# A Hybrid of Dual and Meet-in-the-Middle Attack on Sparse and Ternary Secret LWE

Jung Hee Cheon[1], Minki Hhan[1], Seungwan Hong[1], and Yongha Son[1]

Seoul National University (SNU), Republic of Korea

**Abstract.** The dual attack is one of the most efficient attack algorithms for the Learning with Errors (LWE) problem. Recently, an efficient variant of the dual attack for sparse and small secret LWE was reported by Albrecht [Eurocrypt 2017], which forces some LWE-based cryptosystems, especially fully homomorphic encryptions (FHE), to change parameters. In this work, we propose a new hybrid of dual and meet-in-the-middle (MITM) attack, which outperforms the improved variant on the same LWE parameter regime. To this end, we adapt the MITM attack for NTRU due to Odlyzko to LWE, and give a rigorous analysis for it. The performance of our MITM attack depends on the relative size of error and modulus, and hence for a large modulus LWE samples, our MITM attack works well for quite large error. We then combine our MITM attack with Albrecht's observation that understands the dual attack as dimension-error tradeoff, which finally yields our hybrid attack. We also implement a sage module that estimates the attack complexity of our algorithm upon LWE-estimator, and our attack shows significant performance improvement for the LWE parameter for FHE. For example, for the LWE problem with dimension $n = 2^{15}$, modulus $q = 2^{628}$ and ternary secret key with Hamming weight 64 which is one parameter set used for HEAAN bootstrapping [Eurocrypt 2018], our attack takes $2^{112.5}$ operations and $2^{70.6}$ bit memory while the previous best attack requires $2^{127.2}$ operations as reported by LWE-estimator.

**Keywords:** Cryptanalysis, Fully homomorphic encryption, Learning with Errors, Meet-in-the-middle.

## 1 Introduction

The Learning with Errors (LWE) problem has brought many fruitful applications in the cryptographic world [Reg05,Pei09,ADPS16,CKLS18,HS14,BCD+16, LP11]. The strongest advantage of the LWE problem for cryptosystems is that the LWE problem is provably difficult to solve [Reg05, BLP+13, Pei09]. In other words, the LWE problem is as intractable as known hard problems of lattices, even in the average cases in the certain parameter regime. Thanks to this property, the LWE problem plays the important role in the cryptography, especially for homomorphic encryptions (HE) [BVG12, CKKS17, CGGI16, DM15, GSW13, BV14, FV12]. HE is an encryption scheme that allows computations over encrypted data including additions and multiplications. Since HE enables the operation without knowledge of the message information at all, the need of HE has been boosted with the necessity of entrusting industrial object, such as outsourced computation and privacy-preserving neural networks, without disclosing personal data.

In accordance with this growing attention on HE, some practical variants of LWE problem has been studied to boost the efficiency of HE. First, since the norm of secret vector deeply affects the performance of HE, most of the HE implementations including `HElib`, `SEAL` and `HEAAN` use the *ternary*[1] secret vectors [HS13, LP16, Kim18]. However, those variants of LWE using such small secret vectors currently lie outside of the currently known provable secure parameter regime. That is, the security guarantee for HE implementations is somewhat weaker than the original LWE problem. Moreover, to support infinite numbers of operations, in other words, to have *fully* homomorphic encryptions, one needs to perform a technique named *bootstrapping*. Since the performance of bootstrapping depends on the Hamming weight of secret vector, aforementioned HE implementations further uses *sparse*[2] ternary secret vectors in practice [HS15, CH18, CHK$^+$18].

In this situation, Albrecht [Alb17] recently pointed out that the security of LWE with (sparse) ternary secret key is far weaker than previous thoughts by suggesting a new variant of the *dual attack*, which is one of primary solving algorithms for LWE. This attack is covered by a program that gives the estimated bit-security level of queried LWE parameters called LWE-estimator [APS15], and indeed shows the best performance for LWE parameters used for HE; large modulus and the sparse ternary secret vector.

### 1.1 The Dual Attack

Informally, LWE asks one to determine, given a matrix $A \in \mathbb{Z}_q^{m\times n}$ chosen uniformly at random and a vector $\boldsymbol{b} \in \mathbb{Z}_q^n$, whether $\boldsymbol{b}$ is also chosen uniformly at random or of the form $\boldsymbol{b} = A\boldsymbol{s} + \boldsymbol{e}$ for some vector $\boldsymbol{s}$ and small error vector $\boldsymbol{e}$. The *dual attack* strategy finds a short vector $\boldsymbol{v}$ that is orthogonal to matrix $A$. Then, by computing $\langle \boldsymbol{v}, \boldsymbol{b}\rangle$, one can guess whether $(A, \boldsymbol{b})$ is an LWE sample; if $\boldsymbol{b} = A\boldsymbol{s} + \boldsymbol{e}$, one has $\langle \boldsymbol{v}, \boldsymbol{b}\rangle = \langle \boldsymbol{v}, \boldsymbol{e}\rangle \mod q$, which would be still small if $\boldsymbol{v}$ is sufficiently short.

If the secret vector $\boldsymbol{s}$ is sparse, then the columns of $A$ that correspond to the zero components of $\boldsymbol{s}$ have no influence on $\boldsymbol{b} = A\boldsymbol{s} + \boldsymbol{e}$. Now one can apply the dual attack by choosing some random portion of columns of $A$, and this strategy also works if all the other columns correspond to zero component part of $\boldsymbol{s}$. This strategy naturally drops the attack success probability by the guessing success probability, but since the dimension is reduced, it takes shorter time for finding a short vector, which enables one to choose the optimal point where the total attack complexity is minimized.

Albrecht [Alb17] further observed here, although some columns are wrongly guessed, one can compensate it by a brute-force method: Let $A'$ be the matrix

[1] all entries are in $\{-1, 0, 1\}$

[2] the Hamming weight is small

consisting of the ignored columns of $A$ in the above strategy and $\boldsymbol{s}'$ be the part of secret key corresponding to $A'$. Then, one has

$$\langle \boldsymbol{v}, \boldsymbol{b} \rangle = \boldsymbol{v}A'\boldsymbol{s}' + \langle \boldsymbol{v}, \boldsymbol{e} \rangle \mod q,$$

which can be understood as a new LWE sample $(\boldsymbol{v}A', \boldsymbol{v}A'\boldsymbol{s}' + \boldsymbol{e}')$ with $\boldsymbol{e}' = \langle \boldsymbol{v}, \boldsymbol{e} \rangle$. In this regard, the dual attack strategy can be considered as a *dimension error trade-off*. Now, by exhaustively searching possible $\boldsymbol{s}'$ to some extent, one can succeed to have $\langle \boldsymbol{v}, \boldsymbol{e} \rangle$ even if some guess are incorrect, which increases the attack success probability with proper amount of exhaustive search.

### 1.2 Our Contributions

In this paper, upon the current dual attack framework, we apply MITM attacks for LWE instead of exhaustive search. For that, we first observe that Odlyzko's MITM attack on NTRU [HGHSW03] can be easily adapted to the literature of LWE, and we give an explicit algorithm and rigorous analysis for it. The cost of this attack is proportional to the square root of the number of candidate secret vector, while it is less sensitive to the absolute size of error when the ratio of error and modulus is sufficiently small. Thus, this MITM attack is highly appropriate for the trade-offed LWE sample for the large modulus case and from this observation,

From this observation, we propose a new hybrid attack of the dual attack and MITM attack. Our hybrid attack shows significant performance improvement on the sparse ternary secret LWE problems, which are used in two homomorphic encryptions `HElib` [HS14] and `HEAAN` [CKKS17][3]. We estimate our attack complexity for several parameters that are in the currently used parameter range for the HEs. This result shows that our attack can solve the sparse ternary secret LWE problems in more than 1000 times faster compared to the previous attacks on average.

### 1.3 Related Works and Discussions

**Other Hybrid Attacks** There is another *hybrid* of lattice reduction and MITM attack proposed by Howgrave-Graham [HG07], which attacked another primary lattice based problem named NTRU. After then, some series of works have adapted this hybrid attack into LWE problems [BGPW16, Wun16]. These hybrid approaches use lattice reduction to solve closest vector problem (CVP) with Babai's nearest plane algorithm [Bab86], and this is quite different from our usage of lattice reduction. Since these attacks apply MITM strategy on error vector (not secret vector), they only improve for extremely small and non-standard error distribution; such as binary or ternary error.

One may consider modifying the Howgrave-Graham's hybrid attack so that it can be applied for small secrets (not small error), and combining it with

[3] `SEAL` also needs to use the sparse ternary key to support the bootstrapping. See [CH18].

the dimension error trade-off. However, since the success probability of Babai's nearest plane drops doubly exponentially along with dimension of LWE samples, it would be highly inefficient for our interest parameter regime that has quite large dimension (and modulus size). For this reason, we only adapt the original MITM approach of Odlyzko for the following step of dimension-error trade-off.

