Let's Meet Ternary Keys on Babai's Plane: A Hybrid of Lattice-reduction and Meet-LWE

Minki Hhan[1], Jiseung Kim[2], Changmin Lee[1], and Yongha Son[3]

[1] Korea Institute for Advanced Study
{minkihhan, changminlee}@kias.re.kr
[2] Jeonbuk National University
jiseungkim@jbnu.ac.kr
[3] Samsung SDS
yongha.son@samsung.com

Abstract. A cryptographic primitive based on the Learning With Errors (LWE) problem with variants is a promising candidate for the efficient quantum-resistant public key cryptosystem. As the parameters for such cryptosystems are chosen by the concrete attack cost for the corresponding LWE problem, improving LWE solving algorithm has a significant importance.

In this paper, we present a new hybrid attack on the LWE problem. This new attack combines the primal lattice attack and an improved variant of the MitM attack, Meet-LWE, recently suggested by May [Crypto'21]. This resolves the major open problem posed in the same paper.

To this end, we develop several technical tools for hybrid attacks; a new property of Babai's nearest plane algorithm with respect to projection, an approximate variant of Meet-LWE, and a locality-sensitive hashing-based near-collision finding algorithm.

We also present a comprehensive analysis of the proposed attack, which involves the complicated arguments of both lattice and representation techniques. In particular, we take special care in specifying the relevant heuristics and also provide some experimental evidence for them.

We finally estimate the concrete cost of our attack, and observe better cost for sparse LWE/NTRU keys than the other existing attacks. In particular, this result has a direct effect on the currently deployed parameter sets used in some fully homomorphic encryption libraries.

Keywords: LWE, Hybrid lattice attack, Meet-in-the-middle algorithm

1 Introduction

The (search) learning with errors (LWE) problem [44] asks to find the secret key s given $(A, b) \in \mathbb{Z}_q^{m\times(n+1)}$ such that $b = As + e \bmod q$, where e is a small noisy integer vector. The hardness of the LWE problem is well established by the reduction from the worst-case lattice problems, which provides strong confidence in the security of LWE-based cryptosystems. However, this so-called provable security is only guaranteed on a somewhat restricted parameter range, where the schemes have poor efficiency. Thus, most of practical LWE-based schemes

uses other parameters for efficiency far from the provable range. In particular, the size of secret key has a great effect on efficiency, so many LWE-based cryptosystems choose the secret keys as binary/ternary vectors [18, 24, 29] or even sparse vectors [10, 16, 37, 38].

Meanwhile, as the parameters are outside the provable range, the concrete security is measured against known attacks that solves the underlying LWE parameters [7]. Naturally, the use of small secret vector also opens new possibilities in the adversary's view, and the combinatorial attack is one of the most natural approaches to exploit this feature. The Meet-in-the-middle (MitM) strategy due to Odlyzko [33] that finds the secret key in time $S^{0.5}$ has remained the best algorithm for a long time, beyond the naive brute-force search that takes (roughly) S time.

As a recent breakthrough on this line, May [42] proposes a new combinatorial attack called Meet-LWE based on the representation technique, originally introduced in the context of subset sum-type problems [12, 15, 35]. The Meet-LWE attack takes asymptotically S^c time to solve the LWE problem for $c < 0.5$, which achieves $c \approx 0.3$ for some sparse-key based schemes. However, the current practical LWE parameters use a high dimensional vector as a secret key, so the search space size S is extremely large. It turns out that the combinatorial algorithms, even including Meet-LWE, themselves cannot beat the lattice attacks (e.g., [6, 40]) in the practical parameter setting.

The *hybrid* attack refers the attack algorithms that combine lattice reduction algorithms and combinatorial algorithms to take advantage of both attacks. This attack first reduces the dimension of the search space using lattice reduction algorithms; then, the combinatorial strategies are applied to solve the remaining part with a much smaller search space. This line of work has been initiated by the seminal work of Howgrave-Graham [34], which can be understood as a combination of the lattice reduction and the MitM algorithm.

The best-known attack for LWE with sparse ternary keys is presumably the hybrid strategies [34, 43, 45, 47]. As Meet-LWE improves over Odlyzko's MitM, the hybrid attack that combines the lattice reduction algorithm and Meet-LWE is apparently expected to have some implications on the hardness of LWE. This combination was already considered in the original paper of May, but it appears that the direct application of Meet-LWE failed [42, Section 10]. The hybrid attack combining lattice and Meet-LWE remains a central open problem in this landscape of LWE attacks, which is the main subject of this paper.

1.1 Overview of Meet-LWE

We describe a brief review of the original Meet-LWE [42]. Suppose that we are given a matrix $M \in \mathbb{Z}_q^{m \times d}$ and integer q as inputs, and want to find $s \in \mathcal{T}^d(w^{(0)})$ for some $w^{(0)}$ such that $Ms = e \bmod q$ for a unknown small error vector e, where $\mathcal{T}^d(w^{(0)})$ is a set of vectors in $\{0, \pm 1\}^d$ with $w^{(0)}$ nonzero coordinates. Note that LWE problem that asks to find s' from $(A, b = As' + e)$ can be viewed as this problem, by defining $M = (-b|A)$ and $s = (1|s')$.

Meet-in-the-middle idea. To enumerate the sparse secret key $s \in \mathcal{T}^d(w^{(0)})$ efficiently, let us consider the set $S = \left\{ x \mid x \in \mathcal{T}^d(w^{(1)}) \right\}$ for some $w^{(1)} \geq w^{(0)}/2$. It is obvious that there exists two elements $s_1^{(1)}, s_2^{(1)} \in \mathcal{T}^d(w^{(1)})$ such that $s = s_1^{(1)} - s_2^{(1)}$, which is called a weight-$w^{(1)}$ *representation* of s. For such a pair $(s_1^{(1)}, s_2^{(1)})$, it holds that $Ms_1^{(1)} - Ms_2^{(1)} = Ms = e \bmod q$. Since e is a small error, the above equation means that $Ms_1^{(1)}$ and $Ms_2^{(1)}$ are *near-collision*. Thus, we can find s by constructing the set S, then find a representation $(s_1^{(1)}, s_2^{(1)})$ of s using the *near-collision* search in the set $L := \{Ms^{(1)} \bmod q \mid s^{(1)} \in S\}$.

Redundancy and projection constraint. However, note that there are quite many weight-$w^{(1)}$ representations of s, and hence constructing the entire S is an overkill. To minimize this redundancy of representations, Meet-LWE considers the projection map $\pi_r : \mathbb{Z}^m \to \mathbb{Z}^r$ on the first r coordinates to define smaller sets with special *constraints* of the form $\pi_r([Ms^{(1)}]_q) = e$; precisely

$$S_i^{(1)} = \{x \in \mathcal{T}^d(w^{(1)}) \mid \pi_r(Mx \bmod q) = e_i\} \text{ for } i \in \{1, 2\}$$

where e_1, e_2 are proper guesses for $\pi_r(e)$ such that $\pi_r(e) = e_1 - e_2$. The projection dimension r is chosen so that one and only one representation pair survives in $S_1^{(1)} \times S_2^{(1)}$, which suffices to find s.

Error guessing by enumeration. However, the above idea succeeds only when the initial error guessing e_1, e_2 is correct, that is, when it holds that $\pi_r(e) = e_1 - e_2$. Regarding this, the original Meet-LWE simply enumerated all possible error candidates $\pi_r(e)$. As the projection dimension r is set to be a sub-linear number $O(d/\log d)$, this enumeration step can be asymptotically small, making this attack feasible. On the other hand, the concrete complexity for enumeration is definitely not negligibly small, and [42] indeed leaves it as an open question to avoid this error enumeration step. [4]

Extending to higher level. This projection idea can be recursively applied to construct the set $S_*^{(1)}$, using the LWE-like equation $\pi_r(Ms^{(1)} \bmod q) = e_*$. More precisely, each element of $s^{(1)} \in \mathcal{T}^d(w^{(1)})$ also has several representations $s^{(1)} = s_1^{(2)} - s_2^{(2)}$ for $s_1^{(2)}, s_2^{(2)} \in \mathcal{T}^d(w^{(2)})$ where $w^{(2)} \geq w^{(1)}/2$. Then the above constraint idea can be applied again to define smaller sets $S_*^{(2)}$ so that one can recover $S_*^{(1)}$. Finally, the proposed Meet-LWE algorithm comes from a recursive application of this idea, up to level t.

