a given M, we have

$$\overline{L^{(1)}} = L^{(1)} \cdot 2^{\frac{2B+1}{q}M}.$$

Since $\overline{L^{(1)}}$ will influence the runtime, we need to be careful when choosing the dimension M for the new LWE instance to optimize the runtime of Meet-LWE.

5.3 Our attack

Now we are ready to give our attack. For given guessing dimension r, blocksize β and weight parameter $\hat{w} \leq w_r$, the pseudo-code of our attack is shown in Algorithm 3. Line 1-4 is the lattice-phase of our attack, which is the same as other hybrid dual attacks. After this phase, we get a new instance $\left(\bar{\mathbf{A}}, \bar{\mathbf{b}}\right) \in \mathbb{Z}_q^{M \times r} \times \mathbb{Z}_q^M$ and solve it by using Algorithm 4. Then according to the output of Algorithm 4, Algorithm 3 outputs the result of the decision-LWE problem.

Note that Algorithm 4 is essentially the same as Algorithm 2, except that in Algorithm 4, the scope of exhaustive-searching $\pi_k^M(\mathbf{e}_1), \pi_k^M(\mathbf{e}_2)$ in line 6, 8 and the final judgment condition in line 11 are both changed to adapt to the situation of large error in hybrid dual attack.

Analysis. We represent $\mathbf{s}_1 = \mathbf{u}_1 + \mathbf{u}_2$ with $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{T}_1^r(\hat{w}/2)$. The sizes of the two level lists are

$$L^{(1)} = \frac{S^{(1)}}{q^k} = \binom{r}{\hat{w}/2} \cdot 2^{\hat{w}/2} \cdot \frac{1}{q^k},$$
$$L^{(2)} = S^{(2)} = \binom{r/2}{\hat{w}/4} \cdot 2^{\hat{w}/4}.$$

Algorithm 3 Hybrid Dual Meet-LWE Attack

Require: $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m, r, \beta, \hat{w}, \sigma$
Ensure: LWE or Uniform
1: divide $\mathbf{A}$ into two parts $(\mathbf{A}_1, \mathbf{A}_2) \in \mathbb{Z}_q^{m\times r} \times \mathbb{Z}_q^{m\times(n-r)}$
2: construct lattice $\Lambda(\mathbf{A}_2) = \left\{(\mathbf{w}, \mathbf{v}) \in \mathbb{Z}^m \times \mathbb{Z}^{n-r} : \mathbf{w} \cdot \mathbf{A}_2 = \mathbf{v} \bmod q\right\}$
3: perform BKZ algorithm with blocksize β on $\Lambda(\mathbf{A}_2)$ to obtain M short vectors $(\mathbf{w}, \mathbf{v})$ of length ℓ
4: construct new instance $\left(\bar{\mathbf{A}}, \bar{\mathbf{b}}\right) \in \mathbb{Z}_q^{M\times r} \times \mathbb{Z}_q^M$ by computing each row/entry of $\bar{\mathbf{A}}$ and $\bar{\mathbf{b}}$ as $\bar{\mathbf{a}} = \mathbf{w}\mathbf{A}_1 \bmod q$ and $\bar{b} = \langle \mathbf{w}, \mathbf{b}\rangle \bmod q$
5: set B as Lemma 4
6: run Algorithm 4 on input $\left(\bar{\mathbf{A}}, \bar{\mathbf{b}}\right), B$ and $\hat{w}$
7: **if** Algorithm 4 outputs a secret vector **then**
8: **return** LWE
9: **else**
10: **return** Uniform