**Impact on NIST standardizations** Our attack has no improvement on small parameters used in the public key cryptosystem, especially on the recent post-quantum cryptography; the dimension error trade-off phase may increase the error bound $B$ so that $B/q \gtrsim 2^{-10}$. In this case, the MITM takes too much time, which makes our hybrid attack only quite ineffective.

### 1.4 Roadmap

In Section 2, we give some preliminaries of this paper and explain related works in detail. In Section 3, we review the details of the several variants of dual attack. In Section 4, we show LWE can be attacked in MITM approach, by giving an efficient method to perform noisy search. After then in Section 5, we see lattice reduction algorithms can be used to trade-off dimension and error of LWE, which leads to a new hybrid attack for LWE. In Section 6, we discuss some results and consequences of our attack.

## 2 Preliminaries

**Notations.** We write $\mathbb{Z}_q$ by the set $\mathbb{Z}/q\mathbb{Z}$ whose elements are represented in $(-q/2, q/2] \cap \mathbb{Z}$. Every vector will be denoted by small bold letters, and matrix will be denoted by capital letters. We denote the Euclidean norm of vectors by $\|\cdot\|$, and the maximum norm is distinguished by $\|\cdot\|_\infty$. For a set $S$, we denote the uniform distribution over $S$ by $\mathcal{U}(S)$.

A 0-centered distribution $\mathcal{D}$ over $\mathbb{Z}$ is said to be $(B, \varepsilon)$-bounded if the probability $\Pr[|\mathcal{D}| \geq B]$ is less than $\varepsilon$. We denote a 0-centered discrete Gaussian distribution with width parameter $s \in \mathbb{R}$ by $\mathcal{D}_{\mathbb{Z},s}$. The following lemma says, for any $\boldsymbol{x} \in \mathbb{R}^n$, the distribution $\langle \boldsymbol{x}, \mathcal{D}_{\mathbb{Z}^n,s} \rangle$ is $(C \cdot s\|\boldsymbol{x}\|, 2 \cdot \exp(-\pi \cdot C^2))$-bounded.

**Lemma 2.1 (Lemma 2.4 of [Ban95])** *For any real $s > 0$ and $C > 0$, and any $\boldsymbol{x} \in \mathbb{R}^n$, we have*

$$\Pr[|\langle \boldsymbol{x}, \mathcal{D}_{\mathbb{Z}^n,s} \rangle| \geq C \cdot s\|\boldsymbol{x}\|] < 2 \cdot \exp(-\pi \cdot C^2).$$

### 2.1 The Learning With Errors Problem

Let $n, q > 0$ be integers, $\boldsymbol{s} \in \mathbb{Z}_q^n$ and $\chi$ be an error distribution over $\mathbb{Z}$. We define a distribution $\mathcal{A}^{LWE}_{n,q,\chi,\boldsymbol{s}}$ over $\mathbb{Z}_q^{n+1}$ obtained by sampling $\boldsymbol{a} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ and $e \leftarrow \chi$, and then computing

$$(\boldsymbol{a}, b) = (\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e) \in \mathbb{Z}_q^{n+1}.$$

Given many samples $(\boldsymbol{a}_i, b_i)$ from $\mathcal{A}^{LWE}_{n,q,\chi,\boldsymbol{s}}$, we can represent it by a matrix $(A, \boldsymbol{b})$ whose each row corresponds to one sample, and denoted it by LWE samples. Also we define $\mathcal{A}^{LWE}_{n,q,\alpha,\boldsymbol{s}}$ as the distribution $\mathcal{A}^{LWE}_{n,q,\chi,\boldsymbol{s}}$ where $\chi$ is a Gaussian distribution $\mathcal{D}_{\mathbb{Z},\alpha q}$ for $\alpha > 0$.

**Definition 1 (Learning with Errors).** *Let $\mathcal{S}$ be a distribution over $\mathbb{Z}_q^n$.*

- $\mathsf{LWE}_{n,q,\chi}(\mathcal{S})$*(or* $\mathsf{LWE}_{n,q,\alpha}(\mathcal{S})$*) is a problem that asks to find the secret key* $\boldsymbol{s}$*, given LWE samples from* $\mathcal{A}^{LWE}_{n,q,\chi,\boldsymbol{s}}$*(or* $\mathcal{A}^{LWE}_{n,q,\alpha,\boldsymbol{s}}$*) for a fixed* $\boldsymbol{s} \leftarrow \mathcal{S}$*.*
- $\mathsf{DLWE}_{n,q,\chi}(\mathcal{S})$*(or* $\mathsf{DLWE}_{n,q,\alpha}(\mathcal{S})$*) is a problem that asks to determine that, given arbitrarily many samples* $(\boldsymbol{a}_i, b_i) \in \mathbb{Z}_q^{n+1}$*, they are LWE samples from* $\mathcal{A}^{LWE}_{n,q,\chi,\boldsymbol{s}}$*(or* $\mathcal{A}^{LWE}_{n,q,\alpha,\boldsymbol{s}}$*) for a fixed* $\boldsymbol{s} \leftarrow \mathcal{S}$ *or uniform random samples from* $\mathcal{U}(\mathbb{Z}_q^{n+1})$*.*

Note that there is a decision-to-search reduction of LWE problem [Reg05].

**Special Distributions for Secret Vectors.** Several LWE-based cryptosystems takes the secret distribution $\mathcal{S}$ by small portion of $\mathbb{Z}_q^n$ to enhance efficiency. In particular, we will focus on the case where $\mathcal{S}$ is the set of sparse (signed) binary vectors. For the sake of simplicity, we denote

$$\mathcal{B}_{n,h} = \{\boldsymbol{s} \in \{\pm 1, 0\}^n : \mathsf{HW}(\boldsymbol{s}) = h\},$$
$$\mathcal{B}_{n,\leq h} = \{\boldsymbol{s} \in \{\pm 1, 0\}^n : \mathsf{HW}(\boldsymbol{s}) \leq h\}.$$

### 2.2 Lattice Reduction and BKZ Algorithm

Let $\Lambda$ be an $m$-dimensional lattice. Lattice reduction algorithm is an algorithm to find short basis of $\Lambda$ using a given basis $B$ of $\Lambda$. We say that a lattice reduction algorithm with root-Hermite factor $\delta_0$ returns a short basis whose first vector $\boldsymbol{b}_1$ has size $\leq \delta_0^m \cdot \det \Lambda^{1/m}$. The BKZ algorithm [CN11] is a commonly used lattice reduction. We assume the followings for BKZ algorithm for this paper.

- BKZ with blocksize $\beta$ yields root-Hermite factor $\delta_0 \approx \left(\frac{\beta}{2\pi e}(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}}$ [Che13].
- One SVP oracle call on $\beta$-dimensional lattice costs $t_\beta = 2^{0.292\beta+16.4}$, by a sieving method in [BDGL16].
- BKZ with blocksize $\beta$ costs $cn \cdot t_\beta$ clock cycles for dimension $n$ lattice, and we put $c = 16$ according to [Alb17].

We remark that the complexity estimations of our attack and other attacks largely depend on the lattice reduction cost model. In this regard, although we assume a cost model as above, our attack would be described independently from the lattice reduction cost model and then one can estimate our attack cost with their favorite cost model.

# 3 Albrecht's Improved Dual Attack

In this section, we give a detailed descrption of the dual attack and its recent variant suggested by Albrecht [Alb17], which is known as the best attack on the underlying LWE problems of fully homomorphic encryptions.

## 3.1 Simple Dual Lattice Attack

The dual lattice attack is an algorithm to solve $\mathsf{DLWE}_{n,q,\alpha}(\mathcal{S})$. The main idea of the dual attack is to exploit a short vector in the following orthogonal lattice

$$\Lambda_q^{\perp}(A) = \{\boldsymbol{v} \in \mathbb{Z}^n : \boldsymbol{v}^t A \equiv_q \boldsymbol{0}\}.$$

More precisely, for a short vector $\boldsymbol{y}$ in $\Lambda_q^{\perp}$ and an LWE sample $(A, \boldsymbol{b})$, one has

$$\langle \boldsymbol{y}, \boldsymbol{b}\rangle = \langle \boldsymbol{y}, A\boldsymbol{s} + \boldsymbol{e}\rangle = \langle \boldsymbol{y}, A\boldsymbol{s}\rangle + \langle \boldsymbol{y}, \boldsymbol{e}\rangle \equiv_q \langle \boldsymbol{y}, \boldsymbol{e}\rangle$$

and this yields $[\langle \boldsymbol{y}, \boldsymbol{b}\rangle]_q = \langle \boldsymbol{y}, \boldsymbol{e}\rangle$, which is significantly shorter than $q$. On the other hand, if the given sample $(A, \boldsymbol{b})$ is uniform random then $[\langle \boldsymbol{y}, \boldsymbol{b}\rangle]_q$ is a random value which is not small compared to the previous case. By applying this procedure for different $\boldsymbol{y}$'s, we obtain the distinguishing algorithm with overwhelming success probability. Thus we can solve the $\mathsf{DLWE}$ problem using the smallness of this inner product.