Obstacle against hybrid with lattice. The main idea of the Meet-LWE attack is to reduce the number of representation candidates by using the projection map. However, when applying a similar strategy for the hybrid attack, the projection map does not work well. This is because Babai's Nearest Plane algorithm (NP algorithm hereafter) used for the hybrid attack perturbs the overall

[4] To be honest, Kirshanova and May [36] independently suggested an approach that avoids the error enumeration step. However, we argue their algorithm has some flaw; see Appendix A for details.

axis of the space. In other words, we cannot use the LWE-like equation such as $\pi_r(Ms^{(1)} \bmod q) = e_*$. We refer to [42, Section 10] for a more detailed discussion.

1.2 This work

We present a new hybrid attack, Primal-Meet-LWE, that combines the primal lattice attack and Meet-LWE, resolving the open question of [42]. To achieve this goal, we develop the following technical contributions.

1. *(Babai's nearest plane vs. projection (Section 4))* We show that the projection map to the *last Gram-Schmidt basis* coordinates is well compatible with NP algorithm so that we can define an LWE-like equation on the projected space. This opens a door toward the hybrid attack.
2. *(Meet-LWE with approximate constraints (Section 5))* The projected vectors, however, have real-valued coordinates so that it is infeasible to guess the error by enumeration as in [42]. To deal with this, we consider the concept of constraint to *approximate* one, and successfully extend Meet-LWE to our interest setting.
3. *(Near-collision finding method (Section 3))* Our change to approximate constraints requires to find near-collisions in every level. To have better efficiency on this, we opted for the family of the torus LSH functions [23, 43] with the LSH amplification techniques [32].

We also estimate the concrete cost of Primal-Meet-LWE based on our analysis for the cryptosystems using sparse ternary secret, especially fully homomorphic encryption (FHE) [16, 37, 38], and some post-quantum cryptography (PQC) [13, 18]. According to our estimation, our attack beats all the previous attacks for some currently deployed parameters in FHE library [1, 3], whose sparsity is quite extreme; e.g. 128 out of 2^{15}. However, since PQC literature tends to avoid such a extreme sparsity, our attack falls behind the previous attack (precisely, pure lattice attack [6]). We refer to Section 7 for further details and discussions.

Meanwhile, our analysis of Primal-Meet-LWE requires (variants of) some standard heuristic assumptions from the lattice cryptanalysis and the representation theory, and the average-case behavior of the near-collision finding algorithms. In order to make our analysis more convincing, we put some experimental efforts also, such as experimentally validations of the underlying heuristics, and a proof-of-concept implementation of our newly proposed Meet-LWE with some toy parameters. The details can be found in Appendix C.

1.3 Discussions

Necessity of sparse ternary/binary secrets. In most HE schemes, the hamming weight of the secret key plays a significant role for the performance of so-called *bootstrapping* operations. Although a line of works [37, 38] has reported improvements in the efficiency of bootstrapping even for the non-sparse key [16],

the sparse-key bootstrapping procedures still perform much better in practice and have better asymptotic complexity. Therefore, the use of sparse secret key is still an appealing choice for HE regime, despite of the continuous reports of attacks [21, 22] (including ours).

Issues on the previous hybrid attack. Wunderer [47] and Nguyen [43] pointed out several issues in literature's Howgrave-Graham's hybrid attack [34] estimation and provided refined analysis. Our analysis mostly follows their analysis when required and reflects the issues they pointed out. In particular, [43] pointed out the necessity of rigorous treatment for so-called *admissible* probability. However, it only gives a loose lower and upper bound for this, which is hard to compute the concrete value. As our analysis also needs to consider a similar situation to *admissible* [34, 47] (See Figure 5, for example), we take significant care about this probability on both theoretic and experimental sides, considering the aforementioned points.

Related works. The other major lattice-based attack called *dual* attacks has been significantly improved so far [27,30,41]. In particular, as a concurrent study, a hybrid of dual and Meet-LWE attack is suggested by [14]. Unfortunately, [26] claimed that the success probability of some dual attacks could be overestimated, by raising some concern about the flaw of the assumption.

Glaser and May [28] recently extended Meet-LWE to the non-ternary key cases, and Hoof, Kirshanova, and May [46] show the quantum version of Meet-LWE, respectively. As our Primal-Meet-LWE only considers the original Meet-LWE algorithm, exploring the extended hybrid attacks based on those works should be a fascinating direction.

2 Preliminaries

Notations. For integers n, we denote $[n]$ by the set $\{1, \cdots, n\}$. For a real number a, we write by $[a]_q$ the unique real number in $[-q/2, q/2)$ such that q divides $a - [a]_q$. We write by $\mathcal{C}_\ell^r$ a r-dimensional hypercube of radius ℓ, say $[-\ell, \ell]^r$. The discrete Gaussian distribution with standard deviation σ is denoted by $\mathcal{G}_\sigma$. We also write the uniform distribution over a set S by $\mathcal{U}(S)$, and particularly write $\mathcal{U}(\mathcal{C}_\ell^r)$ by $\mathcal{U}_\ell^r$. For any distribution $\mathcal{D}$, we denote a sampling from $\mathcal{D}$ by $x \leftarrow \mathcal{D}$, but in particular sampling from $\mathcal{U}(S)$ is simply denoted by $x \leftarrow S$.

For a full-rank matrix $B \in \mathbb{R}^{m \times n}$, denote the Gram-Schmidt orthogonalized matrix of B by $B^* = [b_1^* \cdots b_n^*]$, and the normalized matrix of B by $\bar{B}^* = [\bar{b}_1^* \cdots \bar{b}_n^*]$, i.e., $\bar{b}_i^* = b_i^* / \|b_i^*\|$. For any matrix B, we define the fundamental parallelepiped $\mathcal{P}(B) := \{Bx \ : x \in [-1/2, 1/2)^n\}$.

2.1 Learning With Errors and Small Secrets

Definition 1 ((Search) Learning with error problem). *For a secret key s sampled from a distribution $\mathcal{D}_{key}$, the LWE problem asks to find s given a tuple $(A, b = As + e \bmod q) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ for a uniform random matrix A and an error vector e sampled from $\mathcal{D}_{err}$.*

As this paper focuses on the sparse ternary keys, we denote the set of n-dimensional ternary vectors with weight w nonzero entries by

$$\mathcal{T}^d(w) = \{x \in \{\pm 1, 0\}^n \mid s \text{ has } w \text{ nonzero entries}\}$$ [5]

and consider the case of $\mathcal{D}_{key} = \mathcal{U}(\mathcal{T}^d(w))$. The error distribution is usually taken by the discrete Gaussian distribution $\mathcal{G}_\sigma$ over $\mathbb{Z}$ with standard deviation σ, i.e., $\mathcal{D}_{err} = \mathcal{G}_\sigma^m$. The superscript m will be omitted if it is clear from the context. The discrete Gaussian distribution is occasionally approximated by the *continuous* Gaussian distribution with the same standard deviation.

Representations of Secret. Our attack utilizes the representation of a ternary vector.

Definition 2. *For a vector $x \in \mathcal{T}^d(h)$, we call a pair $(x_r, x'_r) \in \mathcal{T}^d(w) \times \mathcal{T}^d(w)$ such that $x = x_r - x'_r$ by a weight-w representation pair, or a w-rep pair of x.*

The following lemma computes the number of w-rep pairs of $x \in \mathcal{T}^d(h)$ from simple combinatorics.

Lemma 1 (Adapted from [45]). *Let h be an even integer and w be an integer such that $w \geq h/2$. For any $x \in \mathcal{T}^d(h)$, the number of w-rep pairs of x is*

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

Proof. See Appendix B.1. □

2.2 Lattices

A lattice is a discrete subgroup of $\mathbb{R}^m$. For a (full-rank) matrix $B = (b_1, ..., b_n) \in \mathbb{R}^{m \times n}$, we denote the lattice $\{Bx \mid x \in \mathbb{Z}^n\}$ by $\mathcal{L}(B)$, and call B as a basis matrix of the lattice $\mathcal{L}(B)$. The Gaussian Heuristic then says that the nonzero shortest element is (approximately) equal to $\sqrt{\frac{n}{2\pi e}} \cdot |\det(B)|^{1/n}$.

The lattice reduction algorithm takes a basis of lattice $\mathcal{L}$ of rank n as input and returns another "better" basis of $\mathcal{L}$. We especially consider the Block-Korine-Zolotarev (BKZ) algorithm [19] that is parameterized by a block size β, denoted by BKZ-β. Assuming the Gaussian Heuristic, the Geometric Series Assumption (GSA) predicts that the Gram-Schmidt norms $\|b_i^*\|$ of the basis B reduced by BKZ-β are estimated as follows

$$\|b_i^*\| = \delta_0^{-2(i-1)+n-1} \cdot \det(B)^{\frac{1}{n}} \tag{1}$$

for $i = 1, ..., n$, where $\delta_0 = \left(\frac{\beta}{2\pi e}(\pi\beta)^{1/\beta}\right)^{1/(2(\beta-1))}$.

