Revisiting the Hybrid attack on sparse and ternary secret LWE

Yongha Son[1] and Jung Hee Cheon[1]

Seoul National University, Seoul, Korea

Abstract. In the practical use of the Learning With Error (LWE) based cryptosystems, it is quite common to choose the secret to be extremely small: one popular choice is ternary ($\pm 1, 0$) coefficient vector, and some further use ternary vector having only small numbers of nonzero coefficient, what is called *sparse* and ternary vector. This use of small secret also benefits to attack algorithms against LWE, and currently LWE-based cryptosystems including homomorphic encryptions (HE) set parameters based on the attack complexity of those improved attacks.

In this work, we revisit the well-known Howgrave-Graham's hybrid attack, which was originally designed to solve the NTRU problem, with respect to sparse and ternary secret LWE case, and also refine the previous analysis for the hybrid attack in line with LWE setting. Moreover, upon our analysis we estimate attack complexity of the hybrid attack for several LWE parameters. As a result, we argue the currently used HE parameters should be raised to maintain the same security level by considering the hybrid attack; for example, the parameter set $(n, \log q, \sigma) = (65536, 1240, 3.2)$ with Hamming weight of secret key $h = 64$, which was estimated to satisfy ≥ 128 bit-security by the previously considered attacks, is newly estimated to provide only 113 bit-security by the hybrid attack.

Keywords: Lattice-based Cryptography; Learning with Errors; Homomorphic Encryption; The Hybrid Attack

1 Introduction

During the past decades, the *Learning With Errors* (LWE) problem [32] has brought many fruitful applications in the modern cryptographic world; from public key encryptions [6, 19, 31] and digital signatures [5, 21] to homomorphic encryptions (HE) [11,18,20,23]. Informally, the LWE problem asks to distinguish the following two distributions:

$$(\boldsymbol{a}_i, \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + e_i) \quad \text{versus} \quad (\boldsymbol{a}_i, u_i)$$

where $\boldsymbol{s}$ is chosen on some fixed distribution over $\mathbb{Z}_q^n$, and $\boldsymbol{a}_i \in \mathbb{Z}_q^n$ and $u_i \in \mathbb{Z}_q$ are sampled uniformly at random, and the error e_i is sampled from a discrete Gaussian having small width. The originally proposed LWE problem chooses the secret vector $\boldsymbol{s}$ also uniformly over $\mathbb{Z}_q^n$, but several recent constructions restricts the choice of secret for the sake of efficiency.

Also in HE fields, most of implementations including HElib [25], SEAL [33] and HEAAN [17] use *ternary*[1] secret vector. Even more, HEAAN and HElib use *sparse* ternary secret vectors having only a few nonzero components. This use of sparsity is quite serious for *fully* homomorphic encryptions (FHE) which refers to HE supporting literally infinite numbers of operations: All known FHEs are realized by an essential technique so-called *bootstrapping* following Gentry's blueprint [22], and the running time of bootstrapping is highly sensitive to the size of the secret key [13, 16, 24].

However, the use of small secret opens some vulnerabilities, and several attacks have actually been proposed that benefit from such weakness; the smallness [3,9] and the sparsity [2,3] both are known to significantly drop the security level independently. Those attacks [2, 3, 9] are currently considered as the most important attacks for initiating parameters of LWE-based cryptosystems, also including HE implementations.

About this issue, in the *homomorphic encryption standardization* [1], HE community reaches a consensus of using *ternary* secrets while expecting there would be no more significant improvement on ternary secrets. However for the use of *sparse* secrets, it represents some uncertainty by stating

> *"However, we will not present tables for sparse secrets because the security implications of using such sparse secrets is not well understood yet."*

Meanwhile, for another well-known cryptographic hard problem NTRU [28], which also serves as another foundation for past decade lattice-based cryptography. It is well-known that for NTRU, a hybrid of lattice reduction and meet-in-the-middle attack (*the hybrid attack* from now) was proposed by [29], and this is still considered as one of the most powerful attacks for choosing NTRU parameters [27].

1.1 Our contribution

We revisit the hybrid attack in the context of the LWE problem using sparse and ternary secret, together with various techniques derived from other LWE attack literature. We further refine the analysis of the hybrid attack to be align with LWE setting, and derive more accurate and reliable security estimate.

Upon our analysis, we estimate the complexity of the hybrid attack for various parameters currently used in the HE literature. As a result, we observe that the hybrid attack outperforms the previously considered attacks[2] on currently used HE parameter regime by Table 1, which urges parameter update to maintain the same security level.

We finally remark that, our result again confirms that the security implication of the use of sparse secret is not well understood yet, as the homomorphic encryption standardization states.

[1] All entries are in $\{-1, 0, 1\}$ and sparse means the Hamming weight is small

[2] For the previous attack estimation, we exploit LWE-estimator [4].

Table 1: Solving costs for LWE instances with $h = 64$ and $\alpha = 8/q$ where BKZ$_\beta$ cost is measured by $2^{O(\beta)}$ by [10].

Strategy		Dual [2]	Primal [3]	**Hybrid**
n	$\log q$		Bit-security	
2048	45	127.7	135.6	**96.7**
4096	82	129.5	144.4	**102.1**
8192	158	128.6	148.6	**104.9**
16384	350	128.3	140.3	**101.8**
32768	628	127.2	151.3	**109.3**
65536	1240	130.5	153.4	**112.9**

1.2 Discussions

The most direct question would be application of this approach to *non-sparse* ternary secret case, which is much widely used for applications where the bootstrapping is unnecessary. However, for our analysis, it is crucial to know the exact Hamming weight, which makes unnatural to apply our algorithm to uniform ternary secret where we can only guess the Hamming weight. Moreover, although one can artificially assume the Hamming weight $h \approx 2n/3$ to estimate for complexity estimation, we see worse results than the previous results or only tiny speed-ups. Thus we will only consider the sparse secret case in this paper.

As another question, one may wonder about the implication of this attack on other parameter regimes of LWE-based cryptosystems, for example NIST Post-Quantum Cryptography Standardization project [34]. We actually find one public key encryption scheme named `Round5` [7] using sparse and ternary secret, and in addition, it also considered the hybrid attack for the security analysis. However we find their analysis was done in somewhat inaccurate sense which brings too *conservative* parameter setting. In other words, their parameter enjoys the more security level than claimed. As our main focus is HE parameters, we briefly argue this in Appendix A.

1.3 Related Works.

There are two main strategies for solving the LWE problem, called *primal* and *dual*, which differ on the corresponding lattice where the attack investigate. For the primal strategy, [3] gave highly plausible cost estimation with experimental verification. As our target lattice is in an almost same shape to that in the primal strategy, we give a brief review of the primal strategy of [3] in Section 2.4. For the other strategy dual, [2] reported an algorithm based on it, which especially shows better performance for sparse secret than the primal attack for huge parameters used in HE fields. Currently, these two attack algorithms are mainly considered to derive LWE parameters, and their estimation can be easily done by publicly available Sage module [4].

For the hybrid attacks, the seminal work [29] that is originally designed to attack `NTRUencrypt` provided the framework of hybrid of lattice reduction and combinatorics and the analysis framework. Upon this work, [12] firstly applied this to LWE having *binary error*. After then, [36] gave intensive analysis for the hybrid attack and applied the result to several lattice-based schemes related to `NTRU` and LWE. In particular it argued that, in contrast to the common belief, the hybrid attack may not show the best performance for `NTRU`. Aside this fact, our interest LWE variant-*sparse* and small secret LWE- lies outside of its main focus, and it only consider *binary error* LWE in light of rectifying the previous analysis of [12]. As mentioned above, there is NIST Post-Quantum Cryptography Standardization project [34] submission `Round5` [7] considered the hybrid attack to estimate security.

1.4 Roadmap

We give some preliminaries for lattice theory and the previous cost estimation of the primal attack in Section 2. In Section 3, we review the previous hybrid attack and its analysis. Then in Section 4, we give an explicit description of our primal hybrid attack for the LWE problem with small and sparse secret. Finally in Section 5, based on our analysis, we evaluate the bit-security of LWE parameters that are currently used for HE.