For the LWE cases with small secrets, a natural improvement of dual attack can be obtained by considering the scaled or normal form of dual lattice. More precisely, the scaled normal dual lattice is defined by

$$\Lambda_{q,c}(A) = \{(\boldsymbol{v}_1, \boldsymbol{v}_2) \in \mathbb{Z}^m \times (\tfrac{1}{c}\mathbb{Z})^n : \boldsymbol{v}_1^t A \equiv_q c \cdot \boldsymbol{v}_2\}.$$

As in the dual attack, we find a short vector $(\boldsymbol{y}_1, \boldsymbol{y}_2) \in \Lambda_{q,c}(A)$ and then compute the inner product as follows

$$\langle \boldsymbol{y}_1, \boldsymbol{b}\rangle = \langle \boldsymbol{y}_1, A\boldsymbol{s}\rangle + \langle \boldsymbol{y}_1, \boldsymbol{e}\rangle \equiv_q c \cdot \langle \boldsymbol{y}_2, \boldsymbol{s}\rangle + \langle \boldsymbol{y}_1, \boldsymbol{e}\rangle$$

for the LWE sample $(A, \boldsymbol{b} = A\boldsymbol{s} + \boldsymbol{e})$ that allows us to solve the DLWE problem.

**Choice of $c$.** We take the constant $c$ to satisfy $|c \cdot \langle \boldsymbol{y}_2, \boldsymbol{s}\rangle| \approx \mathrm{E}[|\langle \boldsymbol{y}_1, \boldsymbol{e}\rangle|]$, in order that each summand equally contributes to error $e$. First we estimate $\mathrm{E}[|\langle \boldsymbol{y}_1, \boldsymbol{e}\rangle|] \approx \frac{\alpha q}{\sqrt{2\pi}} \cdot \|\boldsymbol{y}_1\|$, and then $c$ would be taken to satisfy

$$c \approx \frac{\alpha q}{\sqrt{2\pi}} \cdot \frac{\|\boldsymbol{y}_1\|}{|\langle \boldsymbol{y}_2, \boldsymbol{s}\rangle|}.$$

Although we assume that $\boldsymbol{y}$ is short, Since the exact size of $\boldsymbol{y}_1$ and $\langle \boldsymbol{y}_2, \boldsymbol{s}\rangle$ are not sure, we heuristically assume that $\|\boldsymbol{y}_1\| \approx \sqrt{\frac{m}{m+n}}\|\boldsymbol{y}\|$ and $|\langle \boldsymbol{y}_2, \boldsymbol{s}\rangle| \approx \sqrt{\frac{h}{m+n}}\|\boldsymbol{y}\|$.

**Assumption 1** *Let $\boldsymbol{y} \in L_c(A)$ be a short vector obtained from lattice reduction. Then each entry of $\boldsymbol{y}$ has similar size $\|\boldsymbol{y}\|/\sqrt{m+n}$.*

### 3.2 Improved Dual Attack

Now we review the improvement on the dual attack on the sparse secret LWE problem [Alb17]. Most of the techniques described in this section are applicable to our hybrid attack. Hereafter we assume that the secret key $\boldsymbol{s}$ is in $\mathcal{B}_{n,h}$ for some $h \ll n$.

**Assumption on $\boldsymbol{s}$.** To exploit the sparsity of secret key, Albrecht suggests to solve the LWE problem by dual lattice attack with the assumption that some coordinates of secret key are zero. More precisely, parse the matrix $A$ into $A_1||A_2$ for two matrix $A_1 \in \mathbb{Z}_q^{m\times(n-k)}$ and $A_2 \in \mathbb{Z}_q^{m\times k}$. If the part of secret key that corresponds to $A_2$ is the zero vector, Then it holds that $\boldsymbol{b} = A\boldsymbol{s}+\boldsymbol{e} = A_1\boldsymbol{s}_1+\boldsymbol{e}$, for the parsed secret key $\boldsymbol{s} = (\boldsymbol{s}_1||\boldsymbol{s}_2) \in \mathbb{Z}_q^{n-k} \times \mathbb{Z}_q^k$ such that $\boldsymbol{s}_2 = \boldsymbol{0}$. Thus the dual attack on $A_1$ using $(\boldsymbol{y}_1, \boldsymbol{y}_2) \in \Lambda_{q,c}(A_1)$ proceeds

$$\begin{aligned}\langle \boldsymbol{y}_1, \boldsymbol{b}\rangle &= \langle \boldsymbol{y}_1, A_1\boldsymbol{s}_1 + \boldsymbol{e}\rangle\\ &= \langle \boldsymbol{y}_1, A_1\boldsymbol{s}_1\rangle + \langle \boldsymbol{y}_1, \boldsymbol{e}\rangle\\ &\equiv_q c\cdot\langle \boldsymbol{y}_2, \boldsymbol{s}_1\rangle + \langle \boldsymbol{y}_1, \boldsymbol{e}\rangle.\end{aligned}$$

Since it is sufficient to run the lattice reduction algorithm in dimension $n-k$ instead of $n$, this assumption yields the faster time to solve the DLWE problem. The drawback is the probability that the assumption holds; we minimize the product of the inverse of the probability and the time complexity to solve DLWE with this assumption by choosing appropriate $k$.

**Relaxed Assumption.** Albrecht introduces another method to relax the assumption. When $\boldsymbol{s}_2 \neq \boldsymbol{0}$, the dual attack on $A_1$ yields

$$\begin{aligned}\langle \boldsymbol{y}_1, \boldsymbol{b}\rangle &= \langle \boldsymbol{y}_1, A_1\boldsymbol{s}_1\rangle + \langle \boldsymbol{y}_1, A_2\boldsymbol{s}_2\rangle + \langle \boldsymbol{y}_1, \boldsymbol{e}\rangle\\ &\equiv_q \boldsymbol{y}_1^t A_2\boldsymbol{s}_2 + c\cdot \boldsymbol{y}_2^t\boldsymbol{s}_1 + \boldsymbol{y}_1^t\boldsymbol{e}\end{aligned}$$

and $c\cdot\boldsymbol{y}_2^t\boldsymbol{s}_1 + \boldsymbol{y}_1^t\boldsymbol{e}$ is relatively small when the sample is from LWE. We assume that the coordinates of $\boldsymbol{s}_2$ are all but up to $h'$ zero, instead of zero vector. Then the attack is done by searching possible secret $\boldsymbol{s}_2' \in \mathcal{B}_{n,\le h'}$ and check whether $\langle \boldsymbol{y}_1, \boldsymbol{b}\rangle - y_1^t A_2 \cdot \boldsymbol{s}_2'$ is far less than $q$ or not. If there is such $\boldsymbol{s}_2'$ then we decide that the given sample is from LWE.

In this strategy, the probability that assumption holds is highly increased whereas the time complexity is not much increased; in practice the adversary choose $h' \lessapprox 10$ so that the dominated part is the lattice reduction algorithm. Thus this relaxation induces the smaller estimated security of LWE. We remark that this approach can be viewed as a tradeoff between dimension and error, as also noted by Albrecht.

**Amortized Costs for Lattice Reductions.** To verify the guessed $\boldsymbol{s}_2'$ is correct or not, we should obtain several short $(\boldsymbol{y}_1, \boldsymbol{y}_2) \in \Lambda_{q,c}(A_1)$. To obtain several short vectors of similar length in a given lattice $\Lambda$, the easiest way would be repeating a lattice reduction that yields root Hermite factor $\delta_0$, which gives vectors $\boldsymbol{v}_i$ of length less than $\delta_0^m \cdot \det \Lambda^{1/m}$.

Instead, Albrecht suggested a way that performs one expensive lattice reduction (e.g. BKZ$_\beta$) on given basis to have a sufficiently short basis $B$, and apply cheap lattice reductions (e.g. LLL) repeatedly while re-randomizing the short basis $B$ by multiplying some short and sparse unimodular matrix $U$. Using sufficiently short and sparse $U$, the short vectors $\boldsymbol{v}_i$ obtained by this cheap lattice reduction which is estimated by

$$E(\|\boldsymbol{v}_i\|) = 2 \cdot \delta_0^m \cdot \det \Lambda^{1/m}.$$

For more details we refer [Alb17, Section 3].