Babai's Nearest Plane Algorithm. Let $\mathcal{L}$ be a lattice with the basis $B \in \mathbb{R}^{m \times n}$. Intuitively, Babai's NP algorithm considers a representation of v with

[5] Our definition of $\mathcal{T}^d(w)$ is different from [42], which defines $\mathcal{T}^d(w)$ by the set of vectors having w numbers of 1 and w numbers of -1.

respect to the Gram-Schmidt orthonormal basis $\bar{B}^*$, and then reduces all components into $[-1/2, 1/2)$. Note that we define the output of the algorithm by $t - v$, denoted by $\mathsf{NP}_B(t)$, for later convenience, while the standard NP algorithm aims at finding a lattice point $v \in \mathcal{L}(B)$ such that $t - v \in \mathcal{P}(B^*)$. The following lemma summarizes the classic fact on the output of the NP algorithm.

Lemma 2. *Given an input vector t and basis B, the output e of the Babai's nearest plane algorithm lies in $\mathcal{P}(B^*)$. Further, e is the unique vector in $\mathcal{P}(B^*)$ such that $t - e \in \mathcal{L}(B)$.*

2.3 Primal Hybrid Strategy

All primal hybrid attacks [17,34,45,47] share similar structures; the combination of lattice reduction and combinatorial attack exploiting Babai's nearest plane algorithm, with some differences in targets, analysis, and optimizations. Our attack almost identically follow the first lattice reduction part, and this section review it.

Consider an LWE instance $(A, b = As' + e' \bmod q) \in \mathbb{Z}_q^{m' \times n} \times \mathbb{Z}_q^{m'}$ with a secret key $s' \in \mathcal{T}^n(w)$ and an error vector e' sampled from Gaussian distribution $\mathcal{G}_\sigma$. For ease of representation, we consider an augmented matrix $\bar{M} := (-A||b)$ and $\bar{s} = (s', 1) \in \mathcal{T}^{n+1}(w+1)$ so that the LWE equation $b = As' + e' \bmod q$ is equivalent to $\bar{M}\bar{s} = e' \bmod q$. Then the primal lattice $\mathcal{L}_P$ defined by the following basis matrix $P = \begin{pmatrix} qI_{m'} & \bar{M} \\ & I_{n+1} \end{pmatrix}$. Note that $\mathcal{L}_P$ contains a short vector $(e', \bar{s}) = P \cdot (x, \bar{s})$ for a vector $x \in \mathbb{Z}^{m'}$.

For hybrid attack, we consider some guessing dimension $d \leq n$, and divide the matrix $\bar{M}$ into $\bar{M} = (M_0, M)$ by $n - d + 1$ and d columns, and the secret key $\bar{s}$ into (s_0, s) according to M_0 and M. In this view, we can re-write the basis matrix P by

$$P = \begin{pmatrix} B & M \\ & I_d \end{pmatrix} \text{ where } B = \begin{pmatrix} qI_{m'} & M_0 \\ & I_{n+1-d} \end{pmatrix}. \tag{2}$$

This representation converts $(e', \bar{s}) = P \cdot (x, \bar{s})$ into $Bk + Ms = e$ where $e = (e', s_0)$ and $k = (x, s_0)$. It means that $Ms - e$ is in the lattice $\mathcal{L}(B)$, or equivalently

$$Ms = e \bmod \mathcal{L}(B).$$

For later usage, we formally write this process by Algorithm 1.

After defining the equation $Ms = e \bmod \mathcal{L}(B)$, the primal hybrid attack runs as follows. First, sufficiently reduce the basis matrix B using lattice reduction algorithms so that $\mathsf{NP}_B(Ms) = \mathsf{NP}_B(e) = e$. Second, using combinatorial guessing strategies, finds the partial secret key s such that $\mathsf{NP}_B(Ms) = e$.

The probability of $\mathsf{NP}_B(e) = e$. It should be remarked that the event $\mathsf{NP}_B(e) = e$ does not always happen, even if the basis B is highly reduced. Thus the probability of the event, denoted by p_{NP}, should be taken into account when estimating the attack complexity. The previous works [39, 45, 47] already analyzed this for $\mathcal{G}_\sigma$ by $p_{NP} = \prod_{i=1}^{m} \operatorname{erf}\left(\frac{\|b_i^*\|}{2\sqrt{2}\sigma}\right)$, and we also used this.

Algorithm 1: Primal hybrid conversion

Input: An LWE instance $(A, b) \in \mathbb{Z}_q^{m' \times n} \times \mathbb{Z}_q^{m'}$ s.t. $b = As' + e' \bmod q$, and a guessing dimension $0 < d \leq n$

Output: Matrices $M \in \mathbb{Z}_q^{m \times d}$ and $B \in \mathbb{Z}_q^{m \times m}$ for $m = m' + n - d + 1$, s.t. $Ms = e \bmod \mathcal{L}(B)$ for $e = (e', s_0)$ and $s = (\bar{s}, 1)$ where $s' = (s_0, \bar{s}) \in \mathbb{Z}^{n-d+1} \times \mathbb{Z}^{d-1}$.

1 Set a matrix $\bar{M} := (-A \| b)$

2 Split the matrix $\bar{M}$ into $(M_0, M) \in \mathbb{Z}_q^{m' \times (n-d+1)} \times \mathbb{Z}_q^{m' \times d}$

3 Define a matrix B as $\begin{pmatrix} qI_{m'} & M_0 \\ & I_{n+1-d} \end{pmatrix}$

4 **return** M and B

Normalizing the new error vector. Note that in the new error vector $e = (e', s_0)$ in the equation $\mathsf{NP}_B(Ms) = e$ is in a bit weird shape: the former part e' comes from the error distribution and the latter part s_0 is sparse ternary. We found that no previous work explicitly point out this issue, but simply resolve this by a well-known technique [9] that switches some secret coordinates by other coordinates of the error vector. Using this, we can change the corresponding part of the secret key s_0 into some error vector sampled from Gaussian, and this makes $e = (e', s_0)$ exactly follow $\mathcal{G}_\sigma$.

Remark 1 (Scaling factor normalizing). Many attacks on small secret use an optimization called *scaling factor technique* [11], which uses new parameter $\nu > 0$ and consider an error vector $e_\nu = (e', \nu s_0)$ to optimize the attack cost. This argument is indeed experimentally validated in pure-lattice attack context [6], and reflected to usvp attack in lattice-estimator [2]. However, we found that [45] uses this scaling factor technique in a different context, to assume that $e_\nu = (e', \nu s_0)$ follows Gaussian.[6] Judging that this argument is a bit lack of rigorous validation, we do not follow this despite the benefit of this on attack cost.

3 Near-Collision Finding Algorithm

This section presents a heuristic algorithm to find pairs of two close points among a given set, based on the locality-sensitive hashing (LSH) method [23,32,43]. This algorithm is called the *near-collision finding (NCF) algorithm*, and will be used as a subroutine in our main attack in Section 5.

Throughout this section, we fix $\mathcal{B}$ by a zero-centered r-dimensional hypercube having each coordinate length q_i, namely $\mathcal{B} := \prod_{i=1}^{r} [-q_i/2, q_i/2)$.

Definition 3. *We say that an (ordered) pair $(y_1, y_2) \in \mathcal{B} \times \mathcal{B}$ is an $(\mathcal{B}, \ell)$-near-collision if $\|y_1 - y_2\|_\infty \leq \ell$ and $y_1 \neq y_2$. When the domain $\mathcal{B}$ is clear from the context, we simply say it by an ℓ-near-collision.*

[6] Furthermore, the estimation of previous primal hybrid attack [34] in lattice-estimator [2] is also using this.

Informally speaking, the NCF algorithm is given an input list $L \subset \mathcal{B}$ and asks to find ℓ-near collision pairs in $L \times L$. We present an NCF algorithm that exploits a variant of *the torus LSH* functions, adapting [43] that divides each coordinate interval $[-q_i/2, q_i/2)$ into the equal-length bin of length b.

Definition 4. *Let $n_i := \max\{1, \lfloor q_i/b \rfloor\}$, and $b_i := q_i/n_i$ for each $i \leq [r]$. Then for a block-length parameter b, the family $\mathcal{H}(\mathcal{B}; b)$ of torus LSH functions $h_c : \mathcal{B} \to \prod_{i\in[r]} \mathbb{Z}_{n_i}$ is defined as follows:*

$$\mathcal{H}(\mathcal{B}; b) := \left\{ h_c(y) = \left(\left\lfloor \frac{y_i + c_i}{b_i} \right\rfloor \bmod n_i \right)_{i\in[r]} \;\middle|\; c \in \mathcal{B} \right\}.$$

Our near-collision finding algorithm based on this LSH family proceeds by randomly selecting a hash function h_c from $\mathcal{H}(\mathcal{B}; b)$, and labeling each element y in L with $h_c(y)$. Then it checks whether each pair of elements with the same label is ℓ-near-collision. This procedure is repeated R times with different choices of h_c to find more near-collisions, where R is determined later. Algorithm 2 describes the procedure in details.