2 Preliminaries

We write $\mathbb{Z}_q$ by the set $\mathbb{Z}/q\mathbb{Z}$ whose elements are represented in $(-q/2, q/2] \cap \mathbb{Z}$. We denote the Euclidean norm of vectors by $\|\cdot\|$, and the maximum norm is distinguished by $\|\cdot\|_\infty$. For a set $\mathcal{S}$, we denote a sampling from the uniform distribution over $\mathcal{S}$ by $a \leftarrow \mathcal{S}$.

2.1 The Learning With Errors Problem

Let $n, q > 0$ be integers, $\boldsymbol{s} \in \mathbb{Z}_q^n$ and χ be an error distribution over $\mathbb{Z}$. We define a distribution $\mathcal{L}_{n,q,\chi,\boldsymbol{s}}$ 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 m samples $(\boldsymbol{a}_i, b_i)$ from $L_{n,q,\chi,\boldsymbol{s}}$, we can represent it by a matrix $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m \times (n+1)}$ whose each row corresponds to one sample.

Definition 1 (Learning with Errors). *Let $\mathcal{S}$ be a distribution over $\mathbb{Z}_q^n$, and χ be a small error distribution over $\mathbb{Z}$. The (search) LWE problem, denoted by $\mathsf{LWE}_{n,q,\chi}(\mathcal{S})$, asks to find the secret vector $\boldsymbol{s}$, given polynomially many samples $(\boldsymbol{a}_i, b_i)$ from $L_{n,q,\chi,\boldsymbol{s}}$ for a fixed $\boldsymbol{s} \leftarrow \mathcal{S}$.*

For many cases, the error distribution χ is taken by a discrete Gaussian distribution $\mathcal{D}_{\alpha q}$ of standard deviation $\alpha q/\sqrt{2\pi}$, which case we denote by $\mathsf{LWE}_{n,q,\alpha}(\mathcal{S})$.

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

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

If the dimension n is obvious from the context, we simply write the set by $\mathcal{B}_h$.

2.2 Lattices

A lattice is a discrete additive subgroup of $\mathbb{R}^d$. A full rank matrix $B \in \mathbb{R}^{d \times n}$ is called a basis of a lattice Λ if it holds that

$$\Lambda = \{B\boldsymbol{x} : \boldsymbol{x} \in \mathbb{Z}^n\}.$$

We write $\Lambda(B)$ to represent a lattice determined by basis B. The dimension of a lattice Λ is defined as the cardinality of any basis of Λ. In particular, a lattice in $\mathbb{R}^d$ whose dimension is maximal is called full-rank lattice and without any special mention, we will only consider full-rank lattices throughout this paper.

The fundamental parallelepiped of a lattice basis $B = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_d] \in \mathbb{R}^{d \times d}$ is given by

$$\mathcal{P}(B) = \left\{ \boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{x} = \sum_{i=1}^{d} c_i \boldsymbol{b}_i \text{ for } -1/2 \le c_i < 1/2 \right\}$$

The determinant det of lattice Λ is defined as the d-dimensional volume of its fundamental parallelepiped.

Lattice Reduction Algorithm Lattice reduction algorithm with root-Hermite factor δ_0 returns a short basis, especially whose first vector $\boldsymbol{b}_1$ has size $\le \delta_0^d \cdot \det \Lambda^{1/d}$. The BKZ algorithm [15] is a commonly used lattice reduction algorithm. For inputs d-dimensional basis B of some lattice and *blocksize* β, the BKZ algorithm repeatedly solves the shortest vector problem (SVP) on dimension β blocks obtained from B, and it is known that BKZ terminates after polynomial numbers of SVP solver call. Thus the time complexity of BKZ closely related to the core SVP oracle call, and we will mention the explicit formula in later Section 5. We denote an BKZ algorithm call with blocksize β for a basis T by $\mathsf{BKZ}_\beta(T)$.

Regarding the quality of BKZ algorithm, in [14] it is experimentally verified that BKZ with blocksize β yields root-Hermite factor

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

and we also accept this for our analysis.

There is an useful assumption that estimates the lengths of the Gram-Schmidt vectors of a reduced basis.

Geometric Series Assumption (GSA) Let $B \in \mathbb{Z}^{d\times d}$ be a reduced basis of some full-ranked lattice with root-Hermite factor δ_0 and let $\boldsymbol{b}_i^*$ denote the i-th Gram-Schmidt vectors of B. Then the geometric series assumption (GSA) predicts that the length of $\boldsymbol{b}_i^*$ decreases geometrically. More precisely, GSA predicts $R_i := \|\boldsymbol{b}_i^*\|$ by

$$R_i = \delta_0^{-2(i-1)+d} \cdot \det(\Lambda(B))^{1/d}. \tag{1}$$

2.3 The Nearest Plane Algorithm

We will exploit Babai's nearest plane algorithm [8] (denoted by NP shorthand) in our attack as a subroutine, whose property is summarized as following.

Lemma 2.1 *Let B be a lattice basis and $\boldsymbol{t} \in \mathbb{R}^d$ be a target vector. Then Babai's nearest plane algorithm NP given input B and $\boldsymbol{t}$ returns the unique vector $\boldsymbol{e} = \mathsf{NP}_B(\boldsymbol{t}) \in \mathcal{P}(B^*)$ satisfying $\boldsymbol{t} - \boldsymbol{e} \in \Lambda(B)$, where B^* is the Gram-Schmidt basis of B.*

We denote the output vector by $\mathsf{NP}_B(\boldsymbol{t}) = \boldsymbol{e}$. For the runtime of nearest plane algorithm, we follow the heuristic assumption due to Hirschhorn et al. [26], which says the number of operations T_{NP} of NP algorithm on d-dimensional lattice input is upper bounded by

$$T_{\mathsf{NP}} = d^2/2^{1.06}. \tag{2}$$

For more details on the nearest plane algorithm, we refer Babai's original work [8] or Linder and Peikert's work [31].

2.4 The Primal Lattice Attack

The primal lattice attack for LWE solves the bounded distance decoding (BDD) problem directly. That is, given LWE samples $(A, \boldsymbol{b})$, it finds a vector $\boldsymbol{w} = A\boldsymbol{s}$ such that $\|\boldsymbol{b} - \boldsymbol{w}\|$ is unusually small. The literature has mainly considered two approaches to solve BDD: the first one directly solves BDD using Babai's nearest algorithm followed by lattice reduction [31], and the second one converts the BDD instance into (u)SVP instance, and solves it by lattice reduction [3,6]. We here only explain the second method that is more widely considered. For this method one converts the given LWE samples into some lattice. The *Kannan* embedding [30] considers the column echelon form $[I_n || A'^t]^t$ of $A \in \mathbb{Z}_q^{m\times n}$ (after appropriate permutation of rows) and construct the lattice Λ_{Kan} generated by the following matrix

$$B_{Kan} = \begin{pmatrix} qI_{m-n} & A' & \\ 0 & I_n & \boldsymbol{b} \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{Z}^{(m+1)\times(m+1)}$$

which has a short vector $(\boldsymbol{e}, 1) \in \mathbb{Z}^{m+1}$. However, this approach cannot benefit when the secret is small, which information may lead to better attack by allowing the attacker to exploit it.

In this regard, another lattice embedding is proposed by [9]:

$$\varLambda_{BG} = \{\boldsymbol{x} \in \mathbb{Z}^m \times (\nu\mathbb{Z})^n \times \mathbb{Z}\} : \left(I_m \mid\mid \frac{1}{\nu} A \mid\mid -\boldsymbol{b} \right) \cdot \boldsymbol{x} = \boldsymbol{0} \bmod q\}.$$

This lattice contains an unusual short vector $(\boldsymbol{e}, \nu\boldsymbol{s}, 1)$. Thus, we can find the secret vector $\boldsymbol{s}$ along with error vector $\boldsymbol{e}$ by solving SVP on a lattice generated by basis

$$B_{BG,\nu} = \begin{pmatrix} qI_m & A & -\boldsymbol{b} \\ 0 & \nu I_n & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The scaling factor ν is determined so that the short vector $(\boldsymbol{e}, \nu\boldsymbol{s}, 1)$ is balanced, or explicitly

$$\|\boldsymbol{e}\| \approx \|\nu\boldsymbol{s}\|.$$

Upon the choice of such ν, the vector $(\boldsymbol{e}, \nu\boldsymbol{s}, 1)$ is assumed to be of the form $(\boldsymbol{e}', 1)$ where $\boldsymbol{e}'$ is sampled from Gaussian distribution having same standard deviation with $\boldsymbol{e}$.