To obtain statistically independent $(\boldsymbol{y}_1, \boldsymbol{y}_2) \in \Lambda_{q,c}(A_1)$, we have to assume that we can obtain arbitrarily many samples of DLWE. On the other hand, in many actual uses of LWE problem, there are only bounded number of samples $(A, \boldsymbol{b})$ are given; typically the number of samples would be $m = O(n)$. In this case we instead sample several short vectors $\boldsymbol{y}_i = (\boldsymbol{y}_{i,1} \| \boldsymbol{y}_{i,2})$ in a fixed lattice $\Lambda_{q,c}(A_1)$. One can perform BKZ algorithm iteratively with re-randomizing basis, or can perform LLL algorithm iteratively according to the amortizing technique.

- Iterating BKZ: For a basis $B$ of $\Lambda_{q,c}(A_1)$, iteratively perform BKZ on $B \cdot U$ while randomly sample arbitrary unimodular $U$.
- Iterating LLL: Perform BKZ on $B$ to have $B_{BKZ}$. Randomly sample a *small and sparse* unimodular $U$, and run LLL on $B_{BKZ} \cdot U$ to have a short vector. Repeat this while changing unimodular $U$.

However, if we use the same lattice $\Lambda_{q,c}(A_1)$, new $k$-dimensional samples are not independent to each other anymore, since $\boldsymbol{y}_i$ comes from the same lattice $\Lambda_{q,c}(A_1)$. Thus we heuristically assume that, the short vectors $\boldsymbol{y}_i \in \Lambda_c(A_1)$ are independent to each other, that is, we still obtain LWE$_{k,q,\chi}$ samples from $\boldsymbol{y}_i$.

**Assumption 2** *Each iterative call of BKZ (or LLL) algorithm for randomized basis of $\Lambda_{q,c}(A_1)$ gives an independent short vector $\boldsymbol{y}_i$.*

## 4 Meet-in-the-Middle Attack on LWE

In this section, we describe an attack algorithm to solve LWE by meet-in-the-middle strategy. Let $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m \times (n+1)}$ be $\mathsf{DLWE}_{n,q,\alpha}(\mathcal{B}_{n,\leq h})$ samples with secret vector $\boldsymbol{s}$. For the MITM approach, it is natural to consider the noisy relation

$$A\boldsymbol{s}_1 \approx \boldsymbol{b} - A\boldsymbol{s}_2$$

for some $\boldsymbol{s}_1 \in \mathcal{B}_{n,\leq h/2}$ and $\boldsymbol{s}_2 \in \mathcal{B}_{n,\leq h/2}$ satisfying $\boldsymbol{s} = \boldsymbol{s}_1 + \boldsymbol{s}_2$. We first prepare a table

$$\mathcal{T} = \{A\boldsymbol{v}_1 \in \mathbb{Z}_q^m : \boldsymbol{v}_1 \in \mathcal{B}_{n,\leq h/2}\}^4.$$

Then, we exhaustively investigate $\boldsymbol{v}_2 \in \mathcal{B}_{n,\leq h/2}$, while checking whether $\boldsymbol{b} - A\boldsymbol{v}_2 \in \mathbb{Z}_q^m$ is close to the set $\mathcal{T}$ where such closeness depends on the size of error $\boldsymbol{e}$. Now, if such case occurs for some $\boldsymbol{v}_2$, then we can expect that the vector $\boldsymbol{v}_2$ is the right half of secret $\boldsymbol{s}$. Otherwise, we cannot see such case for all possible $\boldsymbol{v}_2$, we conclude that the given sample is from the uniform distribution.

In this approach, finding an element in $\mathcal{T}$ that is close to $\boldsymbol{b} - A\boldsymbol{v}_2 \in \mathbb{Z}_q^m$ is the main task. A simple exhaustive method that checks every close vector to $\boldsymbol{b} - A\boldsymbol{v}_2 \in \mathbb{Z}_q^m$ surely works, but it costs too much time. We here resolve it by a search algorithm in the presence of noise that uses a locality sensitive hashing-like technique, which is adapted from Odlyzko's MITM attack on NTRU [HGHSW03].

Before explaining our algorithm, we would like to remark that this MITM attack alone does not affect the practical parameter choice of the current schemes, but this attack serves as a main subroutine of our hybrid attack algorithm that will be introduced in Section 5.

**Remark.** To the best of our knowledge, there has been two papers that mentioned the MITM approach on LWE, but both of them are problematic; Bai and Galbraith [BG14] mentioned that there is a MITM attack on LWE, but they do not give the explicit algorithm, and Albrecht, Player and Scott [APS15] presented a MITM attack on LWE based on lexicographic order sorting, which has a flaw in the analysis. We describe this flaw in Appendix. We note that a very similar algorithm is considered in a different context; for example the inhomogeneous short integer solution problem under the name *approximate merge algorithm.*

### 4.1 Noisy Collision Search

For a vector $\boldsymbol{a} \in \mathbb{Z}_q^m$, we call a vector $\boldsymbol{t} \in \mathbb{Z}_q^m$ by $B$-noisy collision of $\boldsymbol{a}$ if $\|\boldsymbol{a} - \boldsymbol{t}\|_\infty \leq B$ for some $B < q/2$. Consider a set $\mathcal{T} \subset \mathbb{Z}_q^m$ and a vector $\boldsymbol{a} \in \mathbb{Z}_q^m$. Our purpose is to determine whether there is a $B$-noisy collision $\boldsymbol{t}$ of $\boldsymbol{a}$ in $\mathcal{S}$, and if so returns such vector $\boldsymbol{t}$. We mainly exploits a simple locality sensitive hashing $\mathsf{sgn} : \mathbb{Z}_q \to \{0, 1\}$, which defined as $\mathsf{sgn}(x) = 1$ for $x \in [0, q/2)$ and 0 otherwise. For every $B$-noisy collision $\boldsymbol{t} = (t_i)$ of $\boldsymbol{a} = (a_i)$, the sign of $i$-th entries $\mathsf{sgn}(a_i)$ and $\mathsf{sgn}(b_i)$ must coincide if $a_i \in V_B := [-q/2 + B, -B) \cup [B, q/2 - B)$.

For a vector $\boldsymbol{a} = (a_i) \in \mathbb{Z}_q^m$, define an index set $I_{\boldsymbol{a}} := \{i : a_i \in V_B\}$, and define a function $\mathsf{sgn}' : \mathbb{Z}_q \to \{0, 1, \mathsf{x}\}$ that returns $\mathsf{sgn}(a)$ if $a \in V_B$, and

[3] Another way to use MITM method is to parse $A$ and $\boldsymbol{s}$ into $[A_l|A_r]$ and $\boldsymbol{s} = (\boldsymbol{s}_l||\boldsymbol{s}_r)$ for $n/2$ dimension vectors. In the regards of the overall attack complexity that product of the time and the inverse of probability, the method discussed in the main body is better; The MITM with parsing takes less time and memory but the success probability is far less compared to the MITM in the paper.

otherwise $\mathsf{x}$. Then from the above observation, we have the following fact that becomes a foundation of our algorithm

*If $\mathcal{T}$ has a $B$-noisy collision of $\boldsymbol{a}$, then there is a binary string $(b_1, \cdots, b_m) \in \mathsf{sgn}(\mathcal{T})$ such that $b_i = \mathsf{sgn}'(a_i)$ for every index $i$ in $I_{\boldsymbol{a}}$.*

**Detailed Algorithms.** We give two algorithms `Preprocess` and `Search`, where the former literally preprocess the set $\mathcal{T}$, and the latter investigate whether $\mathcal{T}$ has a $B$-noisy collision of input $\boldsymbol{a} \in \mathbb{Z}_q^m$.

- `Preprocess`: On input $\mathcal{T} \subset \mathbb{Z}_q^m$,

```
1. Initialize an empty hash table $\mathcal{H}$ with $2^m$ (empty) linked lists with indexes in $\{0,1\}^m$.
2. For each $\boldsymbol{t} \in \mathcal{T}$,
   (a) append $\boldsymbol{t}$ into the linked list indexed $\mathsf{sgn}(\boldsymbol{t})$.
3. Return nonempty linked lists $\mathcal{H}$.
```

- `Search`: On input a hash table $\mathcal{H}$, a query $\boldsymbol{a} \in \mathbb{Z}_q^m$ and distance bound $B$,

```
1. For each $\mathsf{bin} \in \{0,1\}^m$ obtained from $\mathsf{sgn}'(\boldsymbol{a})$ by replacing $\mathsf{x}$ by 0 or 1,
   (a) If $\mathcal{H}$ has a linked list indexed $\mathsf{bin}$, for each $\boldsymbol{t}$ in the list,
       i. Check whether $\|\boldsymbol{a} - \boldsymbol{t}\|_\infty \leq B$. If so, return $\boldsymbol{t}$.
2. Return $\perp$.
```

**Algorithm Analysis.** First, the following proposition asserts that our algorithm can find the $B$-noisy collision, if exists.