Algorithm 2: LSH-based near-collision finding algorithm

Input: A domain $\mathcal{B}$, a list $L \subset \mathcal{B}$, and a norm bound $\ell > 0$
Params : A block-length b, and a repetition number R
Output: A set C of near-collision pairs in $L \times L$

Set a family of torus LSH functions $\mathcal{H} = \mathcal{H}(\mathcal{B}; b)$
Set $C \leftarrow \emptyset$
for $i = 1$ to R **do**
  Sample a function $h_c \leftarrow \mathcal{H}$ uniformly at random
  Set $T \leftarrow \emptyset$
  for each $y \in L$, store y in the bin labelled by $T(h_c(y))$
  for each bin $T(\mathsf{addr})$ where $\mathsf{addr} \in \mathsf{Range}(h_c)$ **do**
    for each pair (y_1, y_2) in $T(\mathsf{addr}) \times T(\mathsf{addr})$ **do**
      if $\|y_1 - y_2\|_\infty \leq \ell$, **then** $C \leftarrow C \cup \{(y_1, y_2)\}$
    end
  end
end
return S

Analysis. In order to analyze Algorithm 2, we need to establish underlying model of the list L. As the first one, we assume that the elements in L are uniformly and independently distributed in the domain $\mathcal{B}$. This assumption is analogous to the well-distributed assumptions used in [12, 15, 35]. Formally, we state the following model.

Model 1 (Purely random model) *Each element in L is uniformly sampled from $\mathcal{U}(\mathcal{B})$ and independent of any other element in L.*

However, in some cases, our target the list L contains an unusually close near-collision which occurrence probability is negligible under the purely random model. Regarding this situation, we introduce another model that assumes the existence of an additional random near-collision pair whose difference follows some (discrete Gaussian) distribution $\mathcal{D}$.

Model 2 (One special near-collision) *The list $L = L(\mathcal{D})$ is the union of a pair $(y_1, y_2) \in \mathcal{B}$ sampled as $y_1 \leftarrow \mathcal{U}(\mathcal{B})$, and $y_2 = y_1 + e$ with $e\mathcal{D}$ and purely random L'. We call the pair (y_1, y_2) by* the special near-collision *from $\mathcal{D}$.*

Finally, the following proposition describes the analysis of Algorithm 2 under the models we established.

Proposition 1. *Let $\mathcal{B} = \prod_{i=1}^{r}[-q_i/2, q_i/2)$ be an r-dimensional domain, $L \subset \mathcal{B}$ be an input list, and ℓ be a norm bound for a near-collision. Let b be a block-length parameter, and let n_i and b_i be defined as Definition 3.*

(Complexity) Let $T_{\mathsf{hash}}(r)$ and $T_{\mathsf{check}}(r)$ denote the time cost of the hash query and of checking if near-collision for the dimension-r vector(s), respectively. Under Model 1 and Model 2, for any $0 < \epsilon < 1$, Algorithm 2 terminates in time

$$R\left(|L| \cdot T_{\mathsf{hash}}(r) + N_{col}(|L|, \epsilon) \cdot T_{\mathsf{check}}(r)\right)$$

where

$$N_{col}(|L|, \epsilon)^{7} = \frac{|L|^2}{2} \cdot \left(\prod_{1 \le i \le r} \frac{1}{n_i}\right) + \sqrt{\frac{R}{\epsilon} \cdot \frac{|L|^2}{2} \cdot \left(\prod_{1 \le i \le r} \frac{1}{n_i}\right)}$$

with probability at least $1 - \epsilon$, and requires $|L|$ words of space for hash function outputs.

(Correctness) Define $p_{lsh}(e) = \prod_{\substack{1 \le i \le r \\ n_i > 1}} \left(1 - \frac{|e_i|}{b_i}\right)$, and

$$p_{ncf}(\ell; \chi) := \Pr_{e \leftarrow \chi}[\|e\|_\infty \le \ell] \cdot \mathop{\mathbb{E}}_{e \leftarrow \chi}\left[1 - (1 - p_{lsh}(e))^R\right]. \tag{3}$$

for any r-dimensional distribution χ. Then

- *Under Model 1, Algorithm 2 finds a random ℓ-near-collision with probability at least $p_{ncf}(\ell; \mathcal{U}([-\ell, \ell]^r))$.*
- *Under Model 2 with the special near-collision from $\mathcal{D}$, Algorithm 2 finds the special near-collision with probability at least $p_{ncf}(\ell; \mathcal{D})$.*

Proof. See Appendix B.2. □

[7] For the parameters appeared in our main attack, the first term dominates $N_{col}(|L|, \epsilon)$ since $|L| \gg R$.

Concrete choice of the repetition number. Although $p_{ncf}(\ell;\chi)$ in Proposition 1 is expressed in a comprehensible form, it is hard to derive the choice of R to make $p_{ncf}(\ell;\chi) \approx 1$, due to the complexity of the integral-based formula

$$\mathbb{E}_{x\leftarrow\chi}[1-(1-p_{lsh}(x))^R] = \int \Pr[\chi = x] \cdot \left(1-(1-p_{lsh}(x))^R\right) dx. \tag{4}$$

Fortunately, we can express at least $\mathbb{E}_\chi[p_{lsh}(x)]$ in the close-form for our interest distribution, e.g., $\chi = \mathcal{U}_\ell$ or $\chi = \mathcal{G}_\sigma$; precisely, for $X = \frac{b}{\sqrt{2}\sigma}$,

$$\mathbb{E}_{x\leftarrow\mathcal{G}_\sigma}[1-x/b] = \mathrm{erf}(X) + \frac{e^{-X^2}-1}{\sqrt{\pi}X} \text{ and } \mathbb{E}_{x\leftarrow\mathcal{U}_\ell}[1-x/b] = \begin{cases} 1-\frac{\ell}{2b} & \text{if } \ell \le b \\ \frac{b}{2\ell} & \text{o.w.} \end{cases}$$

Using this fact, we explicitly set

$$R \ge C_{lsh}/\mathbb{E}_\chi[p_{lsh}(x)] \tag{5}$$

for some $C_{lsh} > 0$, expecting that the following approximation holds:

$$p_{ncf}(\ell;\chi) = \mathbb{E}_\chi[1-(1-p_{lsh}(x))^R] \approx 1-(1-\mathbb{E}_\chi[p_{lsh}(x)])^R. \tag{6}$$

With the choice of R in Eq. (5), the right-hand side is lower bounded by $1-e^{-C_{lsh}}$. However, Eq. (6) is in fact false so we cannot ensure our desired probability is close to $1-e^{-C_{lsh}}$. Regarding this, we experimentally check that the choice of $R \ge C_{lsh}/\mathbb{E}_\chi[p_{lsh}(x)]$ indeed makes the success probability $p_{ncf}(\ell;\chi)$ sufficiently large for our interested parameters b, ℓ and for $C_{lsh} = 10$ in Appendix C.2, and use the values for the attack cost estimation in later Section 7.

4 Matrix Modulus

In this section, we introduce a matrix modulus notion inspired by Babai's NP algorithm, and define a projection map compatible with this matrix modulus notion. This enables us to generalize the projection-based constraint idea of the original Meet-LWE [42] into the primal hybrid case.

Definition 5. *Let $B = (b_1|\cdots|b_n) \in \mathbb{R}^{m\times n}$ be a matrix. For vectors $a, b \in \mathbb{R}^m$, we write $a = b \bmod B$ if $a - b \in \mathcal{L}(B)$.*

We first review some basic properties of the map NP_B. Let $B = (b_1|\cdots|b_n) \in \mathbb{R}^{m\times n}$ be a full-rank matrix with the Gram-Schmidt orthogonal and orthonormal form B^* and $\bar{B}^* = (\bar{b}_1^*|\cdots|\bar{b}_n^*)$, respectively. We define the coordinate vector with respect to $\bar{B}^*$

$$\pi_B : v \mapsto \left(\langle \bar{b}_i^*, v\rangle\right)_{1\le i\le n} \tag{7}$$

with respect to the Gram-Schmidt basis. Since Babai's algorithm does not depend on the choice of basis, for any vector $v \in \mathbb{R}^m$, we have

$$\pi_B(\mathsf{NP}_B(v)) = \mathsf{NP}_{B^*}(\pi_B(v)), \tag{8}$$

which only differ in the order of applying the basis change (from the standard to Gram-Schmidt) and the Babai's nearest plane algorithm.