Unique-SVP estimate One attack model based on the primal strategy was proposed in [6] and rigorously analyzed in [3]. We remark that, the `usvp` tab of `LWE-estimator` currently considers this attack model. When the BKZ algorithm is applied for a random d-dimensional lattice, the SVP oracle finds the shortest vector of the last projected lattice of size β, whose length is expected to be

$$\delta_0^{2\beta-d} \cdot \det(\varLambda(B))^{1/d}$$

under GSA assumption. Meanwhile, in the embedding lattice for the primal strategy, the projection of $(\boldsymbol{e}, 1)$ to the last β Gram-Schmidt vectors has size

$$\sqrt{\beta/d} \cdot \|(\boldsymbol{e}, 1)\| \approx \sqrt{\beta}\sigma$$

where σ is the standard deviation of each component of $\boldsymbol{e}$. Upon this facts, [3] argues and confirms on an experimental basis that, for β satisfying

$$\sqrt{\beta}\sigma \leq \delta_0^{2\beta-d} \cdot \det(\varLambda(B))^{1/d}, \tag{3}$$

one can totally recover the short vector using BKZ with such β.

Sparse secret case When the secret is further assumed to be sparse, most of columns of A are irrelevant to $\boldsymbol{b} = A\boldsymbol{s} + \boldsymbol{e}$. From this observation, one can randomly remove some columns of $A \in \mathbb{Z}_q^{m\times n}$ to have $A' \in \mathbb{Z}_q^{k\times n}$ $(k < n)$, and then apply the primal strategy to $(A', \boldsymbol{b})$ that requires smaller blocksize β for (3). This succeeds if $(A', \boldsymbol{b})$ is also LWE samples, or equivalently, all the removed columns correspond to zero component of $\boldsymbol{s}$. Note that it happens with adequate probability, say p_k, due to sparsity of the secret. Considering this into account, the attack complexity for sparse secret is calculated by

$$\min_k \frac{1}{p_k} \cdot T_k$$

where T_k is the time cost for the primal attack on k-dimensional LWE sample.

3 The Hybrid Attack for SVP

In this section we recall the description and bird-eye analysis flow of the hybrid attack [36]. Generally, the hybrid attack finds a short vector $\boldsymbol{v} = (\boldsymbol{v}_l, \boldsymbol{v}_g)$ in a lattice Λ, whose basis is of the form

$$B = \begin{pmatrix} T & C \\ 0 & I_r \end{pmatrix} \in \mathbb{Z}^{(d+r)\times(d+r)}.$$

For our interest case, we assume that $\boldsymbol{v}_l$ is sampled from a small Gaussian distribution $\mathcal{D}^d_{\alpha q}$ and $\boldsymbol{v}_g$ is ternary vector having low Hamming weight $h \leq r$.

3.1 Hybrid with Exhaustive-search

The main observation for the hybrid attack is

$$\boldsymbol{v} = \begin{pmatrix} \boldsymbol{v}_l \\ \boldsymbol{v}_g \end{pmatrix} = B \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{v}_g \end{pmatrix} = \begin{pmatrix} T\boldsymbol{x} + C\boldsymbol{v}_g \\ \boldsymbol{v}_g \end{pmatrix}$$

for some $\boldsymbol{x}$. Then we have $\boldsymbol{v}_l = T\boldsymbol{x} + C\boldsymbol{v}_r$, which implies

$$\mathsf{NP}_T(C\boldsymbol{v}_g) = \mathsf{NP}_T(T\boldsymbol{x} + C\boldsymbol{v}_g) = \mathsf{NP}_T(\boldsymbol{v}_l).$$

From this we consider the following hybrid attack of lattice reduction and exhaustive search:

1. Reduce the matrix T so that $\mathsf{NP}_T(\boldsymbol{v}_l) = \boldsymbol{v}_l$
2. Guess $\boldsymbol{v}_r$ and compute $\mathsf{NP}_T(C\boldsymbol{v}_r)$; if the guess is correct, one has unusually short result, namely $\boldsymbol{v}_l$.

The detailed procedure is given below by Algorithm 1.

3.2 Speedup with MitM

Upon this basic attack, one can speed up the guessing step by MitM approach. For two vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ of low weight satisfying $\boldsymbol{v}_g = \boldsymbol{v}_1 + \boldsymbol{v}_2$, we have

$$C\boldsymbol{v}_1 = -C\boldsymbol{v}_2 + C\boldsymbol{v}_g = -C\boldsymbol{v}_2 + \boldsymbol{v}_l - T\boldsymbol{x},$$

and hence

$$\mathsf{NP}_T(C\boldsymbol{v}_1) = \mathsf{NP}_T(-C\boldsymbol{v}_2 + \boldsymbol{v}_l).$$

For MitM strategy, one hopes that the NP algorithm works homomorphically, that is,

$$\mathsf{NP}_T(-C\boldsymbol{v}_2 + \boldsymbol{v}_l) = \mathsf{NP}_T(-C\boldsymbol{v}_2) + \mathsf{NP}_T(\boldsymbol{v}_l) \tag{4}$$

in order to have

$$\mathsf{NP}_T(C\boldsymbol{v}_1) = \mathsf{NP}_T(-C\boldsymbol{v}_2 + \boldsymbol{v}_l) = \mathsf{NP}_T(-C\boldsymbol{v}_2) + \mathsf{NP}_T(\boldsymbol{v}_l).$$

Algorithm 1: A Hybrid of Exhaustive Search

```
Input  : A matrix $B = \begin{pmatrix} T & C \\ 0 & I_r \end{pmatrix} \in \mathbb{Z}^{(d+r)\times(d+r)}$
         A blocksize $\beta$
         A weight parameter $h_g$
         An expected bound $y$ for $\|\boldsymbol{v}_l\|_\infty$.
Output: A short vector $\boldsymbol{v}$ in $\Lambda(B)$
$T \leftarrow \mathsf{BKZ}_\beta(T)$;
for for each $\boldsymbol{w} \in \{\pm 1, 0\}^r$ of Hamming weight $h_g$ do
    $\boldsymbol{v}'_l \leftarrow \mathsf{NP}_T(C\boldsymbol{w}) \in \mathbb{Z}^d$;
    if $\boldsymbol{v} = (\boldsymbol{v}'_l \| \boldsymbol{v}_g) \in \Lambda(B)$ and $\|\boldsymbol{v}_l\|_\infty \leq y$ then
        return $\boldsymbol{v}$.
    end
end
return False
```

As we reduce the matrix T so that $\mathsf{NP}_T(\boldsymbol{v}_l) = \boldsymbol{v}_l$, one reaches

$$\mathsf{NP}_T(C\boldsymbol{v}_1) = \mathsf{NP}_T(-C\boldsymbol{v}_2) + \boldsymbol{v}_l \approx \mathsf{NP}_T(-C\boldsymbol{v}_2) = -\mathsf{NP}_T(C\boldsymbol{v}_2) \tag{5}$$

from which one tries to detect the (noisy) collision in MitM manner. The event (4) definitely not always happens, and indeed the probability for (4) plays a crucial role to analyze the attack complexity.

To detect the collision, we need to store vector $\boldsymbol{v}$ in a table having addresses related to $\mathsf{NP}(C\boldsymbol{v})$. In this regard, we define the address set $\mathcal{A}_{\boldsymbol{x}}$ below: note that for a bound y such that $\|\boldsymbol{v}_l\|_\infty \leq y$, we have

$$\mathcal{A}^{(d,y)}_{\mathsf{NP}_T(C\boldsymbol{v}_1)} \cap \mathcal{A}^{(d,y)}_{-\mathsf{NP}_T(C\boldsymbol{v}_2)} \neq \emptyset,$$

which enables one to find the collision.