**Proposition 4.1 (Correctness)** *Let $\mathcal{T}$ be a nonempty subset of $\mathbb{Z}_q^m$ and $\mathcal{H}$ be the output of Preprocess algorithm on input $\mathcal{T}$. Then `Search` algorithm with input $(\mathcal{L}, \boldsymbol{a}, B)$ returns a vector if and only if there is a $B$-noisy collision of $\boldsymbol{a}$ in $\mathcal{H}$. In particular, every returned vector is a $B$-noisy collision of $\boldsymbol{a}$.*

*Proof.* The second claim is immediate. For the first claim, one direction is clear since the output vector itself is a noisy collision in $\mathcal{T}$. Conversely, suppose that $\mathcal{T}$ has a noisy collision $\boldsymbol{t}$. Since $\mathsf{sgn}(\boldsymbol{t})$ would be one of strings obtained from $\mathsf{sgn}'(\boldsymbol{a})$, it outputs $\boldsymbol{t}$ unless it terminates before then with some vector $\boldsymbol{t}'$.

To investigate the (time) cost of Algorithms, we presents some lemmas.

**Lemma 4.2** *If $\boldsymbol{a} \overset{\$}{\leftarrow} \mathbb{Z}_q^m$, $|I_{\boldsymbol{a}}|$ follows a binomial distribution $B(m, 1 - 4B/q)$.*

*Proof.* Since $\boldsymbol{a}$ is sampled from $\mathcal{U}(\mathbb{Z}_q^m)$, the probability that each component $a_i$ is not in $V_B$ is $4B/q$. Each component of $a_i$ is independent, and then we know the number of $\mathsf{x}$ in $\mathsf{sgn}'(\boldsymbol{a})$ follows a binomial distribution $B(m, 4B/q)$.

**Lemma 4.3** *Suppose the elements of the table $\mathcal{T}$ come from uniform distribution over $\mathbb{Z}_q^m$. For any $\textsf{bin} \in \{0,1\}^m$,*

$$\Pr\left[L_{\textsf{bin}} \neq \emptyset\right] \leq \frac{|\mathcal{T}|}{2^m}.$$

*Proof.* Note that $L_{\textsf{bin}} \neq \emptyset$ if and only if $\textsf{bin} \in \textsf{sgn}(\mathcal{T})$. Since $\mathcal{T}$ is uniformly distributed, the probability of $\textsf{bin} \notin \textsf{sgn}(T)$ is $\left(1 - \frac{1}{2^m}\right)^{|\mathcal{T}|} \geq 1 - \frac{|\mathcal{T}|}{2^m}$, which proves the claim.

Now assuming that the linked list insertion costs $O(1)$, the cost of `Preprocess` is clearly $O(|\mathcal{T}|)$. The costs of `Search` consists of $2^{m-|I_{\boldsymbol{a}}|}$ times of hash table lookups, and some computations of $\|\cdot\|_\infty$ norm. We first claim that $|I_{\boldsymbol{a}}|$ would be $m(1-4B/q)$ (stated in Lemma 4.2), which implies `Search` look ups the hash table about $2^{4mB/q}$ times.

Next we claim that by Heuristic 4.4, if $m$ is sufficiently large[5], the computation of $\|\cdot\|_\infty$ almost never occur for a randomly chosen query $\boldsymbol{a} \in \mathbb{Z}_q^m$.

**Heuristic 4.4** *Let $m, q > 0$ be positive integers and $B \in (0, q/4)$, and consider $\mathcal{T} \subset \mathbb{Z}_q^m$ whose element is sampled from uniform distribution. Let $\mathcal{H}$ be output of `Preprocess` on input $\mathcal{T}$. If*

$$m \geq 2\log(|\mathcal{T}|)/(1-4B/q), \tag{1}$$

*then for a random vector $\boldsymbol{a} \leftarrow \mathbb{Z}_q^m$, the probability that `Search` never computes $\|\cdot\|_\infty$ norm is $\geq 1 - 1/|\mathcal{T}|$.*

We justify the heuristic as follows: Since $|I_{\boldsymbol{a}}| = m(1-4B/q)$ for random $\boldsymbol{a} \in \mathbb{Z}_q^m$ on average by Lemma 4.2, we heuristically assume that `Search` visits $2^{4mB/q}$ indexes. Since $\Pr\left[L_{\textsf{bin}} \neq \emptyset\right] \leq \frac{|\mathcal{T}|}{2^m}$ by Lemma 4.3, we bound the probability that `Search` never visits nonempty linked lists by $\left(1 - \frac{|\mathcal{T}|}{2^m}\right)^{4mB/q}$. One can easily check that if such choice of $m$ yields the claim.

Considering all above, we assess the total time cost in Table 1.

| `Preprocess` | `Search`[6] |
|---|---|
| $\lvert\mathcal{T}\rvert \cdot m$<br>(operations on $\mathbb{Z}_q$) | $O(2^{4mB/q})$<br>(table look-ups) |

Table 1: Time cost for noisy search

[5] Note that, when we use noisy collision search to solve LWE, the parameter $m$ is the number of samples of given LWE instances so it can be freely chosen by adversary.

[6] Per one query in average.

### 4.2 Noisy Meet-in-the-middle Attack on LWE

We now present a (noisy) MITM attack for LWE, using noisy collision search. Formal description is given by Algorithm 1. We would like to remark that, since we mainly exploit this algorithm as a subroutine of the main hybrid attack for DLWE, Algorithm 1 is also described for DLWE although it can actually solve the search version of LWE. Here, we define $\mathcal{B}_{n,h}$ by a set of vectors in $\{0, \pm 1\}_q^n$ with $h$ number of nonzero entries. Also, $\mathcal{B}_{n,\leq h}$ denotes $\cup_{i=0}^{h} \mathcal{B}_{n,i}$.

**Algorithm 1** Meet-in-the-middle attack for binary sparse LWE problems

```
Input: A matrix $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m \times (n+1)}$
    Hamming weight parameters $h_1, h_2 > 0$
Output: 1 if $(A, \boldsymbol{b})$ is from LWE distribution, and 0 otherwise
Compute $\mathcal{T} = \{A\boldsymbol{v}_1 : \boldsymbol{v}_1 \in \mathcal{B}_{n,\leq h_1}\}$
Run Preprocess on input $\mathcal{T}$ to have a hash table $\mathcal{H}$.
for $\boldsymbol{v} = \boldsymbol{b} - A\boldsymbol{v}_2 \in \mathbb{Z}_q^m$ for each $\boldsymbol{v}_2 \in \mathcal{B}_{n,h_2}$ do
    if Search on input $(\mathcal{H}, \boldsymbol{v}, B)$ returns a vector, then return 1
end for
return 0
```

One can easily check that correctness of Algorithm 1 comes immediately from the correctness of noisy search.

**Proposition 4.5** *Let $h_1, h_2 > 0$ be positive integers, $\chi$ be a $(B, \varepsilon)$-bounded distribution over $\mathbb{Z}$, and let $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m \times (n+1)}$ be $\mathcal{A}_{n,q,\chi,s}^{LWE}$ samples where $\boldsymbol{s} \in \mathcal{B}_{n,\leq h_1+h_2}$. Then Algorithm 1 returns 1 for input $(A, \boldsymbol{b})$ and $h_1, h_2$ with probability $\geq (1-\varepsilon)^m$.*

*Proof.* If input $(A, \boldsymbol{b})$ is LWE sample with sparse ternary secret $\boldsymbol{s} \in \mathcal{B}_{n,\leq h_1+h_2}$, we exhaustively run the noise search on $\boldsymbol{v}_1 \in \mathcal{B}_{n,t_1}$ for $t_1 \leq h_1$ and $\boldsymbol{v}_2 \in \mathcal{B}_{n,t_2}$ for $t_2 \leq h_1$. These search should find $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ such that $\boldsymbol{s} = \boldsymbol{s}_1 + \boldsymbol{s}_2$ and in this case the following equations holds:

$$\|A\boldsymbol{s}_1 - (\boldsymbol{b} - A\boldsymbol{s}_2)\|_\infty = \|A\boldsymbol{s} - \boldsymbol{b}\|_\infty = \|\boldsymbol{e}\|_\infty$$

Since Algorithm 1 returns 1 if $\|\boldsymbol{e}\|_\infty \leq B$ and each coordinate of error $\boldsymbol{e}$ follows $\chi$, we conclude the algorithm succeeds with probability $\geq (1-\varepsilon)^m$.