Then the time $T^{(1)}$ to construct list $L_1^{(1)}$ (respectively $L_2^{(1)}$) is computed as

$$T^{(1)} = \max\left\{L^{(1)}, L^{(2)}\right\}.$$

Finding a representation $\mathbf{u}_1 + \mathbf{u}_2$ from $L_1^{(1)}$ and $L_2^{(1)}$ can be realized via Odlyzko's hash function on the $M-k$ coordinates in time

$$T^{(0)} = \max\left\{\overline{L^{(1)}}, 2^{-(M-k)}\left(\overline{L^{(1)}}\right)^2\right\} \tag{4}$$

where $\overline{L^{(1)}} = L^{(1)} \cdot 2^{\frac{2B+1}{q}M}$. Then the time of list construction is

$$T_s = \max\left\{T^{(1)}, T^{(0)}\right\}. \tag{5}$$

And the time of enumerating $\pi_k^M(\mathbf{e}_1)$ and $\pi_k^M(\mathbf{e}_2)$ is

$$T_e = (2B+1)^{k/2}. \tag{6}$$

Combining Eq.(5), Eq.(6) and Lemma 4, we get the following theorem.

Theorem 1. *The runtime of our hybrid dual Meet-LWE attack in Algorithm 3 is*

$$T_{DUAL\text{-}MEET} = T_{reduction} + T_{meet},$$

where $T_{reduction} = T_{BKZ}(\beta)$, $T_{meet} = T_s \cdot T_e$ *and* T_s, T_e *are defined as Eq.(5), Eq.(6) respectively. The success probability of the attack is*

$$p_{DUAL\text{-}MEET} = p_{MitM\text{-}2} \cdot p_M,$$

where $p_{MitM\text{-}2} = \sum_{h=0}^{\hat{w}/2} p_s(2h) \cdot p_\pi(h)$, *and* $p_M = (1 - 2\cdot\exp(-4\pi))^M$.

Algorithm 4 Generalized Meet-LWE on LWE with Ternary-2 Secret and large error

Require: $\left(\bar{\mathbf{A}}, \bar{\mathbf{b}}\right) \in \mathbb{Z}_q^{M\times r} \times \mathbb{Z}_q^M, B, \hat{w}$
Ensure: $\mathbf{s}_1 \in \mathcal{T}_2^r(\hat{w})$ satisfying $\bar{\mathbf{A}}\mathbf{s}_1 - \bar{\mathbf{b}} \bmod q \in \{0, \cdots, \pm B\}^M$ or $\perp$
1: **for** each $h \in [0, \hat{w}/2]$ **do**
2: compute the number $R(h)$ of representations of $\mathbf{s}_1 = \mathbf{u}_1 + \mathbf{u}_2$ where $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{T}_1^r(\hat{w}/2)$
3: compute $k(h) = \lfloor \log_q(R) \rfloor$
4: choose a $k \in [\min_h(k(h)), \max_h(k(h))]$ (we will choose the optimal value of k to optimize the complexity in Section 6)
5: sample a random $\mathbf{t} \in \mathbb{Z}_q^k$
6: **for** all $\pi_k^M(\mathbf{e}_1) \in \{0, \cdots, \pm B\}^{k/2} \times 0^{k/2}$ **do**
7: construct $L_1^{(1)} = \left\{\left(\mathbf{u}_1, \varphi\left(\bar{\mathbf{A}}\mathbf{u}_1\right)\right) : \pi_k^M\left(\bar{\mathbf{A}}\mathbf{u}_1 + \mathbf{e}_1\right) = \mathbf{t} \bmod q\right\}$ via a standard MitM
8: **for** all $\pi_k^M(\mathbf{e}_2) \in 0^{k/2} \times \{0, \cdots, \pm B\}^{k/2}$ **do**
9: construct $L_2^{(1)} = \left\{\left(\mathbf{u}_2, \varphi\left(\bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{u}_2\right)\right) : \pi_k^M\left(\bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{u}_2 + \mathbf{e}_2\right) = \mathbf{t} \bmod q\right\}$ analogously
10: **for** all matched of $(\mathbf{u}_1, \cdot)$ and $(\mathbf{u}_2, \cdot)$ in the second component of $L_1^{(1)}$ and $L_2^{(1)}$ **do**
11: **if** $\mathbf{s}_1 = \mathbf{u}_1 + \mathbf{u}_2 \in \mathcal{T}_2^r(\hat{w})$ and $\bar{\mathbf{A}}\mathbf{s}_1 - \bar{\mathbf{b}} \bmod q \in \{0, \cdots, \pm B\}^M$ **then**
12: **return** $\mathbf{s}_1$
13: **return** $\perp$