Definition 2 (Definition 1 of [36]). *For a vector $\boldsymbol{x} \in \mathbb{Z}^d$ the set $\mathcal{A}^{(d,y)}_{\boldsymbol{x}} \subset \{0,1\}^d$ is defined as*

$$\mathcal{A}^{(d,y)}_{\boldsymbol{x}} = \left\{ \boldsymbol{a} \in \{0,1\}^d : \begin{array}{l} a_i = 1 \; if \; x_i > \lceil \frac{y}{2} - 1 \rceil \\ a_i = 0 \; if \;\; x_i < \lfloor -\frac{y}{2} \rfloor \end{array} \right\}.$$

Algorithm 2 below describes the detail. The main loop investigates vectors of Hamming weight h_M, while expecting $\boldsymbol{v}_g$ is represented by the sum of two vectors of weight h_M. Note that this happens not only for $\mathsf{HW}(\boldsymbol{v}_g) = 2h_M$ case, but $\mathsf{HW}(\boldsymbol{v}_g) = 2k$ for some $k \leq h_M$ case.

Analysis for MitM hybrid The time cost of Algorithm 2 and its main parts consist of the lattice reduction cost T_{BKZ} and the guessing cost T_{guess}. The reduction cost T_{BKZ} can be easily estimated from blocksize β and dimension $d - r$, and hence in the following we mainly focus on T_{guess}.

Algorithm 2: A Hybrid MitM Attack

```
Input  : A matrix $B = \begin{pmatrix} T & C \\ 0 & I_r \end{pmatrix} \in \mathbb{Z}^{(d+r)\times(d+r)}$
         A blocksize $\beta$
         A weight parameter $h_M$
         An expected bound $y$ for $\|\boldsymbol{v}_l\|_\infty$.
Output: A short vector $\boldsymbol{v}$ in $\Lambda(B)$
$T \leftarrow \mathsf{BKZ}_\beta(T)$;
for each $\boldsymbol{w} \in \{\pm 1, 0\}^r$ of Hamming weight $h_M$ do
    $\boldsymbol{v}'_l \leftarrow \mathsf{NP}_T(C\boldsymbol{w}) \in \mathbb{Z}^d$;
    store $\boldsymbol{w}$ in all the boxes having address in a set $\mathcal{A}^{(d,y)}_{\boldsymbol{v}'_l} \cup \mathcal{A}^{(d,y)}_{-\boldsymbol{v}'_l}$;
    for each $\boldsymbol{w}' \neq \boldsymbol{w}$ in all boxes of address in $\mathcal{A}^{(d,y)}_{\boldsymbol{v}'_l} \cup \mathcal{A}^{(d,y)}_{-\boldsymbol{v}'_l}$ do
        $\boldsymbol{v}_g \leftarrow \boldsymbol{w} + \boldsymbol{w}'$ and $\boldsymbol{v}_l \leftarrow \mathsf{NP}_T(C\boldsymbol{v}_g) \in \mathbb{Z}^{d-r}$;
        if $\boldsymbol{v} = (\boldsymbol{v}_l \| \boldsymbol{v}_g) \in \Lambda(B)$ and $\|\boldsymbol{v}_l\|_\infty \leq y$ then
            return $\boldsymbol{v}$.
        end
    end
end
return False
```

We estimate T_{guess} by one inner loop cost multiplied by the expected number of loops, say L, and for the sake of simplicity, we establish the following assumption.

Assumption 3.1 *We assume that one inner loop cost of Algorithm 2 is dominated by nearest plane algorithm cost T_{NP}.*

Explanation. This assumption is closely related to the expected bound y of $\|\boldsymbol{v}_l\|_\infty$: Too small y makes the algorithm fail to find the answer, and too large y increases the size of address set so that Assumption 3.1 fails. We will consider

$$y = 6\frac{\alpha q}{\sqrt{2\pi}},$$

that is 6 times of standard deviation of $\mathcal{D}_{\alpha q}$. Indeed, this value is sufficiently large so that $\|\boldsymbol{v}_l\|_\infty \leq y$ holds with high probability, and sufficiently small so that Assumption 3.1 makes sense. However this is unnecessary for understanding our main contents, so we put the detailed justification for this argument in Appendix C.

From Assumption 3.1, we have $T_{guess} = L \cdot T_{\mathsf{NP}}$ where $T_{\mathsf{NP}} = d^2/2^{1.06}$ according to (2). Toward an estimation for L, we start by defining two sets

$$W = \{\boldsymbol{w} \in \{\pm 1, 0\}^r : \mathsf{HW}(\boldsymbol{w}) = h_M\}$$

and

$$V = \{\boldsymbol{w} \in W : (\boldsymbol{v}_g - \boldsymbol{w} \in W) \wedge (\mathsf{NP}_T(C\boldsymbol{w}) + \mathsf{NP}(C\boldsymbol{v}_g - C\boldsymbol{w}) = \boldsymbol{v}_l)\},$$

and two probabilities

$$p_s := \Pr_{\substack{\boldsymbol{w} \leftarrow W \\ \boldsymbol{v}_l \leftarrow \mathcal{D}_{\alpha q}^d}} [\mathsf{NP}_T(C\boldsymbol{w}) + \mathsf{NP}_T(C\boldsymbol{v}_g - C\boldsymbol{w}) = \boldsymbol{v}_l]$$

and

$$p_c := \Pr_{\boldsymbol{w} \leftarrow W} [\boldsymbol{v}_g - \boldsymbol{w} \in W]$$

for which we make the following assumption.

Assumption 3.2 *We assume that two probabilities p_s and p_c are independent, and further assume that*

$$|V| = p_s p_c |W|.$$

Explanation. We will apply this analysis for the MitM speed-up only when

$$|W| \geq \frac{1}{p_s p_c}.$$

If this inequality is unsatisfied with given parameters, the set V is likely to be empty and Lemma 3.1 becomes vacuous, and hence this analysis for the MitM speed-up becomes utterly improper.

Regarding the set V, the following lemma gives an algorithm terminates condition.

Lemma 3.1 *Algorithm 2 terminates with $\boldsymbol{v}_g$ right after the main loop chooses two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in V$ such that $\boldsymbol{v}_1 + \boldsymbol{v}_2 = \boldsymbol{v}_g$.*

Proof. Since $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ belong to V, we have $\mathsf{NP}_T(C\boldsymbol{v}_1) + \mathsf{NP}(C\boldsymbol{v}_2) = \boldsymbol{v}_l$. Then $\mathsf{NP}_T(C\boldsymbol{v}_1)$ and $-\mathsf{NP}(C\boldsymbol{v}_2)$ differ by $\boldsymbol{v}_l$, and hence from the definition of address set, we have $\mathcal{A}_{\mathsf{NP}(C\boldsymbol{v}_1)} \cap \mathcal{A}_{-\mathsf{NP}(C\boldsymbol{v}_2)} \neq \emptyset$. Thus $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are stored in at least one box, and Algorithm 2 detects them and return $\boldsymbol{v}_g = \boldsymbol{v}_1 + \boldsymbol{v}_2$.

From Assumption 3.2, we expect that the main loop samples one vector in V for every $\frac{1}{p_s p_c}$ repeats, and by Lemma 3.1 we estimate the number of loops are estimated by the birthday paradox as

$$L \approx \frac{\sqrt{|V|}}{p_s p_c} = \sqrt{\frac{|W|}{p_s p_c}} = \sqrt{\frac{2^{h_M} \binom{r}{h_M}}{p_s p_c}}. \tag{6}$$

It remains to compute the probabilities p_s and p_c to completely represent (6) by the parameters d, β, r and h_M. Rather than giving too generalized formula for this, we postpone this later in Section 4.2 after we give the detail for the hybrid attack against LWE case.

4 The Primal-Hybrid attack for LWE

In this section, we apply the hybrid attack algorithm to the primal lattice attack against LWE, and adapt previous analysis in accordance with our interest LWE setting: small and sparse secret with (discrete) Gaussian error. Without any special mention, we assume that LWE sample $(A, \boldsymbol{b})$ is given by $\mathsf{LWE}_{n,q,\alpha}(\mathcal{B}_h)$.