To apply the analyses of noisy collision search, we need the following assumption that says that the vectors in table and queries are randomly distributed over $\mathbb{Z}_q^m$.

**Assumption 3** *For a fixed matrix $A \in \mathbb{Z}_q^{m \times n}$, a distribution of vectors of the form $A\boldsymbol{s}$ where $\boldsymbol{s} \leftarrow \mathcal{B}_{n,\leq h}$ is sufficiently close to the uniform distribution over $\mathbb{Z}_q^m$.*

**Proposition 4.6** *Suppose that Assumption 3 holds. Then for a uniformly random matrix* $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m \times (n+1)}$*, Algorithm 1 returns* 0 *for input* $(A, \boldsymbol{b})$ *and parameters* $h_1, h_2$ *with probability* $\geq 1 - N_{\mathcal{T}} N_q \cdot \left(\frac{2B}{q}\right)^m$*, where* $N_{\mathcal{T}}(n, h_1)$ *and* $N_q(n, h_2)$ *denotes the number of vectors in table and the number of query.*

*Proof.* By Assumption 3, we consider every query $\boldsymbol{v} = \boldsymbol{b} - A\boldsymbol{v}_2$ as a random sample from $\mathbb{Z}_q^m$. Then again from the assumption, the set $\mathcal{T}$ is randomly distributed on $\mathbb{Z}_q^m$, and we conclude that the probability that a $B$-noisy collision of $\boldsymbol{v}$ is in $\mathcal{T}$ is less than $N_{\mathcal{T}}(2B/q)^m$. Since we try at most $N_q$ queries, the claim holds.

Clearly, the time complexity of Algorithm 1 is the sum of table construction and `Preprocess` time $T_{pre}$, and total noisy search time $T_{search}$. Clearly, the size of table $N_{\mathcal{T}}$ and the number of query $N_q$ is given by

$$N_{\mathcal{T}} = \sum_{i=1}^{h_1} \binom{n}{i} \cdot 2^i, \quad N_q = \sum_{i=1}^{h_2} \binom{n}{i} \cdot 2^i \tag{2}$$

for given $h_1, h_2$. Finally, by supposing Assumption 3 holds and the condition for $m$ (1), we have the following cost estimation.

- $T_{pre}$ consists of $N_{\mathcal{T}} \cdot n^2$ operations over $\mathbb{Z}_q$ on constructing table $\mathcal{T}$, and `Preprocess` also requires $N_{\mathcal{T}} \cdot m$ operations.
- Since each `Search` call for each query costs $2^{4mB/q}$ in average, we have $T_{search} = O(N_q \cdot 2^{4mB/q})$.

| Memory | Time | |
|---|---|---|
| | $T_{pre}$ | $T_{search}$ |
| $N_{\mathcal{T}} \cdot m$<br>(bits) | $N_{\mathcal{T}} \cdot (n^2 + m)$<br>(operations) | $O(N_q \cdot 2^{4mB/q})$<br>(table look-ups) |

Table 2: Cost for Algorithm 1 with inputs a matrix in $\mathbb{Z}_q^{m \times (n+1)}$ and $h_1, h_2$.

## 5 A New Hybrid Attack for the LWE Problem

In this section, we propose a hybrid attack that combines lattice reduction and the MITM attack. More precisely, we use dual attack as a trade-off method for LWE sample, which increases the error size and reduces dimension and Hamming weight of secret vector. Since MITM attack of the previous section cost heavily depends on the dimension of secret vector but less sensitive to error size, this trade-off largely decreases the MITM attack cost.

### 5.1 Dimension-error Trade-off of LWE

In this section we interpret Albrecht's dual attack as dimension-error trade-off with detailed analysis. For given LWE samples $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m\times(n+1)}$ from $\mathcal{A}_{n,q,\alpha,\boldsymbol{s}}^{LWE}$ for $k < n$, divide $A$ into $A_1$ and $A_2$ consisting of the first $n-k$ columns and the remaining $k$ columns. For any vectors $(\boldsymbol{y}_1, \boldsymbol{y}_2) \in \Lambda_{q,c}(A_1)$, it holds that

$$\begin{aligned}\langle \boldsymbol{y}_1, \boldsymbol{b}\rangle &= \langle \boldsymbol{y}_1, A_1\boldsymbol{s}_1\rangle + \langle \boldsymbol{y}_1, A_2\boldsymbol{s}_2\rangle + \langle \boldsymbol{y}_1, \boldsymbol{e}\rangle \\ &\equiv_q \boldsymbol{y}_1^t A_2 \boldsymbol{s}_2 + c\cdot \boldsymbol{y}_2^t \boldsymbol{s}_1 + \boldsymbol{y}_1^t \boldsymbol{e}\end{aligned}$$

where $\boldsymbol{s}_2$ is the last $k$ entries of $\boldsymbol{s}$. Now, if $(\boldsymbol{y}_1, \boldsymbol{y}_2)$ is sufficiently short to satisfy $\langle \boldsymbol{y}_1, \boldsymbol{e}\rangle, \langle \boldsymbol{y}_2, \boldsymbol{s}_1\rangle \ll q$, we have a new LWE-like sample

$$({\boldsymbol{y_1}}^t A_2, \langle \boldsymbol{y}_1, \boldsymbol{b}\rangle) = (\boldsymbol{a}', \langle \boldsymbol{a}', \boldsymbol{s}_2\rangle + e') \in \mathbb{Z}_q^{k+1},$$

with new secret vector $\boldsymbol{s}_2$ and error $e' = c\cdot\langle \boldsymbol{y}_2, \boldsymbol{s}_1\rangle + \langle \boldsymbol{y}_1, \boldsymbol{e}\rangle$.

**Algorithm 2** A Dimension-error Trade-off

```
Input: A matrix $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m\times(n+1)}$
    Root Hermite factor $\delta_0$
    Dimension trade-off parameter $0 < k < n$
Output: A vector $(\boldsymbol{a}', b') \in \mathbb{Z}_q^{k+1}$.
Parse $A$ into $[A_1 || A_2]$ with $A_1 \in \mathbb{Z}_q^{m\times(n-k)}$ and $A_2 \in \mathbb{Z}_q^{m\times k}$
$\boldsymbol{y} = (\boldsymbol{y}_1 || \boldsymbol{y}_2) \leftarrow BKZ_{\delta_0}(\Lambda_{q,c}^{\perp}(A_1))$
return $(\boldsymbol{a}', b') \leftarrow (\boldsymbol{y}_1^t A_2, \langle \boldsymbol{y}, \boldsymbol{b}\rangle) \in \mathbb{Z}_q^{k+1}$.
```

We now have Algorithm 2 for the dimension-error trade-off, while assuming Assumption 1 to justify the choice for $c$ in Section 3.1. In other words, we choose $c = \frac{\alpha q}{\sqrt{2\pi}} \cdot \frac{\|\boldsymbol{y}_1\|}{|\langle \boldsymbol{y}_2, \boldsymbol{s}_1\rangle|}$ and assume that each entry of $\boldsymbol{y}$ has similar size $\|\boldsymbol{y}\|/\sqrt{m+n}$. We formally state that Algorithm 2 can serve a trade-off algorithm on the LWE problem as follows.

**Proposition 5.1** *Assume that Assumption 1 holds for outputs of BKZ algorithm with root-Hermite factor $\delta_0$. Then for given $\mathcal{A}_{n,q,\alpha,\boldsymbol{s}}^{LWE}$ samples $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m\times n}$, Algorithm 2 returns one $\mathcal{A}_{k,q,\chi,\boldsymbol{s}'}^{LWE}$ sample $(\boldsymbol{a}', b') \in \mathbb{Z}_q^{k+1}$, where $\boldsymbol{s}' = (s_{n-k+1}, \cdots, s_n)$. In particular, the error distribution $\chi$ is $(B, 2e^{-4\pi})$-bounded with*

$$B = \left(2 + \frac{1}{\sqrt{2\pi}}\right) \cdot \sqrt{\frac{m}{m+n}} \cdot \alpha q \cdot \|\boldsymbol{y}\| \tag{3}$$

*Proof.* It only remains to show the error bound part. From Lemma 2.1, we know $\|\langle \boldsymbol{y}_1, \boldsymbol{e}\rangle\| < 2\alpha q \cdot \|\boldsymbol{y}_1\|$ with probability $\geq 1 - 2e^{-4\pi}$. Therefore, with probability

$\geq 1 - 2e^{-4\pi}$, we have

$$\begin{aligned}|e'| &\leq |\langle \boldsymbol{y}_1, \boldsymbol{e}\rangle| + |c \cdot \langle \boldsymbol{y}_2, \boldsymbol{s}\rangle| \\ &\leq 2\alpha q \cdot \|\boldsymbol{y}_1\| + \frac{\alpha q}{\sqrt{2\pi}} \cdot \|\boldsymbol{y}_1\| \\ &\leq (2 + \frac{1}{\sqrt{2\pi}}) \cdot \alpha q \cdot \|\boldsymbol{y}_1\|.\end{aligned}$$

Since Assumption 1 guarantees $\|\boldsymbol{y}_1\| \approx \sqrt{\frac{m}{m+n}}\|\boldsymbol{y}\|$, we show (3).