Remark 1. There is one parameter M left to be determined, which is chosen to balance the two components of $T^{(0)}$ in Eq.(4). Accordingly, we set

$$M = \frac{kq}{2B+1+q} \cdot \log L^{(1)}$$

to ensure that $\overline{L^{(1)}} \approx 2^{-(M-k)}\left(\overline{L^{(1)}}\right)^2$.

6 Complexity Estimation and Comparison

In this section, we present a detailed comparison of our attack with the other two hybrid dual attacks in [10] and [16] [7] (we refer to them as Hybrid1 and Hybrid2 respectively) by estimating the bit-security of various parameter settings of sparse ternary LWE.

Our estimators take LWE parameters as input and find optimal parameters for the attack to get the optimal (lowest) bit-security. The estimation of bit-security is computed as $\log T_{\text{attack}} - \log p_{\text{attack}}$ [4]. The concrete formulas for our attack are given in Theorem 1.

The runtime of each hybrid attack T_{attack} consists of two parts: $T_{\text{reduction}}$ and T_{guess}, where $T_{\text{reduction}}$ is the time of lattice reduction, and T_{guess} corresponds to the guess-phase for searching the correct sub-secret in dimension r (which is

[7] Notice that [10] and [16] are the representatives of existing two different categories of hybrid dual attacks and [10] improves the attack in [2] with additional tricks.

Table 1: $\log n = 12$, $\log q = 50$, $w = 128$, $\sigma = 3.2$

Attack		Dual	Hybrid1	Hybrid2	OURS
Cost (bit)	attack	302	215	221	202
	reduction	302	166	183	172
	guess	-	166	181	170=147+23
	prob.	-	49	38	30
Parameter	r	-	1839	2247	2245
	β	925	460	520	483
	k	-	-	-	1

Table 2: $\log n = 10$, $\log q = 20$, $w = 192$, $\sigma = 3.2$

Attack		Dual	Hybrid1	Hybrid2	OURS
Cost (bit)	attack	188	175	285	220
	reduction	188	169	267	204
	guess	-	169	267	203=156+47
	prob.	-	5	19	16
Parameter	r	-	161	357	432
	b	539	475	810	595
	k	-	-	-	5

denoted as T_{meet} in our attack as we use MitM technique to accelerate guessing). Under the optimal parameters we usually have $T_{\text{reduction}} \approx T_{\text{guess}}$. The main parameter to balance $T_{\text{reduction}}$ and T_{guess} is the dimension r. Since we focus on sparse ternary LWE problems, in the guess-phase we usually only cover part of the search-space, which incurs a loss in p_{attack} but reduces T_{guess}. The final estimation is a trade-off between the three components: $T_{\text{reduction}}$, T_{guess}, and p_{attack}. Note that in this paper we assume that $T_{\text{BKZ}}(d, \beta) = 8d \cdot 2^{0.292\beta+16.4}$, where d is the dimension of the lattice and β is the blocksize of BKZ, and use the amortizing model [2] for BKZ performed in dual attack.

We perform the attacks on LWE with FHE-type parameters. Before presenting the complete picture of the comparison, we first analyze 3 typical cases in detail to get a close look into the inner parts of the attacks.

6.1 Case 1

We begin with a case for which our attack works the best. We set $\log n = 12$, $\log q = 50$, $w = 128$, $\sigma = 3.2$. The results for the standalone dual at-

Table 3: $\log n = 13$, $\log q = 200$, $w = 128$, $\sigma = 3.2$

Attack		Dual	HYBRID1	HYBRID2	OURS
Cost (bit)	attack	140	124	120	206
	reduction	140	112	112	198
	guess	-	110	112	199=120+79
	prob.	-	12	9	7
Parameter	r	-	1650	2050	1637
	b	365	270	269	563
	k	-	-	-	1

tack, HYBRID1, HYBRID2, and our attack are shown in Table 1. In addition to $\log T_{\text{reduction}}$, $\log T_{\text{guess}}$, and $-\log p_{\text{attack}}$, we also give the guessing dimension r and blocksize β for each attack. For our attack, we additionally give the enumeration dimension k for the error and we split T_{guess} into T_s and T_e.