4.1 Overview

Given LWE sample $(A, \boldsymbol{b} = A\boldsymbol{s} + \boldsymbol{e})$, we consider Bai-Gal embedding with some change of the order columns and $\nu = 1$

$$B = \begin{pmatrix} qI_m & -\boldsymbol{b} & A \\ 0 & 1 & 0 \\ 0 & 0 & I_n \end{pmatrix}$$

that contains a short vector $\boldsymbol{v} = (\boldsymbol{e}, 1, \boldsymbol{s})$. By taking a MitM dimension parameter $r \leq n$, we divide the matrix by following:

$$B = \left(\begin{array}{cc|c} qI_m & * & * \\ 0 & I_{n+1-r} & 0 \\ \hline 0 & 0 & I_r \end{array}\right),$$

and parse $\boldsymbol{s} = (\boldsymbol{s}_l, \boldsymbol{s}_g)$ with $\boldsymbol{s}_l \in \mathbb{Z}^d$ and $\boldsymbol{s}_g \in \mathbb{Z}^r$ where $d := m + n + 1 - r$. This represents the short vector $\boldsymbol{v}$ by $(\boldsymbol{v}_l, \boldsymbol{v}_g)$ where $\boldsymbol{v}_l = (\boldsymbol{e}, 1, \boldsymbol{s}_l) \in \mathbb{Z}^d$ and $\boldsymbol{v}_g = \boldsymbol{s}_g$ with $\mathsf{HW}(\boldsymbol{v}_g) \leq h$.

Now one can simply apply Algorithm 2 with $h_M = \lfloor h/2 \rfloor$, but it takes enormous time for the most of our interest parameters. Instead, we pick smaller h_M to have feasible MitM cost, while expecting $\boldsymbol{s}_g$ has smaller weight. Since this naturally introduces some chance that algorithm fails, this parameter h_M would be appropriately chosen to minimize the overall complexity by considering the failure probability. We deal with this probability below by p_{h_M} in Lemma 4.1. The detailed algorithm can be found in Algorithm 3.

Adapting Scaling Factor We also adapt the scaling factor technique [9] to our case. Precisely, we use the following basis

$$B_\nu = \left(\begin{array}{cc|c} qI_m & * & * \\ 0 & \nu I_{n+1-r} & 0 \\ \hline 0 & 0 & I_r \end{array}\right)$$

that contains a vector $(\boldsymbol{v}'_l, \boldsymbol{v}_g)$ with $\boldsymbol{v}'_l = (\boldsymbol{e}, \nu, \nu\boldsymbol{s}_l)$ and $\boldsymbol{v}_g = \boldsymbol{s}_g$. The scaling factor ν is chosen to satisfy $\|\boldsymbol{v}'_l\| \approx \frac{\alpha q}{\sqrt{2\pi}}\sqrt{d}$ in order to assume $\boldsymbol{v}'_l$ as a vector sampled from discrete Gaussian $\mathcal{D}^d_{\alpha q}$. The explicit formula is given by

$$\nu = \frac{\alpha q}{\sqrt{2\pi}} \cdot \sqrt{\frac{n+1-r}{h+1-\mathsf{HW}(\boldsymbol{s}_l)}}.$$

Versus the previous primal attack model We give a brief intuition that explains how the primal hybrid attack performs better than the previous primal attack model. Recall from Section 2.4, the previous model takes advantage of sparsity by reducing the dimension of LWE by removing some columns of A,

Algorithm 3: A Primal Hybrid Attack

```
Input  : $\mathsf{LWE}_{n,q,\alpha}(\mathcal{B}_h)$ sample $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m\times(n+1)}$
         A blocksize $\beta$
         MitM dimension parameter $r$
         MitM weight parameter $h_M$
Output: LWE secret vector $\boldsymbol{s} \in \{\pm1, 0\}^n$
$\nu \leftarrow \frac{\alpha q}{\sqrt{2\pi}} \cdot \sqrt{\frac{n+1}{h-2h_M+1}}$;
$y \leftarrow 6\alpha q/\sqrt{2\pi}$ //   According to Assumption 3.1;
Parse $A' = [-\boldsymbol{b} \mid A]$ into $[A_1'|A_2']$ where $A_2'$ has $r$ columns;
$B_\nu \leftarrow \begin{pmatrix} T & C \\ 0 & I_r \end{pmatrix}$ where $T = \begin{pmatrix} qI_m & A_1' \\ 0 & I_{n+1-r} \end{pmatrix}$ and $C = \begin{pmatrix} A_2' \\ 0 \end{pmatrix}$;
Run Algorithm 2 on input $B_\nu, \beta, h_M, y$.
```

while expecting all the removed columns correspond to zero components of the secret. In our view of dividing

$$B = \left(\begin{array}{cc|c} qI_m & * & * \\ 0 & I_{n+1-r} & 0 \\ \hline 0 & 0 & I_r \end{array}\right) \text{ and } \boldsymbol{v} = \begin{pmatrix} \boldsymbol{v}_l \\ \boldsymbol{v}_g \end{pmatrix},$$

this translates into expecting the vector $\boldsymbol{v}_g = \boldsymbol{s}_g$ is zero, and apply the lattice reduction only for the upper-left matrix. Then the success probability is calculated by the probability that $\boldsymbol{s}_g = \boldsymbol{0}$. In this regard, our hybrid attack can be viewed to admit some nonzero components on $\boldsymbol{v}_g$ as long as the cost for investigating them remains not so large, which results in larger success probability.

4.2 Complexity Analysis

In this section we complete the analysis of hybrid attacks in Section 12 by calculating the probabilities with respect to parameters d, r and so on. We remark that although overall flow of analysis is similar to previous works for hybrid attacks [12, 29, 36], but to the best of our knowledge, our analysis based on the MitM weight parameter h_M and Gaussian shape of $\boldsymbol{v}_l$ has never been considered before.

Recall that we defined

$$W = \{\boldsymbol{w} \in \{\pm1, 0\}^r : \mathsf{HW}(\boldsymbol{w}) = h_M\}$$

and

$$V = \{\boldsymbol{w} \in W : (\boldsymbol{v}_g - \boldsymbol{w} \in W) \wedge (\mathsf{NP}_T(C\boldsymbol{w}) + \mathsf{NP}(C\boldsymbol{v}_g - C\boldsymbol{w}) = \boldsymbol{v}_l)\},$$

and two probabilities

$$p_s := \Pr_{\substack{\boldsymbol{w}\leftarrow W \\ \boldsymbol{v}_l \leftarrow \mathcal{D}_{\alpha q}^d}} [\mathsf{NP}_T(C\boldsymbol{w}) + \mathsf{NP}_T(C\boldsymbol{v}_g - C\boldsymbol{w}) = \boldsymbol{v}_l] \quad (7)$$

and

$$p_c := \Pr_{\boldsymbol{w} \leftarrow W}[\boldsymbol{v}_g - \boldsymbol{w} \in W]. \tag{8}$$

Now we will calculate the probabilities as following:

- Lemma 4.1 calculates the probability p_c under the assumption $\mathsf{HW}(\boldsymbol{v}_g) = 2k$ for some $k \leq h_M$ of probability p_{h_M}
- Lemma 4.2 calculates the probability p_s under the assumption $\mathsf{NP}_T(C\boldsymbol{v}_g) = \boldsymbol{v}_l$ of probability p_{NP}.

Then finally we fully represent T_{BKZ} and T_{guess} with regard to n, q, α, h and β, r, h_M, m and we finally conclude the total complexity estimation

$$T_{tot} = \frac{1}{p_{\mathsf{NP}} p_{h_M}} \left(T_{BKZ} + T_{guess} \right). \tag{9}$$

Lemma 4.1 *Let $\boldsymbol{v}_g \in \mathbb{Z}^r$ be a vector obtained by picking r components of vector $\boldsymbol{v}$ sampled uniformly from $\mathcal{B}_{n,h}$. Then the probability p_{h_M} of $\mathsf{HW}(\boldsymbol{v}_g) = 2k$ for some $k \leq h_M$ is*

$$p_{h_M} = \sum_{k=0}^{h_M} \frac{\binom{h}{2k} \cdot \binom{n-h}{r-2k}}{\binom{n}{r}}.$$

Moreover, conditioned on $\mathsf{HW}(\boldsymbol{v}_g) = 2k$ for some $k \leq h_M$, the probability p_c defined as (8) is represented by

$$p_c = \sum_{k=0}^{h_M} \frac{1}{2^k} \frac{\binom{2k}{k}\binom{r-2k}{h_M-k}}{\binom{r}{h_M}} \cdot \frac{\binom{h}{2k}\binom{n-h}{r-2k}}{\sum_{i=0}^{h_M} \binom{h}{2i}\binom{n-h}{r-2i}}.$$