**Amortizing and Heuristic for Algorithm 2.** We remark that Albrecht's amortizing technique and heuristic assumption described in Section 3 works well for this trade-off. More precisely, the amortizing technique reduces the time cost for multiple run of tradeoff algorithm into, essentially, the time cost of one run of Algorithm 2. On the other hand, we can obtain arbitrary many independent trade-offed LWE samples from the bounded number, e.g. $m = O(n)$, of given LWE samples under the heuristic assumption. We employ these techniques in the hybrid attack and estimation as well.

### 5.2 Our Hybrid Attack

Now we are able to describe our hybrid attack, which is formally written in Algorithm 3. We first explain how to choose parameters $m$ and $\tau$ optimally from inputs. The concrete formula for each parameters can be found in Appendix 6.

- The number of $n$-dim DLWE samples $m$ is set to minimize the short vectors obtained from $\mathrm{BKZ}_{\delta_0}$.
- The error bound $B$ is subsequently obtained from $m$ by Proposition 5.1.
- The number of $k$-dim DLWE samples $\tau$ is chosen according to Heuristic 4.4[7], in order to ensure that Algorithm 1 runs in time proportional to $2^{4\tau B/q}$.

The following theorem shows the results of Algorithm 3 for LWE samples.

**Theorem 5.2** *Let $\boldsymbol{s} \in \mathcal{B}_{n,h}$. Given sufficiently many $\mathcal{A}^{LWE}_{n,q,\alpha,\boldsymbol{s}}$ samples, Algorithm 3 returns* 1 *with probability*

$$p = (1 - 2e^{-4\pi})^m \cdot \sum_{0 \leq i \leq h_1 + h_2} \binom{n-h}{k-i}\binom{h}{i} / \binom{n}{k}.$$

*Proof.* Let the secret vector $\boldsymbol{s}$ be $\boldsymbol{s} = (\boldsymbol{s}_1\|\boldsymbol{s}_2)$ which is seperated as $\boldsymbol{y} = (\boldsymbol{y}_1\|\boldsymbol{y}_2)$. This means that we run Algorithm 1 by input $(A', \boldsymbol{b}')$, which has $\boldsymbol{s}_2$ as its LWE

[7] We note that the parameter $\tau$ does not critically affect to the performance when we use the amortization technique. Hence we choose $\tau$ as in heuristical computation.

**Algorithm 3** A new hybrid attack for sparse binary secret LWE Problems

```
Input: (Sufficiently many) $\mathsf{DLWE}_{n,q,\alpha}(\mathcal{B}_{n,h})$ samples $(\boldsymbol{a}_i, b_i) \in \mathbb{Z}_q^{n+1}$
    Root Hermite factor $\delta_0$
    Dimension trade-off parameter $0 < k < n$
    MITM parameter $0 \leq h_1, h_2 \leq h$
Output: 1 if $(\boldsymbol{a}_i, b_i)$'s are sampled from LWE distribution, and 0 otherwise.
Set $m, B$ and $\tau$ as optimal values.
// Dimension-error trade-off
for $i$ from 1 to $\tau$ do
    Let $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m\times(n+1)}$ be DLWE$_{n,q,\alpha}(\mathcal{B}_{n,h})$ samples.
    Run Algorithm 2 on input $(A, \boldsymbol{b})$, $\delta_0$, and $k$ to obtain $(\boldsymbol{a}'_i, b'_i) \in \mathbb{Z}_q^{k+1}$.
end for
$(A', \boldsymbol{b}') \in \mathbb{Z}_q^{\tau\times(k+1)}$ be a matrix having $i$-th row $(\boldsymbol{a}'_i, b'_i)$.
// No need to perform MITM if $\boldsymbol{s}_2 = \boldsymbol{0}$
if $\|\boldsymbol{b}'_i\|_\infty \leq B$ then
    return 1
end if
// Perform MITM
if Algorithm 1 on input $(A', \boldsymbol{b}'), B, h_1$, and $h_2$ outputs 1 then
    return 1
end if
return 0
```

secret. Thus Algorithm 1 returns 1 if and only if $\mathsf{HW}(\boldsymbol{s}_2) \leq h_1 + h_2$. This probability is

$$p' = \sum_{0 \leq i \leq h_1+h_2} \binom{n-h}{k-i}\binom{h}{i} / \binom{n}{k}.$$

From the choice of $B$ and Proposition 5.1, we get a$p = (1 - 2e^{-4\pi})^m \cdot p'$.

Under the amortizing technique and heuristic assumption, the time cost of the trade-off phase is approximately one lattice reduction, and the condition *sufficiently many* is removed. Overall, the total time complexity of Algorithm 3 is dominated by the sum of lattice reduction time $T_{lat}$ and Algorithm 1 time $T_{pre}+T_{search}$. Since we take $\tau$ according to Heuristic 4.4, the table 2 is also applicable to this case, which yields the following time cost table with the amortizing technique.

| Memory | Time | | |
|---|---|---|---|
| | $T_{lat}$ | $T_{pre}$ | $T_{search}$ |
| $N_{\mathcal{T}} \cdot \tau$<br>(bits) | $\approx T_{BKZ,\delta_0}$ | $N_{\mathcal{T}} \cdot (k^2 + \tau)$<br>(operations) | $O(N_q \cdot 2^{4\tau B/q})$<br>(table look-ups) |

Table 3: Cost for Algorithm 3 with a matrix $(A, b) \in \mathbb{Z}_q^{\tau\times(n+1)}$ and inputs $h_1, h_2$.

## 6 Attack Complexity Estimation

In the previous sections, we analyze the running time $T$ and success probability $p$ of our attack for given parameters $k, h_1, h_2$. In this section, we show estimations of the bit-security[8] of the LWE problem with respect to our attack by

$$\log T - \log p \tag{4}$$

for optimized selections of $k, h_1, h_2$ to minimize the above bit-security of the LWE problem.

We implement an estimator that computes the optimized bit-security of the LWE problem against our hybrid attack[9] by appropriately choosing $\delta_0, k, h_1, h_2$. We assume the followings for our complexity estimation.

- The costs of table look-up and linked list insertion are equal to one ring operation in $\mathbb{Z}_q$ in the estimator.
- The cost of `Search` algorithm is estimated by $2^{4mB/q}$.
- The amortizing technique and heuristic assumption discussed in Section 3 and 5 are also applied.

As an example, we give table 4 that estimates our attack complexity by running estimator code for sparse ternary LWE problems for various $n$ and $q$ while $\alpha$ and $h$ is fixed by $8/q$ and 64. We remark that those large scale parameters are actually being used for many applications [CHK+18, JKLS18, CCS19], but not all of them use sparse secret. The 'best' row comes from `LWE-estimator` version 2019-2-14 with `BKZ.sieve` model [APS15].

Our attack shows better performance than the current best attack (Albrecht's dual attack) for modulus $q \geq 2^{40}$, however it is reversed for smaller modulus. In this regard, we note that Albrecht's dual attack that can be regarded as a special case of our attack with $h_1 = 0$, and hence, if we investigate all possible parameter range in our code, our algorithm must outperform Albrecht's dual attack. However it takes too much time to check all possible parameter ranges, and we instead investigate plausible range of parameters; our code only explores the parameter regime that $h_1, h_2 \gtrsim h/2$, and this may not capture the real optimal point. Meanwhile, the estimations for small modulus $q$ size implies the exhaustive search is better than the MITM approach for that parameter, which seems weird at first glance. However this enough make sense because our MITM algorithm runtime exponentially grows with $B/q$, where $B$ is the error size. Then, to have small $B/q$ after the dimension-error trade-off, we may have to find shorter vectors in the lattice reduction stage than Albrecht's dual attack. Particularly for small modulus $q$, the additional cost for finding such shorter

[8] Although there is no formal definition for bit-security, (4) is one of generally accepted methods. Indeed, the widely used LWE attack complexity estimator `LWE-estimator` [APS15] also compute the bit-security according to (4).