All three hybrid attacks achieve lower complexity than the standalone dual attack due to the sparse ternary secret. Our attack achieves the lowest complexity due to its high efficiency in the guessing. Compared with HYBRID1, our attack guesses in a larger dimension (2245 vs 1839) in a slightly longer time (170 vs 166) but achieves a much higher success probability (30 vs 49). Compared with HYBRID2, our attack guesses in a similar dimension with a shorter time (170 vs 181) and achieves a higher success probability (30 vs 38).

Notice that the time of guessing T_{guess} for our attack is close to HYBRID1 and shorter than HYBRID2 even if the time $T_e = 2^{23}$ for enumerating $\mathbf{e}_1, \mathbf{e}_2$ is included. Recall that reference [27] deals with schemes with ternary secrets and the time for enumeration is $T_e = 3^{k/2}$. For us, the new LWE instance after the lattice-phase has a large error range B, which could make $T_e = (2B + 1)^{k/2}$ very large. At first glance, it may look strange that here our T_e is still so small. However, notice that in our case q is large enough for the number of representations R such that we just need to fix a random target $\mathbf{t}$ in one dimension, i.e., $k = 1$, thus $T_e = (2B + 1)^{0.5}$ is not too large.

6.2 Case 2

Next, we look at a different case with a larger weight ratio $\frac{w}{n}$, where HYBRID1 works the best. We choose $\log n = 10$, $\log q = 20$, $w = 192$, $\sigma = 3.2$. The results are shown in Table 2. Different from the first case, now HYBRID1 achieves the lowest complexity. The main reason for the bad performance of our attack is the larger $T_e = 2^{47}$ due to the larger weight ratio $\frac{w}{n} \approx 0.177$. Recall that for case 1 we have $\frac{w}{n} \approx 0.031$. The large weight ratio results in a larger number R of representations, which increases the dimension k for the random target $\mathbf{t}$ and then increases T_e.

It may look weird that our attack and Hybrid2, which use the MitM technique, are even worse than the standalone dual attack. This is due to the fundamental difference between the two different categories of hybrid dual attacks discussed in Section 4. For dual attack and Hybrid1, they need to find short vectors in the lattice-phase such that in the guess-phase the distribution of the new error with range B can be differentiated from the uniform distribution. While for our attack and Hybrid2, we have to guarantee a smaller $\frac{B}{q}$ such that we can recognize the correct solution by checking each entry of $\bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{s}_1 \bmod q$. Therefore, we have to find shorter vectors in the lattice-phase, which makes $T_{\text{reduction}}$ large, especially when $\frac{w}{n}$ is large.

6.3 Case 3

We consider the last case with a very large $q = 2^{200}$, with which Hybrid2 works best. We set $\log n = 13$, $\log q = 200$, $w = 128$, and $\sigma = 3.2$. The results are shown in Table 3. We can see that Hybrid1 and Hybrid2 have similar complexity that are smaller than dual attack, while our attack has a much larger complexity than all of them since we have a very large $T_e = 2^{79}$. Since $k = 1$, the main reason for the large $T_e = (2B+1)^{k/2}$ is that when q is large, the range B of the error after the lattice reduction also becomes large. On the other hand, Hybrid1 and Hybrid2 are mainly influenced by the relative value of $\frac{B}{q}$ instead of the absolute value of B. Notice that in this case with large q and small $\frac{w}{n}$, Hybrid2 outperforms Hybrid1 while in the first two cases Hybrid2 cannot compete with Hybrid1.