Proof. The probability p_{h_M} can be directly obtained from

$$\Pr[\mathsf{HW}(\boldsymbol{v}_g) = 2k] = \frac{\binom{h}{2k}\binom{n-h}{r-2k}}{\binom{n}{r}}.$$

For p_c, we write E be the event $\mathsf{HW}(\boldsymbol{v}_g) = 2k$ for some $k \leq h_M$, and split p_c by the conditional probabilities

$$p_c = \sum_{k=0}^{h_M} \Pr_{\boldsymbol{w} \leftarrow W} \left[\boldsymbol{v}_g - \boldsymbol{w} \in W \mid \mathsf{HW}(\boldsymbol{v}_g) = 2k \right] \cdot \Pr \left[\mathsf{HW}(\boldsymbol{v}_g) = 2k \mid E \right].$$

The latter probability is easily obtained by

$$\Pr \left[\mathsf{HW}(\boldsymbol{v}_g) = 2k \mid E \right] = \frac{\binom{h}{2k}\binom{n-h}{r-2k}}{\sum_{i=0}^{h_M} \binom{h}{2i}\binom{n-h}{r-2i}},$$

and we proceed to compute

$$\Pr_{\boldsymbol{w} \leftarrow W} \left[\boldsymbol{v}_g - \boldsymbol{w} \in W \mid \mathsf{HW}(\boldsymbol{v}_g) = 2k \right].$$

For that we observe, in order that $\boldsymbol{v}_g - \boldsymbol{w} \in \{\pm 1, 0\}^r$, $\boldsymbol{w}$ and $\boldsymbol{v}_g$ should agree on every position where $\boldsymbol{w}$ and $\boldsymbol{v}_g$ are both nonzero; if not, $\boldsymbol{v}_g - \boldsymbol{w}$ contains entry 2 or -2. By writing the number of such coincident components by ℓ, we have

$$\mathsf{HW}(\boldsymbol{v}_g - \boldsymbol{w}) = 2k - \ell + (h_M - \ell),$$

and ℓ should be k in order to have $\mathsf{HW}(\boldsymbol{v}_g - \boldsymbol{w}) = h_M$. Therefore, $\boldsymbol{w}$ should coincide with $\boldsymbol{v}_g$ exactly on k nonzero components for $\mathsf{HW}(\boldsymbol{v}_g - \boldsymbol{w}) = h_M$, from which we have

$$\Pr_{\boldsymbol{w} \leftarrow W} [\boldsymbol{v}_g - \boldsymbol{w} \in W \mid \mathsf{HW}(\boldsymbol{v}_g) = 2k] = \frac{1}{2^k} \frac{\binom{2k}{k}\binom{r-2k}{h_M - k}}{\binom{r}{h_M}}.$$

To proceed to the probability p_s and p_{NP} related to nearest plane algorithm, we require the following assumption.

Assumption 4.1 *We assume that the distribution of*

$$C\boldsymbol{w} \mod \mathcal{P}(T^*)$$

for $\boldsymbol{w} \leftarrow W$ is sufficiently close to the uniform distribution on $\mathcal{P}(T^*)$. Moreover, we assume that the discrete Gaussian $\mathcal{D}_{\alpha q}$ behaves like a continuous Gaussian distribution of standard deviation $\alpha q / \sqrt{2\pi}$.

Explanation. The first claim of this assumption has not been exactly stated in any previous analysis, but all of them also explicitly assumed this. For this to be plausible, it would be better to run Algorithm 2 with

$$T' = \begin{pmatrix} * & qI_m \\ \nu I_{n+1-r} & 0 \end{pmatrix},$$

which perturbs the coordinate axes determined by T'^* away from the standard coordinate axes of $C\boldsymbol{w}$. However, for brevity, we just put this by assumption instead of giving too much detail on this.

Lemma 4.2 *Let R_i be the i-th Gram-Schmidt norm of T, and let $\boldsymbol{v}_l$ be a vector sampled from $\mathcal{D}_{\alpha q}^d$. Provided with Assumption 4.1, the probability p_{NP} of $\mathsf{NP}_T(\boldsymbol{v}_l) = \boldsymbol{v}_l$ is*

$$p_{\mathsf{NP}} = \prod_{i=1}^{d} \operatorname{erf}\left(\frac{R_i\sqrt{\pi}}{2\alpha q}\right).$$

Moreover, conditioned on $\mathsf{NP}_T(\boldsymbol{v}_l) = \boldsymbol{v}_l$, we can represent the probability p_s defined as (7) by

$$p_s = \prod_{i=1}^{d} \left(\operatorname{erf}\left(\frac{R_i\sqrt{\pi}}{\alpha q}\right) + \frac{\alpha q}{R_i} \cdot \frac{e^{-\left(\frac{R_i\sqrt{\pi}}{\alpha q}\right)^2} - 1}{\pi} \right).$$

Proof. For readability, we denote $\sigma := \alpha q/\sqrt{2\pi}$. We first compute the probability for $\mathsf{NP}_T(C\boldsymbol{v}_g) = \boldsymbol{v}_l$, or $\mathsf{NP}_T(\boldsymbol{v}_l) = \boldsymbol{v}_l$. By Lemma 2.1, this is equivalent to $\boldsymbol{v}_l \in \mathcal{P}(T^*)$. We assume that $\mathcal{D}^d_{\alpha q}$ is invariant to coordinate axes, we may assume that $\boldsymbol{v}_l$ is sampled with respect to the coordinate axes determined by T^*. Then we have

$$\begin{aligned}\Pr_{\boldsymbol{v}_l \leftarrow \mathcal{D}^d_\sigma}[\boldsymbol{v}_l \in \mathcal{P}(T^*)] &= \prod_{i=1}^{d} \Pr_{e \leftarrow \mathcal{D}_\sigma}[-R_i/2 \le e \le R_i/2] \\ &= \prod_{i=1}^{d} \operatorname{erf}\left(\frac{R_i}{2\sqrt{2}\sigma}\right).\end{aligned}$$

Toward p_s, we first show that

$$\mathsf{NP}_T(C\boldsymbol{w}) + \mathsf{NP}_T(C\boldsymbol{v}_g - C\boldsymbol{w}) = \boldsymbol{v}_l$$

is equivalent to

$$\mathsf{NP}_T(C\boldsymbol{w}) - \boldsymbol{v}_l \in \mathcal{P}(T^*).$$

Since our assumption says $\boldsymbol{v}_l = \mathsf{NP}_T(C\boldsymbol{v}_l) = \mathsf{NP}_T(C\boldsymbol{v}_g)$, and hence we only need to show that

$$\mathsf{NP}_T(C\boldsymbol{w}) + \mathsf{NP}_T(C\boldsymbol{v}_g - C\boldsymbol{w}) = \mathsf{NP}_T(C\boldsymbol{v}_g)$$

is equivalent to

$$\mathsf{NP}_T(C\boldsymbol{w}) - \mathsf{NP}_T(C\boldsymbol{v}_g) \in \mathcal{P}(T^*) :$$

Since $\mathsf{NP}_T(C\boldsymbol{v}_g - C\boldsymbol{w})$ belongs to $\mathcal{P}(T^*)$ by definition, the forward case directly holds. The reverse case also immediately holds because

$$\mathsf{NP}_T(C\boldsymbol{w}) - \mathsf{NP}_T(C\boldsymbol{v}_g) = -\mathsf{NP}_T(C\boldsymbol{v}_g - C\boldsymbol{w}) + T\boldsymbol{x}$$

for some $\boldsymbol{x}$.