[9] Code can be found at `github.com/swanhong/HybridLWEAttack`. Besides bit-security estimation, we also confirm that our attack actually works by implementing it, whose code can also be found in the same page.

vector offsets the benefit of MITM approach, and finally the results in table 4 occurs.

| $n$ | 1024 | 2048 | | 4096 | | 8192 | | 16384 | | 32768 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\log q$ | 25 | 38 | 45 | 67 | 82 | 125 | 158 | 250 | 350 | 505 | 628 |
| Best Attack [Alb17] | 116.7 | 135.6 | 127.7 | 164.2 | 129.5 | 175.5 | 128.6 | 152.0 | 128.3 | 145.4 | 127.2 |
| Ours | 130.7 | 135.2 | 118.8 | 131.7 | 113.7 | 128.4 | 113.9 | 125.7 | 104.6 | 128.5 | 112.5 |
| Our Mem | 74 | 76.3 | 74.6 | 74.4 | 73.3 | 71.8 | 78.8 | 77.5 | 75.6 | 71.2 | 70.5 |

Table 4: Cost table with memory capacity bound $2^{80}$

Our attack claims that fully homomorphic encryption implementations that uses the sparse ternary LWE problem with large modulus $q$ should change the parameter selection. In particular, `HElib` [HS13] and `HEAAN` [Kim18] use the sparse ternary secret basically. `SEAL` [LP16] uses the (non-sparse) ternary secret key but the paper [CH18] that supports bootstrapping for `SEAL` also uses the sparse ternary secret vector. On the other hand, for the Post-Quantum Cryptography Standardization held by NIST, our attack cannot make any impact on those schemes since they use too small parameter size, although there are some LWE-based schemes using sparse secret.

## References

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange—a new hope. In *Proc. of USENIX Security '16*, pages 327–343. USENIX Association, 2016.

[Alb17] Martin R Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in helib and SEAL. In *Proc. of EUROCRYPT '17*, pages 103–129. Springer, 2017.

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

[Bab86] László Babai. On lovász'lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[Ban95] Wojciech Banaszczyk. Inequalities for convex bodies and polar reciprocal lattices inr n. *Discrete & Computational Geometry*, 13(2):217–231, 1995.

[BCD$^+$16] Joppe Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! practical, quantum-secure key exchange from lwe. In *Proc. of ACM CCS '16*, pages 1006–1018. ACM, 2016.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proc. of SODA '16*, pages 10–24, 2016.

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

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

[BLP$^+$13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proc. of STOC '13*, pages 575–584. ACM, 2013.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.

[BVG12] Zvika Brakerski, Vinod Vaikuntanathan, and Craig Gentry. Fully homomorphic encryption without bootstrapping. In *Proc. of ITCS'12*. Citeseer, 2012.

[CCS19] Hao Chen, Ilaria Chillotti, and Yongsoo Song. Improved bootstrapping for approximate homomorphic encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 34–54. Springer, 2019.

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachene. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 3–33. Springer, 2016.

[CH18] Hao Chen and Kyoohyung Han. Homomorphic lower digits removal and improved fhe bootstrapping. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 315–337. Springer, 2018.

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

[CHK$^+$18] Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. Bootstrapping for approximate homomorphic encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 360–384. Springer, 2018.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Proc. of ASIACRPYT'17*, pages 409–437. Springer, 2017.

[CKLS18] Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yongsoo Song. Lizard: Cut off the tail! a practical post-quantum public-key encryption from lwe and lwr. In *International Conference on Security and Cryptography for Networks*, pages 160–177. Springer, 2018.

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

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

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

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Proc. of CRYPTO'13*, pages 75–92. Springer, 2013.

[HG07] Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against ntru. *Proc. of CRYPTO '07*, pages 150–169, 2007.

[HGHSW03] Nick Howgrave-Graham, Joseph H. Silverman, and William Whyte. A meet-in-the-middle attack on an ntru private key, 07 2003.

[HS13] Shai Halevi and Victor Shoup. Design and implementation of a homomorphic-encryption library. *IBM Research (Manuscript)*, 6:12–15, 2013.

[HS14] Shai Halevi and Victor Shoup. Algorithms in helib. In *Proc. of CRYPTO '14*. Springer Verlag, 2014.

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

[JKLS18] Xiaoqian Jiang, Miran Kim, Kristin Lauter, and Yongsoo Song. Secure outsourced matrix computation and application to neural networks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1209–1222. ACM, 2018.

[Kim18] Andrey Kim. HEAAN. `https://github.com/kimandrik/HEAAN`, 2018.

[LP11] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for lwe-based encryption. In *Proc. of CT-RSA' 11*, volume 65–58, pages 319–339. Springer, 2011.

[LP16] Kim Laine and Rachel Player. Simple encrypted arithmetic library-seal (v2. 0). Technical report, Technical report, September, 2016.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2009.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. of STOC '05*, pages 84–93. ACM, 2005.

[Wun16] Thomas Wunderer. Revisiting the hybrid attack: Improved analysis and refined security estimates. Cryptology ePrint Archive, Report 2016/733, 2016. `https://eprint.iacr.org/2016/733`.

# A Flaw of Meet-in-the-middle Attack in [APS15] on LWE

Albrecht, Player and Scott [APS15] consider a meet-in-the-middle (MITM) attack on LWE based on lexicographic order sorting, but it has a significant flaw in the analysis. We discuss the flaw in this section.

The purpose of the MITM attack in [APS15] is to find $(\boldsymbol{s}_1|\boldsymbol{s}_2) = \boldsymbol{s}$ for $\boldsymbol{s}_1, \boldsymbol{s}_2 \in \mathbb{Z}_q^{n/2}$. The attack proceeds as follows:

- for given DLWE sample $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m\times(n+1)}$ with secret vector $\boldsymbol{s}$, parse $A$ into $(A_1, A_2)$ for $A_1 \in \mathbb{Z}_q^{m\times n/2}$ and $A_2 \in \mathbb{Z}_q^{m\times n/2}$, and store $A_1\boldsymbol{t}_1$ for every possible left candidate $\boldsymbol{t}_1 \in \mathbb{Z}_q^{n/2}$ in *lexicographic* order.
- for each right candidate $\boldsymbol{t}_2 \in \mathbb{Z}_q^{n/2}$, insert $\boldsymbol{b} - A_2\boldsymbol{t}_2$ in the list by binary search and then check that two adjacent vectors $A_1\boldsymbol{t}_1$ satisfy whether $(\boldsymbol{t}_1|\boldsymbol{t}_2) = \boldsymbol{s}$.

Unfortunately, this approach may fail to output appropriate $(\boldsymbol{s}_1|\boldsymbol{s}_2)$ since we cannot guarantee that $\boldsymbol{b} - A_2\boldsymbol{s}_2$ and $A_1\boldsymbol{s}_1$ are the nearest pair in the lexicographic order; there might exist many different elements in the list such that the (lexicographical) distance from $\boldsymbol{b} - A_2\boldsymbol{s}_2$ is less than the distance between $\boldsymbol{b} - A_2\boldsymbol{s}_2$ and $A_1\boldsymbol{s}_1$. This flaw comes from the fact that lexicographic order only ensures that two adjacent vectors have very near entries for some first few coordinates, and for the other coordinates it does not ensure anything. In particular, these elements make the success probability of algorithm be negligibly small in practice.

More precisely, suppose that $A_1\boldsymbol{t}$'s are uniformly, independently distributed and the first coordinates of $\boldsymbol{b} - A_2\boldsymbol{s}_2$ and $A_1\boldsymbol{s}_1$ have a difference $B > 0$. Then the probability that each $A_1\boldsymbol{t}$ are nearer to $\boldsymbol{b} - A_2\boldsymbol{s}_2$ than $A_1\boldsymbol{s}_1$ is at least $(2B-1)/q$. Since those probabilities are independent, the probability that there is such $A_1\boldsymbol{t}$ in the (lexicographic order) list is $1 - ((2B-1)/q)^T$ for the size of list $T$, which is very close to 1 even for polynomially large $T$. (The size of $T$ is usually exponentially large.) Hence the probability that the algorithm success is also negligible.

## Optimal Parameter choices in Algorithm 3

- $m = \sqrt{\frac{n \log q/c}{\log \delta_0}}$,
- $B = (2 + \frac{1}{\sqrt{2\pi}}) \cdot \alpha q \sqrt{\frac{m}{m+n}} \cdot 2^{2\sqrt{n \log \delta_0 \log q/c}}$
- $\tau = \frac{1}{1-4B/q} \log(N_{\mathcal{T}} \cdot N_q)$, where

$$N_{\mathcal{T}} = |\mathcal{T}| = \sum_{i=1}^{h_1} \binom{k}{i} \cdot 2^i, \quad N_q = \sum_{i=1}^{h_2} \binom{k}{i} \cdot 2^i.$$