Definition 6. *Let $B \in \mathbb{R}^{m\times n}$ be a full-rank matrix and $v \in \mathbb{R}^m$. We define*

$$[v]_B := \pi_B(\mathsf{NP}_B(v)) = \mathsf{NP}_{B^*}(\pi_B(v)) \in \prod_{i=1}^{n}\left[-\frac{\|b_i^*\|}{2}, \frac{\|b_i^*\|}{2}\right).$$

For a matrix $M = (M_1|\cdots|M_d) \in \mathbb{R}^{m\times d}$, we define $[M]_B := ([M_1]_B|\cdots|[M_d]_B)$ by the column-wise application of $[\cdot]_B$.

Throughout this paper, we focus on the vectors with coordinates in the Gram-Schmidt basis. Instead of being fully rigorous, we ambiguously write the above output vector and the relevant vectors in the Gram-Schmidt basis by $\mathsf{NP}_B(v)$ or $[v]_B$ if there is no confusion; for example, Lemma 2 implies that

$$[v]_B = \mathsf{NP}_B(v) = v \bmod B,$$

whose explicit meaning is $[v]_B = \pi_B(\mathsf{NP}_B(v)) = \mathsf{NP}_{B^*}(\pi_B(v)) = \pi_B(v) \bmod B^*$.

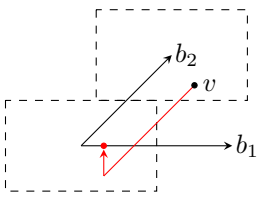
The notation $[\cdot]_B$ satisfies the following useful properties: For two vectors $v, w \in \mathbb{R}^n$ and $c \in \mathbb{R}$, we have

$$[v+w]_B = [v]_B + [w]_B \bmod B, \quad [c\cdot v]_B = c[v]_B \bmod B. \tag{9}$$

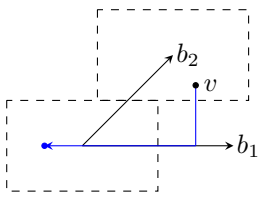
For $s = (s_1, \cdots, s_d) \in \mathbb{Z}^d$ and $M \in \mathbb{R}^{m\times d}$, we also have by the linearity:

$$[Ms]_B = \left[\sum_{i\in[d]} s_i \cdot M_i\right]_B = \sum_{i\in[d]} [s_i \cdot M_i]_B = \sum_{i\in[d]} s_i \cdot [M_i]_B = [M]_B \cdot s. \tag{10}$$

Projection for matrix modulus. To apply the Meet-LWE constraint idea, we need to the projection map $\pi_{B,r}(v)$ that takes some r coordinates from $\pi_B(v)$. However, in general, the r coordinate projection is generally incompatible with modulo B operation, as Figure 1 shows.



(a) Modulo B then project



(b) Project then modulo B

Fig. 1: For a basis $B = (b_1, b_2)$, the dashed line represents each coset modulo B. The projection to the first coordinate and modulo B operation does not commute.

Fortunately, we show the projection map $\pi_{B,r}$ to the *last* r coordinates, precisely

$$\pi_{B,r}(v) := p_r(\pi_B(v)) \text{ for } p_r : (x_1, \cdots, x_n) \mapsto (x_{n-r+1}, \cdots, x_n) \tag{11}$$

solves this issue. Precisely, the following lemma asserts that (a sort of) commutativity between the modulo B and $\pi_{B,r}$ (with the last Gram-Schmidt basis coordinates) holds.

Lemma 3. *Let $B = [b_1, \cdots, b_n] \in \mathbb{R}^{m \times n}$ be a full-rank matrix. Let $\tau_{B,r}(v)$ be the representation of $\pi_{B,r}(v)$ with respect to the standard basis, say $\tau_{B,r} : v \mapsto \sum_{n-r+1 \le i \le n} \langle \bar{b}_i^*, v \rangle \bar{b}_i^*$, and $B_r = [\tau_{B,r}(b_{n-r+1}) | \cdots | \tau_{B,r}(b_n)] \in \mathbb{R}^{m \times r}$. Then it holds that*

$$p_r([v]_B) = [\tau_{B,r}(v)]_{B_r}.$$

Proof. Using QR-decomposition, the matrix B can be represented as a product of two matrices $B = \bar{B}^* \cdot R$. Considering Gram-Schmidt process, the matrix B_r is decomposed by $B_r = \bar{B}_r^* R_r$ where B_r^* is defined as an analogous of B_r for B^*, and R_r is the lower-right submatrix of R of size $r \times r$; as Figure 2. Then Figure 3 represents modulo B operation for v with respect to basis $\bar{B}^*$. The lower red part shows the claim: the left-hand side is $p_r([v]_B)$, and the right-hand side corresponds to $[\tau_{B,r}(v)]_{B_r}$.

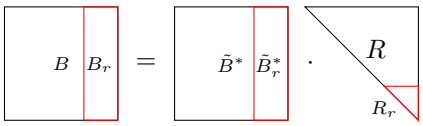


Fig. 2: Gram-Schmidt on B and B_r

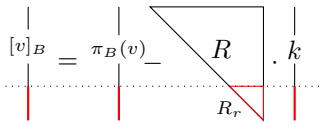


Fig. 3: Modulo B and B_r operations

□

Thanks to Lemma 3, we now able to introduce the notation

$$[v]_{B,r} := p_r([v]_B) = [\tau_{B,r}(v)]_{B_r}. \tag{12}$$

Also note that Lemma 3 directly implies that $[v]_{B,r_1} = [v]_{B,r_2} \bmod B_{r_2}$ for any $r_1 \ge r_2$, which in particular says that $v = [v]_{B,r} \bmod B_r$.

Computation cost of projection. Babai's NP algorithm normally takes $O(n^2)$ operations for the input dimension n. The following lemma provides some computational advantage when the projection dimension r quite smaller than the original attack dimension n.

Lemma 4. *Let $B \in \mathbb{R}^{m \times n}$ be a full-rank matrix. Given the Gram-Schmidt decomposition $B = B^* \cdot R$ and $\pi_B(v)$, the projection $[v]_{B,r}$ can be computed in $O(r^2)$ unit operations.*

Proof. Figure 3 shows that the computation of $[v]_{B,r}$ only depends on the r-dimensional matrix R_r, not the original matrix B or R of the dimension n. Therefore $[v]_{B,r}$ can be computed by a matrix-vector multiplication of dimension r and subtraction of two vectors of dimension r, which can be done in $O(r^2)$ unit (floating point) operations such as addition and multiplication. □

Splitting matrix modulus. The hybrid attack often encounters elements of the form $[v+e]_B$ for some small e such that $[e]_B = e$. Due to the smallness of e, it is expected that

$$[v+e]_B = [v]_B + e$$

where the RHS is coordinate-wise addition with respect to $\bar{B}^*$. However, this is not generally true when any modulo B reduction occurs, so previous hybrid attacks [34, 47] enforced that the small e does not cause modulo B reduction. This is reflected by the probability to the attack cost analysis; the *admissible* definition [34, Definition 3] ([47, Definition 4.2], resp) and the related probability [34, Lemma 6] ([47, Assumption 4], resp).

This condition is slightly mitigated when we consider $[v+e]_{B,r}$, in a way that it only requires no reduction by b_i for $i > n - r$.

Lemma 5. *Let $v, e \in \mathbb{R}^n$, and write $[v]_B = (v_1, \cdots, v_n)$ and $[e]_B = (e_1, \cdots, e_n)$. If $|v_i + e_i| \leq \|b_i^*\|/2$ for every $i \in [n-r+1, n]$, then it holds that*

$$[v+e]_{B,r} = [v]_{B,r} + [e]_{B,r}.$$

Proof. In general, it holds that $[v+e]_B = [v]_B + [e]_B + \sum_{i=1}^{n} c_i \cdot b_i$ for $c_i \in \mathbb{Z}$. Here, the assumption implies that $c_i = 0$ for $i \in [n-r+1, n]$, and we have $[v+e]_B = [v]_B + [e]_B + \sum_{i=1}^{r-1} c_i \cdot b_i$. Since $\pi_{B,r}(b_i) = 0$ for every $i \in [r-1]$, taking $\pi_{B,r}$ on both sides ends the proof. □

5 Meet-LWE for Matrix Modulus

In this section, we describe a generalization of the Meet-LWE algorithm, which finds the solution of the equation $Ms = e \bmod B$. More precisely, given $M \in \mathbb{Z}^{m \times d}$ and $B \in \mathbb{Z}^{m \times m}$, it asks to find $s \in \mathcal{T}^d(w^{(0)})$ such that $[Ms]_B = e$ holds[8] where e is sampled from Gaussian distribution $\mathcal{G}_\sigma$.