Then we can represent

$$\begin{aligned}p_s &= \Pr_{\substack{\boldsymbol{t} \leftarrow \mathcal{P}(T^*) \\ \boldsymbol{e} \leftarrow \mathcal{D}^d_\sigma}} [\boldsymbol{t} + \boldsymbol{e} \in \mathcal{P}(T^*)] \\ &= \prod_{i=1}^{d} \Pr_{\substack{t \leftarrow [-R_i/2, R_i/2] \\ e \leftarrow \mathcal{D}_\sigma}} [t + e \in [-R_i/2, R_i/2]].\end{aligned}$$

We now calculate $p_i := \Pr[-R_i/2 \le t + e \le R_i/2]$. Let $g(z)$ be the probability density function of $t + e$, which can be represented by probability convolution

$$\begin{aligned}g(z) &= \frac{1}{R_i} \cdot \Pr_{e \leftarrow D_\sigma}[z - R_i/2 \le e \le z + R_i/2] \\ &= \frac{1}{2R_i} \cdot \left(\operatorname{erf}\left(\frac{z + R_i/2}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{z - R_i/2}{\sqrt{2}\sigma}\right)\right).\end{aligned}$$

Using the fact $\int \operatorname{erf}(x)dx = x \cdot \operatorname{erf}(x) + \frac{e^{-x^2}}{\sqrt{\pi}} + C$, we reach

$$\begin{aligned} p_i &= \int_{-R_i/2}^{R_i/2} g(z)dz \\ &= \frac{1}{2R_i} \cdot \int_{-R_i/2}^{R_i/2} \operatorname{erf}\left(\frac{z+R_i/2}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{z-R_i/2}{\sqrt{2}\sigma}\right) dz \\ &= \operatorname{erf}\left(\frac{R_i}{\sqrt{2}\sigma}\right) + \frac{\sqrt{2}\sigma}{R_i} \cdot \frac{e^{-\frac{R_i^2}{2\sigma^2}} - 1}{\sqrt{\pi}}. \end{aligned}$$

5 Bit-security estimation

In this section, we estimate the bit-security of LWE with small and sparse secret. Given LWE parameters n, q, α, h we choose optimal algorithm parameters β, r, h_M, m so that the total cost (9)

$$T_{tot} = \frac{1}{p_{\mathsf{NP}} p_{h_M}} \left(T_{BKZ} + T_{guess}\right).$$

is minimized, which determines the bit-security of given LWE parameters. The optimal parameters can be found by investigating possible choices for β, r, h_M, m, and we implement a `Sage` module that finds the (semi-)optimal parameters[3].

We stress again that, our analysis for the MitM hybrid attack is valid only when it holds that $|W| \geq \frac{1}{p_c p_s}$ regarding Assumption 3.2. For the parameters where the opposite case occurs, we estimate the cost with exhaustive search method; we refer Appendix B for detailed cost estimation for the case.

5.1 BKZ cost model

There are two popular choices for BKZ cost model according to core SVP solver; one is from a sieving algorithm [10] and the other from an enumeration algorithm [15]. For blocksize β and dimension d, we assume $T_{BKZ}(\beta, d)$ costs by

- $8d \cdot 2^{0.292\beta + 16.4}$ according to sieving,
- $8d \cdot 2^{0.187\beta \log \beta - 1.019\beta + 16.1}$ according to enumeration.

[3] The optimal parameters can be found by brutally searching all possible choices for β, r, h_M, m but there are too many candidates and hence estimation itself takes too much time. In this regard, we only investigate a plausible range of parameter sets to quickly see the cost estimation, while assuming the optimal point is indeed in our searching scope.

[3] Seems to be a bug, can be reproduced from `LWE-estimator`

5.2 Estimations

Current implementations of `HElib` (commit `5bcae5f`) and `HEAAN` (commit `b45d5f0`) are commonly set sparse ternary secret of Hamming weight $h = 64$, and the noise parameter $\alpha = 8/q$ (yielding standard deviation $\sigma \approx 3.2$) by the default setting. HE-based applications built upon the libraries also use the setting and adjust dimension n and modulus q to reach the desired security level; for example we refer [13, 16, 35]. Thus we estimate attack complexity with the default values for h and α, for several choices of n and q.

We present two cost estimation table: Table 2 is obtained by assuming sieving method for core SVP oracle, and Table 3 assumes enumeration method. Here are some remarks:

- The first row of Table 3 with $n = 2048$ finds its optimal cost with exhaustive search method, and it is marked by $*$ shape. This implies that it is not true that the MitM hybrid strategy always shows the best performance.
- There are other LWE attacks that requires quite many samples of LWE, and hence not suitable for HE where the attack can obtain only n LWE samples. In this regard, we remark that the hybrid attack requires the optimal number of samples m less than n for all cases.

Table 2: Solving costs for LWE instances with $h = 64$ and $\alpha = 8/q$ using sieving SVP oracle

Strategy		Dual [2]	Primal [3]	**Hybrid**	Optimal Parameters			
n	$\log q$	Bit-security			β	r	h_M	m
2048	45	127.7	135.6	**96.7**	186	996	9	1080
4096	82	129.5	144.4	**102.1**	205	1955	9	2167
8192	158	128.6	148.6	**104.9**	197	4038	8	4131
16384	350	128.3	140.3	**101.8**	191	7114	7	9249
32768	628	127.2	151.3	**109.3**	198	15648	7	16797
65536	1240	130.5	153.4	**112.9**	187	32749	6	31847

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 security standard. Tech. rep., HomomorphicEncryption.org, Toronto, Canada (November 2018)
2. Albrecht, M.R.: On dual lattice attacks against small-secret LWE and parameter choices in helib and SEAL. In: Proc. of EUROCRYPT '17. pp. 103–129. Springer (2017)

Table 3: Solving costs for LWE instances with $h = 64$ and $\alpha = 8/q$ using enumeration SVP oracle

Strategy		Dual [2]	Primal [3]	**Hybrid**	Optimal Parameters			
n	$\log q$	Bit-security			β	r	h_M	m
2048	45	130.6	145.2	**107.1***	107	1236	5*	791
4096	82	134.8	152.1	**113.7**	113	2550	5	1546
8192	158	199.8^4	155.2	**114.0**	146	4689	7	3506
16384	350	138.1	146.2	**112.2**	106	9724	4	6630
32768	628	152.9	157.1	**119.3**	128	19510	5	12926
65536	1240	140.4	158.9	**123.9**	116	40507	4	24387

3. Albrecht, M.R., Göpfert, F., Virdia, F., Wunderer, T.: Revisiting the expected cost of solving usvp and applications to lwe. In: Proc. of ASIACRYPT '17. pp. 297–322. Springer (2017)
4. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015)
5. Alkim, E., Barreto, P.S.L.M., Bindel, N., Longa, P., Ricardini, J.E.: The lattice-based digital signature scheme qtesla. Cryptology ePrint Archive, Report 2019/085 (2019), https://eprint.iacr.org/2019/085
6. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange—a new hope. In: Proc. of USENIX Security '16. pp. 327–343. USENIX Association (2016)
7. Baan, H., Bhattacharya, S., Fluhrer, S., Garcia-Morchon, O., Laarhoven, T., Rietman, R., Saarinen, M.J.O., Tolhuizen, L., Zhang, Z.: Round5: Compact and fast post-quantum public-key encryption. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography. pp. 83–102. Springer International Publishing, Cham (2019)
8. Babai, L.: On lovász'lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986)
9. Bai, S., Galbraith, S.D.: Lattice decoding attacks on binary lwe. In: Australasian Conference on Information Security and Privacy. pp. 322–337. Springer (2014)
10. Becker, A., Ducas, L., Gama, N., Laarhoven, T.: New directions in nearest neighbor searching with applications to lattice sieving. In: Proc. of SODA '16. pp. 10–24 (2016)
11. Brakerski, Z., Vaikuntanathan, V., Gentry, C.: Fully homomorphic encryption without bootstrapping. In: Proc. of ITCS'12. Citeseer (2012)
12. Buchmann, J., Göpfert, F., Player, R., Wunderer, T.: 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. pp. 24–43. Springer (2016)
13. Chen, H., Chillotti, I., Song, Y.: Improved bootstrapping for approximate homomorphic encryption. In: Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part II. pp. 34–54 (2019)
14. Chen, Y.: Réduction de réseau et sécurité concrete du chiffrement completement homomorphe. Ph.D. thesis, Paris 7 (2013)