6.4 Overview

To summarize, our attack outperforms Hybrid1 and Hybrid2 when the weight ratio $\frac{w}{n}$ is small and q is not too large. When the ratio $\frac{w}{n}$ is large, our attack and Hybrid2 are both worse than Hybrid1, sometimes even worse than dual attack. When q is very large, our attack suffers from the large T_e, and Hybrid2 achieves the best performance if the ratio $\frac{w}{n}$ is small enough.

To give an overview of the different advantages of the three hybrid dual attacks, we consider a series of sparse ternary LWE problems with FHE-type parameters. For each $\log n = 10, 11, 12, 13$, we choose appropriate q such the corresponding scheme with ternary secret has bit-security around 128 to 256. For each considered case of n and q, we consider three different values of $w = 64, 128, 192$ and fix $\sigma = 8/\sqrt{2\pi} \approx 3.2$.

The comparison results are shown in Fig.1. For each case we give the estimation result of the best attack together with a color indicating the best attack for this case. The figure can be roughly partitioned into three regions, corresponding to the three cases considered above. Our attack is the best for most cases when $\log n = 12$. For $\log n = 10, 11$, as the weight ratio $\frac{w}{n}$ becomes larger, Hybrid1 is the best for most cases and our attack is the best for cases with small weight (e.g., all cases for $w = 64$ and $\log n = 11$). When $\log n = 13$, the corresponding

values of q become large. In this case, Hybrid2 becomes the best attack for most cases while our attack is the best for cases with smaller q.

Based on Fig.1, some FHE implementations (e.g., HElib [22] and HEAAN [15]) with parameters that fall within the advantage area of our attack should re-estimate their parameters. Our results do not make any impact on the schemes in Round 3 of Post-Quantum Cryptography Standardization held by NIST since for the LWE-based schemes, they do not adopt sparse ternary secret terms (except for NTRULPrime, however for these schemes, Hybrid1 works better), and for NTRU-based schemes, dual attacks cannot be applied to estimate the bit-security of them [3].

Remark 2. We do not include dual attack as Hybrid1 always works no worse than dual attack [10]. In addition, we also compare these three attacks with the primal attack and the comparison shows that the hybrid dual attacks work better than the primal attack in most cases. Due to the space limitation, we do not give the specific comparison results here.

7 Conclusion

In this work, we introduce and analyze a new hybrid dual attack named hybrid dual Meet-LWE attack, which combines dual attack and a generalization of Meet-LWE attack [27]. We compare our attack with previous hybrid dual attacks on LWE with FHE-type parameters. The result shows that our attack outperforms those attacks in a large range of parameters. According to our results, some FHE implementations should update their parameters.

For future works, we note that the main drawback of our attack is the additional time of guessing k coordinates of the errors, which increases with q. Recently, [26] introduced a locality sensitive hashing (LSH) technique that avoids the guessing of the errors in Meet-LWE. It is interesting to study whether this technique can improve the performance of our attack.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (No. 61972391) and the Open Project Program of State Key Laboratory of Cryptology (MMKFKT201810).

References

1. Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A., Vaikuntanathan, V.: Homomorphic encryption standard (2018)
2. Albrecht, M.R.: On dual lattice attacks against small-secret LWE and parameter choices in helib and SEAL. In: EUROCRYPT 2017. vol. 10211, pp. 103–129 (2017)
3. Albrecht, M.R., Curtis, B.R., Deo, A., Davidson, A., Player, R., Postlethwaite, E.W., Virdia, F., Wunderer, T.: Estimate all the {LWE, NTRU} schemes! In: SCN 2018. vol. 11035, pp. 351–367 (2018)

4. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Math. Cryptol. **9**(3), 169–203 (2015)
5. Alkim, E., Barreto, P.S.L.M., Bindel, N., Krämer, J., Longa, P., Ricardini, J.E.: The lattice-based digital signature scheme qtesla. In: ACNS (2020)
6. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - A new hope. In: 25th USENIX. pp. 327–343 (2016)
7. Babai, L.: On lovász'lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986)
8. Banaszczyk, W.: Inequalities for convex bodies and polar reciprocal lattices in r^n II: application of k-convexity. Discret. Comput. Geom. (1996)
9. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: NTRU prime: Reducing attack surface at low cost. In: SAC 2017. vol. 10719, pp. 235–260 (2017)
10. Bi, L., Lu, X., Luo, J., Wang, K., Zhang, Z.: Hybrid dual attack on LWE with arbitrary secrets. IACR Cryptol. ePrint Arch. **2021**, 152 (2021)
11. Chen, H., Chillotti, I., Song, Y.: Improved bootstrapping for approximate homomorphic encryption. In: EUROCRYPT 2019. vol. 11477, pp. 34–54 (2019)
12. Chen, H., Han, K.: Homomorphic lower digits removal and improved FHE bootstrapping. In: EUROCRYPT 2018. vol. 10820, pp. 315–337 (2018)
13. Chen, Y.: Réduction de réseau et sécurité concrete du chiffrement completement homomorphe. Ph.D. thesis, Paris 7 (2013)
14. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: Bootstrapping for approximate homomorphic encryption. In: EUROCRYPT 2018. vol. 10820, pp. 360–384. Springer (2018)
15. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: Snucrypto HEAAN (2019), https://github.com/homenc/HElib
16. Cheon, J.H., Hhan, M., Hong, S., Son, Y.: A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret LWE. IEEE Access (2019)
17. Cheon, J.H., Kim, D., Lee, J., Song, Y.: Lizard: Cut off the tail! A practical post-quantum public-key encryption from LWE and LWR. In: SCN 2018. vol. 11035, pp. 160–177 (2018)
18. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2018**(1), 238–268 (2018)
19. Espitau, T., Joux, A., Kharchenko, N.: On a dual/hybrid approach to small secret LWE - A dual/enumeration technique for learning with errors and application to security estimates of FHE schemes. In: INDOCRYPT 2020. vol. 12578, pp. 440–462 (2020)
20. Göpfert, F., van Vredendaal, C., Wunderer, T.: A hybrid lattice basis reduction and quantum search attack on LWE. In: PQCrypto (2017)
21. Halevi, S., Shoup, V.: Bootstrapping for helib. In: EUROCRYPT 2015. vol. 9056, pp. 641–670 (2015)
22. Halevi, S., Shoup, V.: (2019), https://github.com/homenc/HElib
23. Han, K., Ki, D.: Better bootstrapping for approximate homomorphic encryption. In: CT-RSA 2020. vol. 12006, pp. 364–390 (2020)
24. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: ANTS (1998)
25. Howgrave-Graham, N.: A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In: CRYPTO 2007. vol. 4622, pp. 150–169 (2007)
26. Kirshanova, E., May, A.: How to find ternary lwe keys using locality sensitive hashing. In: IMA International Conference on Cryptography and Coding. pp. 247–264. Springer (2021)

27. May, A.: How to meet ternary LWE keys. In: CRYPTO (2021)
28. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM) (2009)
29. Microsoft SEAL: (2019), https://github.com/Microsoft/SEAL
30. Son, Y., Cheon, J.H.: Revisiting the hybrid attack on sparse secret LWE and application to HE parameters. In: WAHC@CCS 2019. pp. 11–20 (2019)
31. Wunderer, T.: Revisiting the hybrid attack: Improved analysis and refined security estimates. IACR Cryptol. ePrint Arch. **2016**, 733 (2016)
32. Wunderer, T.: On the Security of Lattice-Based Cryptography Against Lattice Reduction and Hybrid Attacks. Ph.D. thesis, Darmstadt University of Technology, Germany (2018)
33. Wunderer, T.: A detailed analysis of the hybrid lattice-reduction and meet-in-the-middle attack. J. Math. Cryptol. **13**(1), 1–26 (2019)
34. Wunderer, T., Burger, M., Nguyen, G.N.: Parallelizing the hybrid lattice-reduction and meet-in-the-middle attack. In: CSE 2018. pp. 185–193 (2018)