Equipped with $\pi_{B,r}$ defined in Section 4, we are now able to *define* the sets of representations with projection-based constraint, as the first step to generalize the Meet-LWE to matrix modulus. One can naively mimic the original Meet-LWE by considering two sets

$$S_i^{(1)} := \{x \in \mathcal{T}^d(w^{(1)}) \mid [Mx]_{B,r} = e_i\} \text{ for } i \in \{1, 2\},$$

while guessing e_1, e_2 such that $\pi_{B,r}(e) = e_1 - e_2$ with $w^{(1)} \geq w^{(0)}/2$. At this point, however, recall that the original Meet-LWE algorithm performs a brute-force guess on the error vector e_i, which was manageable because the number of e_i candidates only depends on r. However, the number of e_i candidates remains almost the same as that of e due to the non-orthogonality of the coordinate system; see Figure 4, which makes this error enumeration infeasible.

[8] This implicitly assumes $[e]_B = e$. The probability of this event is already discussed in Section 2.3, and we assume this holds throughout this section.

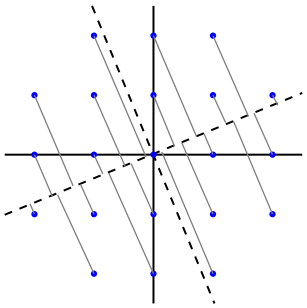


Fig. 4: The blue points denote the integer coordinate points with respect to the standard basis. The dotted line represents the coordinate system determined from $\tilde{B}^*$, and the gray lines show the projection $\tau_{B,r}$.

We solve this problem by defining one level-1 set with *approximate constraints* of the form $\|[Mx]_{B,r}\|_\infty \approx 0$; more precisely,

$$S^{(1)} := \{x \in \mathcal{T}^d(w^{(1)}) \mid \|[Mx]_{B,r^{(1)}}\|_\infty \leq \ell^{(1)}\}.$$

for some small $r^{(1)}$ and $\ell^{(1)}$. By taking this approach, we only need to construct $S^{(1)}$ and then recover the solution s by finding a near-collision in $S^{(1)} \times S^{(1)}$.

However, we need to specify the relevant parameters $r^{(1)}$ and $\ell^{(1)}$ to make this argument substantial. For that, recall from the original Meet-LWE [42] that the purpose of constraint was letting $S_1^{(1)} \times S_2^{(1)}$ contains one and only one rep pair of s, and this was achieved by an proper choice of projection dimension r. As an analogue of this argument for approximate constraint, we need to take proper $r^{(1)}, \ell^{(1)}$ so that $S^{(1)}$ contains one and only one $w^{(1)}$-representation pair $(s_1^{(1)}, s_2^{\prime(1)}) \in S^{(1)} \times S^{(1)}$ of s; Figure 5 graphically explains this. This is indeed the main idea of analysis in the analysis Section 5.2, which requires much complicated argument than the exact constraint case of [42].

Finally, to describe the extension to higher level, observe that the given problem to solve $Ms = e \bmod B$ is almost equivalent to construct

$$S^{(0)} := \{s \in \mathcal{T}^d(w^{(0)}) : \|[Ms]_B\|_\infty \leq \ell^{(0)}\},$$

for some appropriate $\ell^{(0)}$. In this view, the argument above essentially reduces the problem of constructing $S^{(0)}$ into the problem of constructing $S^{(1)}$ and finding near-collisions in $S^{(1)}$. As $S^{(1)}$ is in the exactly same shape with $S^{(0)}$, we again reduce the problem of constructing $S^{(1)}$ to level-2. This can be repeated until arbitrary level-t with the intermediate sets

$$S^{(i)} := \{x \in \mathcal{T}^d(w^{(i)}) \mid \|[Mx]_{B,r^{(i)}}\|_\infty \leq \ell^{(i)}\},$$

where the top level t is chosen so that $|S^{(t)}|$ becomes sufficiently small and it is recoverable by exhaustive search. This ends with an overview of our attack.

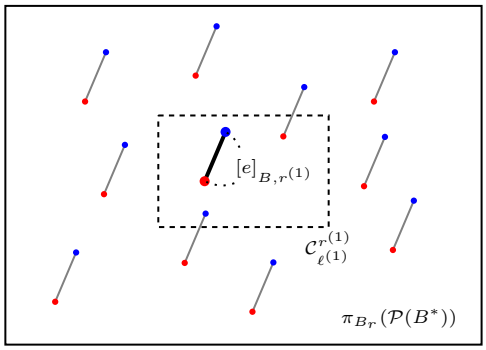


Fig. 5: The blue and red points represent $[Ms_r^{(1)}]_{B,r^{(1)}}$ and $[Ms_r'^{(1)}]_{B,r^{(1)}}$ respectively, which are at distance $[e]_{B,r^{(1)}}$. We expect a representation $(s_1^{(1)}, s_2^{(1)})$ that makes both points have infinity norm less than $\ell^{(1)}$, emphasized by a thick line.

5.1 Full Description with Optimizations

The Algorithm 3 describes the full procedure of our algorithm. Although this algorithm is based on the aforementioned idea, a rigorous analysis needs some more details, so we presented them below.

5.1.1 Algorithm Details.

We present some necessary details for our attack.

The meaning of $\tilde{S}^{(i)}$. In Algorithm 3, we deliberately write the set obtained during the attack by $\tilde{S}^{(i)}$, to distinguish it from $S^{(i)}$. We show that the attack succeeds with a high probability whenever the algorithm recovers a certain fraction $\tilde{S}^{(i)}$ of $S^{(i)}$, which can be efficiently collected through the near-collision finding algorithms. This allows us to avoid recovering the full set $S^{(i)}$, which turns out to be quite burdensome, for example, due to the failure in near-collision finding.

Conditions on $\ell^{(i)}$. The first condition $\ell^{(i)} \leq 2\ell^{(i+1)}$ is to ensure the existence of rep pairs: If $\ell^{(i)} > 2\ell^{(i+1)}$, it holds that $\|[Ms_r]_{B,r^{(i+1)}} - [Ms_r']_{B,r^{(i+1)}}\| \leq 2\ell^{(i+1)} < \ell^{(i)}$ for any $s_r, s_r' \in S^{(i+1)}$, so that some $s^{(i)} \in S^{(i)}$ satisfying $2\ell^{(i+1)} < \|[Ms^{(i)}]_{B,r^{(i+1)}}\|_\infty (\leq \ell^{(i)})$ can never have rep pairs in $S^{(i+1)}$.

For the second condition, observe that we frequently encounter an element of the form $[v+e]_{B,r^{(i)}}$ where $\|[v]_{B,r^{(i)}}\|_\infty \leq \ell^{(i)}$ and $\|[e]_{B,r^{(i)}}\|_\infty \leq \ell^{(i-1)}$. We invoke Lemma 5 for allowing $[v+e]_{B,r^{(i)}} = [v]_{B,r^{(i)}} + [e]_{B,r^{(i)}}$, which requires $\ell^{(i)} + \ell^{(i-1)} \leq \|b^*_{m-r^{(i)}+k}\|/2$ for every $k \in [r^{(i)}]$. Assuming $\|b_i^*\|$ decreases (e.g. GSA), this boils down to the second condition:

$$\ell^{(i)} + \ell^{(i-1)} \leq \|b_m^*\|/2. \tag{13}$$

Near-collision finding. The last $r^{(i)}$ coordinates of $[Mx]_{B,r^{(i-1)}} \in L^{(i)}$ are small due to the constraint of $S^{(i)}$. For the first $r^{(i-1)} - r^{(i)}$ coordinates, we only

Algorithm 3: Meet-LWE for matrix modulus with the top level t

Input: Matrices $M \in \mathbb{Z}^{m\times d}$ and $B \in \mathbb{Z}^{m\times m}$
a norm bound $\ell^{(0)}$ for $e = [Ms]_B$
a solution weight $w^{(0)}$ of $s \in \mathcal{T}^d(w^{(0)})$

Params : Split weights $\{w^{(i)}\}_{i=1}^{t}$ s.t. $w^{(i)}/2 \le w^{(i+1)} < w^{(i)}$,
projection dimensions $\{r^{(i)}\}_{i=0}^{t-1}$ s.t. $r^{(i)} \ge r^{(i+1)}$,
norm bounds $\{\ell^{(i)}\}_{i=0}^{t-1}$ s.t. $\ell^{(i)} \le 2\ell^{(i+1)}$ and $\ell^{(i)} + \ell^{(i-1)} \le \frac{\|b_m^*\|}{2}$,
torus LSH block-lengths $\{b_{lsh}^{(i)}\}_{i=1}^{t}$ s.t. $b_{lsh}^{(i)} \ge \ell^{(i-1)}$,
NCF repetition numbers $\{R_{lsh}^{(i)}\}_{i=1}^{t}$

Output: The solution s such that $\|[Ms]_B\|_\infty \le \ell^{(0)}$.