15. Chen, Y., Nguyen, P.Q.: Bkz 2.0: Better lattice security estimates. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 1–20. Springer (2011)
16. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: Bootstrapping for approximate homomorphic encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 360–384. Springer (2018)
17. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: Snucrypto HEAAN. https://github.com/homenc/HElib (2019)
18. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Proc. of ASIACRPYT'17. pp. 409–437. Springer (2017)
19. 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: International Conference on Security and Cryptography for Networks. pp. 160–177. Springer (2018)
20. Chillotti, I., Gama, N., Georgieva, M., Izabachene, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 3–33. Springer (2016)
21. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 238–268 (2018)
22. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proc. of STOC '09. pp. 169–178. ACM (2009)
23. Halevi, S., Shoup, V.: Algorithms in helib. In: Proc. of CRYPTO '14. Springer Verlag (2014)
24. Halevi, S., Shoup, V.: Bootstrapping for helib. In: Annual International conference on the theory and applications of cryptographic techniques. pp. 641–670. Springer (2015)
25. Halevi, S., Shoup, V.: Helib. https://github.com/homenc/HElib (2019)
26. Hirschhorn, P.S., Hoffstein, J., Howgrave-Graham, N., Whyte, W.: Choosing ntruencrypt parameters in light of combined lattice reduction and mitm approaches. In: International Conference on Applied Cryptography and Network Security. pp. 437–455. Springer (2009)
27. Hoffstein, J., Pipher, J., Schanck, J.M., Silverman, J.H., Whyte, W., Zhang, Z.: Choosing parameters for ntruencrypt. In: Cryptographers' Track at the RSA Conference. pp. 3–18. Springer (2017)
28. Hoffstein, J., Pipher, J., Silverman, J.H.: Ntru: A ring-based public key cryptosystem. In: International Algorithmic Number Theory Symposium. pp. 267–288. Springer (1998)
29. Howgrave-Graham, N.: A hybrid lattice-reduction and meet-in-the-middle attack against ntru. Proc. of CRYPTO '07 pp. 150–169 (2007)
30. Kannan, R.: Minkowski's convex body theorem and integer programming. Mathematics of operations research **12**(3), 415–440 (1987)
31. Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. In: Proc. of CT-RSA' 11. vol. 65–58, pp. 319–339. Springer (2011)
32. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proc. of STOC '05. pp. 84–93. ACM (2005)
33. Microsoft SEAL (release 3.3). https://github.com/Microsoft/SEAL (2019), microsoft Research, Redmond, WA.

34. Nist post-quantum cryptography standardization. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography (2019), nIST, Gaithersburg, MD.
35. Tan, B.H.M., Lee, H.T., Wang, H., Ren, S.Q., Aung, K.M.M.: Efficient private comparison queries over encrypted databases using fully homomorphic encryption with finite fields. Cryptology ePrint Archive, Report 2019/332 (2019), https://eprint.iacr.org/2019/332
36. Wunderer, T.: Revisiting the hybrid attack: Improved analysis and refined security estimates. Cryptology ePrint Archive, Report 2016/733 (2016), https://eprint.iacr.org/2016/733

A Application to Round5 PKE scheme

The round 2 candidates of NIST Post-Quantum Cryptography Standardization includes several lattice-based schemes, and we find one scheme named Round5 [7] that uses sparse and ternary secret. The base problem of Round5 is the *learning with rounding* (LWR) problem, defined in similar way to LWE problem with additional modulus $p < q$ and

$$\left(A, \left\lfloor \frac{p}{q} \cdot A\boldsymbol{s} \right\rceil\right) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_p^m$$

It can be viewed that the noise from the rounding plays the Gaussian error role of LWE. Indeed for the security estimation, LWR with modulus p and q is understood by LWE with error having standard deviation

$$\sigma = \frac{q}{p} \cdot \frac{1}{\sqrt{12}},$$

and the typical LWE attacks are applied to estimate its bit-security.

We find that the authors already considered the hybrid attack to choose parameters while conservatively assuming BKZ$_\beta$ cost, regardless of the dimension d of lattice, by

$$T_{BKZ}(\beta, d) = 2^{0.292\beta}$$

according to [6].

According to their analysis, the hybrid attack indeed shows the best performance for its parameter sets. In this regard, we briefly point out here some flaws and insufficiency of their analysis, However, they merely estimate the guessing cost T_{guess} by $\sqrt{N}$ where N is the expected number of candidates of secret vectors, which is quite improper to derive accurate time cost. Moreover, whereas our algorithm introduces a MitM weight parameter h_M to have a trade-off between the success probability and the guessing cost, they only consider the full cost for guessing every possible candidates. Taking this into account, we re-evaluate the bit-security of the proposed parameters according to our refined analysis, and hence conclude that the security of their parameter choice is overestimated

We first remark that, this inferiority of the hybrid attack for Round5 is in line with the argument that the hybrid attack shows worse performance than

Table 4: Solving costs for LWR instances, which were claimed to have $\lambda = 128$ security level in [7], with BKZ cost model $2^{0.292\beta}$ [6].

(Claimed to be) 128 bit-security				
n	$\log q$	h	σ	**Hybrid**
490	10	162	2.29	147.7
508	10	136	2.29	141.8
586	13	182	4.61	146.0
618	11	104	2.29	131.7

previous thought for NTRU, which was stated by [36]. Moreover, the ratio of Hamming weight to the dimension should also be noticed to understand this inferiority compared to HE; `Round5` has weight 162 out of 490 (33%) while `HEAAN` has weight 64 out of from 2048 to 65536 (from 3% to 0.1%), and this may let combinatorial strategy of the hybrid attack bring larger performance gain for the extremely sparse secret of HE.

B Exhaustive-search hybrid

Since the reduction cost is exactly same to Algorithm 2, it only suffices to clarify the guessing cost T_{guess}, which was estimated by $L \cdot T_{\mathsf{NP}}$ with Assumption 3.1 where L is the expected number of loops. For Algorithm 1 with weight parameter h_M, we simply upper bound L by $|W| = 2^{h_M}\binom{r}{h_M}$. Moreover, one can easily check that a sufficient condition for Algorithm 1 success is $\mathsf{NP}_T(\boldsymbol{v}_l) = \boldsymbol{v}_l$ and $\mathsf{HW}(\boldsymbol{v}_g) = h_g$, whose probabilities are denoted by p_{NP} and p_{h_g}. Note that p_{NP} is already computed by Lemma 4.2, and p_{h_g} can be easily computed by

$$p_{h_g} = \frac{\binom{h}{h_g}\binom{n-h}{r-h_g}}{\binom{n}{r}}.$$

Putting together everything, we conclude the total complexity of Algorithm 1 by

$$\frac{1}{p_{\mathsf{NP}} p_{h_g}} \left(T_{BKZ} + T_{guess}\right). \tag{10}$$

where $T_{guess} = 2^{h_M}\binom{r}{h_M} \cdot d^2/2^{1.06}$.

C Error bound choice

For the choice of $y = 6\sigma = 6\alpha q/\sqrt{2\pi}$, we will justify the following for our interest parameters.

- We have
$$\Pr_{\boldsymbol{v}_l \leftarrow \mathcal{D}^d_{\alpha q}} [\|\boldsymbol{v}_l\|_\infty \leq y] \geq 0.99.$$

– For $\boldsymbol{x} \leftarrow \mathcal{P}(T^*)$, the address set $\mathcal{A}_{\boldsymbol{x}}^{(d,y)}$ consists of only one element with overwhelming probability.

For the first claim, note that the probability $\Pr_{e \leftarrow \mathcal{D}_{\alpha q}}[|e| \geq y]$ is about 2^{-28} by approximating the discrete Gaussian as a continuous one. Then $\|\boldsymbol{v}_l\| \leq y$ with probability at least $(1 - d \cdot 2^{-28})$, and since our all parameters satisfy $d \leq 2^{20}$, this is still larger than 0.99.

We now explain the second claim. From the definition, one can check that the number of address set $\mathcal{A}_{\boldsymbol{x}}^{(d,y)}$ is 2^ℓ where ℓ is the number of components of $\boldsymbol{x}$ in $[-\frac{y}{2}, \frac{y}{2}]$. Then for a random choice of $\boldsymbol{x} \leftarrow \mathcal{P}(T^*)$, the probability of x_i is in $[-\frac{y}{2}, \frac{y}{2}]$ is $\frac{y}{R_i}$ where R_i is the i-th Gram-Schmidt length of T. Then we establish an expectation for ℓ by

$$E[\ell] = \sum_{i=1}^{d} \frac{y}{R_i}.$$

By assuming GSA, we have an upper bound for that expectation by

$$\ell \leq d \cdot \frac{y}{R_d} = d \cdot \frac{y}{\delta_0^{-d} \cdot \det(T)^{1/d}}.$$

For all of our parameters in Table 2 and 3 one can check that the right hand side value is much smaller than 1.