1 Set the top-level set $\tilde{S}^{(t)}$ by a random subset of $\mathcal{T}^d(w^{(t)})$
2 **for** $i = t$ down to $i = 1$ **do**
  `// Find` $\ell^{(i-1)}$`-near-collisions with respect to` $[Mx]_{B,r^{(i-1)}}$
3 Compute $L^{(i)} = \{[Mx]_{B,r^{(i-1)}} \mid x \in \tilde{S}^{(i)}\}$
4 $C_{S^{(i)}} \leftarrow$ Algorithm 2$(\mathcal{B}^{(i)}, L^{(i)}, \ell^{(i-1)}; b_{lsh}^{(i)}, R_{lsh}^{(i)})$ with $\mathcal{B}^{(i)}$ defined as Eq. (14)
5 Set $\tilde{S}^{(i-1)} = \emptyset$
6 **for** each pair $([Ms_r]_{B,r^{(i-1)}}, [Ms'_r]_{B,r^{(i-1)}}) \in C_{S^{(i)}}$ **do**
    `// Check the weight condition`
7   **if** $s_r - s'_r \in \mathcal{T}^d(w^{(i-1)})$ **then** $\tilde{S}^{(i-1)} \leftarrow \tilde{S}^{(i-1)} \cup \{s_r - s'_r\}$
8 **end**
9 **end**
10 **return** $\tilde{S}^{(0)}$

can use the fact that $[Mx]_B$ lies in $\mathcal{P}(B^*)$. Considering this, the domain $\mathcal{B}^{(i)}$ of the torus LSH is defined by

$$\mathcal{B}^{(i)} := \prod_{j\in[r^{(i-1)}]} \left[-\frac{q_j}{2}, \frac{q_j}{2}\right) \text{ for } q_j = \begin{cases} \|b^*_{m-r^{(i-1)}+j}\| & \text{if } j < r^{(i-1)} - r^{(i)} \\ 2\ell^{(i)} & \text{if } j \ge r^{(i)}. \end{cases} \tag{14}$$

5.1.2 Optimizations. The following optimizations provide concrete improvements to the attack complexity.

Bottom-level optimization. Unlike the upper-level lists, the target near-collision in $L^{(1)} \times L^{(1)}$ is extremely rare. We choose an additional parameter $r^{(0)} \le m$ and consider the last $r^{(0)}$ coordinates (instead of the entire m-dimension) to execute NCF for $L^{(1)}$, whose detailed choice is presented in Remark 3. Asymptotically, the NCF cost for $L^{(1)}$ now only depends on $r^{(0)}$ instead of m. Lemma 4 shows that the cost to compute $[\cdot]_{B,r}$ is $O((r^{(0)})^2)$ floating point operations. This is especially useful for large parameters used in homomorphic encryptions, where m is about ≥ 10000 but $r^{(0)}$ can be chosen as ≤ 100.

Top-level optimization. The top-level set $\tilde{S}^{(t)} = S^{(t)}$ is defined (in Line 1 of Algorithm 3) by a simple subset of $\mathcal{T}^d(w^{(i)})$ *without any constraint* because

there is no way to use the constraints in the top level. Instead, as in the previous works [34, 42], we take $S^{(t)}$ as a *random* subset of $\mathcal{T}^d(w^{(t)})$ of some fixed size, whose detailed choice and analysis are presented in Lemma 6 below. To be compatible with the notation for the intermediate levels such as Eq. (14), we set $r^{(t)}$ by a dummy dimension 0.

Remark 2 (Optimizing the hybrid attack of Howgrave-Graham). Our attack with the top-level $t = 1$ is essentially a reinterpretation of the original Howgrave-Graham's hybrid attack [34], in the sense that both attacks aim to find one representation pair (s_1, s_2) of the desired solution s in a smaller search space $S^{(1)}$, such that $[Ms_1]_B$ and $[Ms_2]_B$ are close enough.

Looking at further details, our top-level $t = 1$ attack retains some optimizations that were not considered in the previous works: the LSH and bottom-level optimizations. First, the original description of the meet-in-the-middle in [34] can also be viewed as utilizing a sort of NCF algorithm. This part is already addressed by the paper [43], replacing this part by (a variant of) Algorithm 2. Furthermore, the optimization that only considers the last $r^{(0)}$ coordinates for the bottom level can also be applied as well. In other words, every computation of $[\cdot]_B$ in Howgrave-Graham's hybrid attack can be replaced by $[\cdot]_{B,r^{(0)}}$ for some $r^{(0)} \leq m$, bringing concrete improvement on the cost of NCF.

5.2 Analysis of Meet-LWE for Matrix Modulus

This section presents the analysis of Algorithm 3, which is probably the most involved part of this paper. The resulting statement is presented in the end of this section as Theorem 1, and this entire section is a proof for that theorem.

5.2.1 Success Probability Overview.

We start from the observation that the attack success condition is $s \in \tilde{S}^{(0)}$, and write the event by $E^{(0)}_{rec}$. Then the success probability p^{meet}_{suc} becomes $\Pr[E^{(0)}_{rec}]$ by definition. As an underlying idea of the level reduction, we consider the following three events that imply $E^{(0)}_{rec}$.

$E^{(0)}_{sp}$: $\exists$ a $w^{(1)}$-rep pair $(s^{(1)}_1, s^{(1)}_2) \in S^{(1)} \times S^{(1)}$ of s, i.e., s *splits* into $s^{(1)}_1 - s^{(1)}_2$.
We say $(s^{(1)}_1, s^{(1)}_2)$ by the level-1 *target pair*, and each $s^{(1)}_k$ by the level-1 *target element* for $k = 1, 2$.
$E^{(1)}_{rec}$: The target elements $s^{(1)}_1, s^{(1)}_2$ are included in $\tilde{S}^{(1)}$ conditioned on $E^{(0)}_{sp}$.
$E^{(0)}_{ncf}$: NCF on $\tilde{S}^{(1)}$ find the target pair $(s^{(1)}_1, s^{(1)}_2)$ conditioned on $E^{(1)}_{rec}$.

This provides

$$p^{meet}_{suc} = \Pr[E^{(0)}_{rec}] \geq \Pr[E^{(0)}_{sp} \wedge E^{(1)}_{rec} \wedge E^{(0)}_{ncf}] = \Pr[E^{(0)}_{sp}] \cdot \Pr[E^{(1)}_{rec}] \cdot \Pr[E^{(0)}_{ncf}]$$

where the last equality holds because of the definitions of $E^{(1)}_{rec}$ and $E^{(0)}_{ncf}$.

We can recursively extend this idea to higher levels for $i \in [t-1]$.

$E_{sp}^{(i)}$: $\exists$ a $w^{(i+1)}$-rep pair $(s_{2k-1}^{(i+1)}, s_{2k}^{(i+1)}) \in S^{(i+1)} \times S^{(i+1)}$ for each level-i target element $s_k^{(i)} \in S^{(i)}$ for $k \in [2^i]$. We say each $(s_{2k-1}^{(i+1)}, s_{2k}^{(i+1)})$ by a level-$(i+1)$ target pair, and each $s_k^{(i+1)}$ by the level-$(i+1)$ target element.

$E_{rec}^{(i+1)}$: All level-$(i+1)$ target elements $s_k^{(i+1)}$ for $k \in [2^{i+1}]$, i.e., $s_k^{(i+1)} \in \tilde{S}^{(i+1)}$ for $k \in [2^{i+1}]$ conditioned on $E_{sp}^{(i)}$.

$E_{ncf}^{(i)}$: NCF on $\tilde{S}^{(i+1)}$ find all level-$(i+1)$ target pairs $(s_{2k-1}^{(i+1)}, s_{2k}^{(i+1)})$ for $k \in [2^i]$ conditioned on $E_{rec}^{(i+1)}$.

Then we have the following inequality

$$\Pr[E_{rec}^{(i)}] \geq \Pr[E_{sp}^{(i)}] \cdot \Pr[E_{rec}^{(i+1)}] \cdot \Pr[E_{ncf}^{(i)}].$$

By applying this inequality for $i = 0, \cdots, t-1$, we have

$$p_{suc}^{meet} \geq \prod_{i=0}^{t-1} \Pr[E_{sp}^{(i)}] \cdot \prod_{i=0}^{t-1} \Pr[E_{ncf}^{(i)}] \cdot \Pr[E_{rec}^{(t)}].$$

Since the top-level set $\tilde{S}^{(t)}$ is directly constructed and is defined by $\tilde{S}^{(t)} = S^{(t)}$, we have $\Pr[E_{rec}^{(t)}] = 1$, and hence finally we have

$$p_{suc}^{meet} \geq \prod_{i=0}^{t-1} \Pr[E_{sp}^{(i)}] \cdot \prod_{i=0}^{t-1} \Pr[E_{ncf}^{(i)}].$$

Therefore, the success probability computation reduces to the computation of the probabilities of E_{sp} and E_{ncf}, what we call by the splitting and near-collision finding probability, respectively.

5.2.2 Computation of Splitting Probability. The goal of this part is Proposition 2 and Lemma 6 at the end, which computes $E_{sp}^{(i)}$ for intermediate levels $i < t-1$ and $i = t-1$, respectively.

Intermediate level. We consider the partial event $E_{sp,k}^{(i)}$ that the k-th level-i target element $s_k^{(i)} \in S^{(i)}$ splits into $S^{(i+1)}$. This event means that there is a $w^{(i+1)}$-rep pair $(s_{2k-1}^{(i+1)}, s_{2k}^{(i+1)})$ of $s_k^{(i)}$ such that

$$([Ms_{2k-1}^{(i+1)}]_{B,r^{(i+1)}}, [Ms_{2k}^{(i+1)}]_{B,r^{(i+1)}}) \in \mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}} \times \mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}} \tag{15}$$

See again Figure 5 for a graphical illustration of this event.

Clearly, $\Pr[E_{sp}^{(i)}] = \Pr[\wedge_{k \in [2^i]} E_{sp,k}^{(i)}]$ holds by definition. However, the target elements $s_k^{(i)}$ are tied by some relation, for example,

$$s_k^{(i-2)} = (s_{4k-3}^{(i)} - s_{4k-2}^{(i)}) - (s_{4k-1}^{(i)} - s_{4k}^{(i)}), \tag{16}$$

which makes hard to compute the probability $\Pr[\wedge_{k \in [2^i]} E_{sp,k}^{(i)}]$. So we establish the following heuristic that makes the computation feasible.

Heuristic 1 (Independency) *It holds that* $\Pr[E_{sp}^{(i)}] \geq \prod_{k\in[2^i]} \Pr[E_{sp,k}^{(i)}]$.

Proof. See Appendix B.3 for justification, and Appendix C.3 for relevant experiments. □

We now proceed to compute each probability $\Pr[E_{sp,k}^{(i)}]$ of the splitting event occurs for $s_k^{(i)} \in S^{(i)}$. Let $P(s_k^{(i)})$ be the set

$$\left\{ ([Ms_r]_{B,r^{(i+1)}}, [Ms'_r]_{B,r^{(i+1)}}) \;\middle|\; (s_r, s'_r) \text{ is a } w^{(i+1)}\text{-rep pair of } s^{(i)} \right\},$$

whose size is the number of $w^{(i+1)}$-rep pairs of $s^{(i)}$, say $R^{(i)} := R(d, w^{(i+1)}, w^{(i)})$ that can be computed by Lemma 1. Then the event $E_{sp,k}^{(i)}$ means that at least one pair in $P(s_k^{(i)})$ lies in $\mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}}$. To analyze the probability, we need to specify the distribution of $P(s_k^{(i)})$. Since the second element is automatically determined by the first element $x := [Ms_r]_{B,r^{(i+1)}}$ as $[x+e]_{B,r^{(i+1)}}$ where $e = [Ms_k^{(i)}]_{B,r^{(i+1)}}$, we only regard the first element.

Heuristic 2 (Uniformity) *We model the set*

$$P_1(s_k^{(i)}) := \left\{ [Ms_r]_{B,r^{(i+1)}} \;\middle|\; (s_r, s'_r) \textit{ is a } w^{(i+1)}\textit{-rep pair of } s^{(i)} \right\}$$

as follows:

- *Let* $e = [Ms_k^{(i)}]_{B,r^{(i+1)}}$*, and repeat the following* $R^{(i)}/2$ *times:*
- *Sample* $x \leftarrow \mathcal{P}(B^*_{r^{(i+1)}})$*, and insert* x *and* $([-x-e]_{B_{r^{(i+1)}}})$ *into* $P(s_k^{(i)})$.

Proof. See Appendix B.4 for justification, and Appendix C.1 for relevant experiments.

Under this heuristic, we compute $\Pr[E_{sp,k}^{(i)}]$ and then $\Pr[E_{sp}^{(i)}]$ as follows.

Proposition 2. *Assume Heuristic 2 holds for* $0 \leq i < t-1$ *and Heuristic 1 holds for* $1 \leq i < t-1$*. Then for* $0 \leq i < t-1$*,*

$$\Pr[E_{sp,k}^{(i)}] = p_{sp}(\chi_{sp}^{(i)}) := \mathop{\mathbb{E}}_{e\leftarrow\chi_{sp}^{(i)}} \left[1 - (1 - p_{rep}(e))^{R^{(i)}/2}\right]. \tag{17}$$

where

$$\chi_{sp}^{(i)} = \begin{cases} \mathcal{G}_\sigma^{r^{(1)}} & \text{if } i = 0 \\ \mathcal{U}_{\ell^{(i)}}^{r^{(i+1)}} & \text{otherwise,} \end{cases} \qquad p_{rep}(e_1, \cdots, e_{r^{(i+1)}}) = \prod_{j\in[r^{(i+1)}]} \frac{2\ell^{(i+1)} - |e_j|}{\|b^*_{m-r^{(i+1)}+j}\|}.$$

Further assuming Heuristic 1, we have $\Pr[E_{sp}^{(i)}] \geq \left(p_{sp}(\chi_{sp}^{(i)})\right)^{2^i}$ *for* $0 \leq i < t-1$.

Proof. See Appendix B.5. □

Top-level split. For the top-level set $S^{(t)}$ having no constraint, the analysis becomes much simpler.

Lemma 6. $\Pr[E_{sp}^{(t-1)}] \geq \left(1 - e^{-R^{(t-1)} \cdot |S^{(t)}|^2 / |\mathcal{T}^d(w^{(t)})|^2}\right)^{2^{t-1}}$. *In particular, if* $|S^{(t)}| = \sqrt{3} \cdot \frac{|\mathcal{T}^d(w^{(t)})|}{\sqrt{R^{(t-1)}}}$, *we have* $\Pr[E_{sp}^{(t-1)}] \geq (1 - e^{-3})^{2^{t-1}} \geq 0.95^{2^{t-1}}$.

Proof. See Appendix B.6. □

5.2.3 Computation of Near-collision Finding Probability. The overall story is similar to the $E_{sp}^{(i)}$: We also consider a joint probability $\Pr[\wedge_{k \in [2^i]} E_{ncf,k}^{(i)}]$ where $E_{ncf,k}^{(i)}$ expects the level-$(i+1)$ target pair $(s_{2k-1}^{(i+1)}, s_{2k}^{(i+1)})$ is found by NCF on $\tilde{S}^{(i+1)}$. Similarly to $E_{sp}^{(i)}$, we establish Heuristic 3 to lower bounds the probability, and Heuristic 4 to uniformly model the target list.

Heuristic 3 (Independency) *It holds that* $\Pr[E_{ncf}^{(i)}] \geq \prod_{k \in [2^i]} \Pr[E_{ncf,k}^{(i)}]$.

Proof. See Appendix B.7 for justification, and Appendix C.3 for relevant experiments. □

Heuristic 4 (Uniformity) *We model the list*

$$L^{(i+1)} = \{[Mx]_{B,r^{(i)}} \mid x \in \tilde{S}^{(i+1)}\} \subset \mathcal{B}^{(i+1)}$$

as follows:

- *For* $i \geq 1$, $L^{(i+1)}$ *follows Model 1 (purely random model)*
- $L^{(1)}$ *follows Model 2 (one special near-collision model) with* $\mathcal{D} = \mathcal{G}_\sigma$.

Proof. See Appendix B.8 for justification, and Appendix C.1 for relevant experiments. □

Under these heuristics, we compute $\Pr[E_{ncf,k}^{(i)}]$ and $\Pr[E_{ncf}^{(i)}]$ as follows.

Proposition 3. *Assume Heuristics 3 and 4. Let*

$$\chi_{ncf}^{(i)} = \begin{cases} \mathcal{G}_\sigma^{r^{(0)}} & \text{if } i = 0 \\ \mathcal{U}_{\ell^{(i)}}^{r^{(i)}} & \text{otherwise,} \end{cases} \quad \text{and } p_{nc}(\chi_{ncf}^{(i)}) := \mathop{\mathbb{E}}_{d \leftarrow \chi_{ncf}^{(i)}} \left[\prod_{j=r^{(i)}+1}^{r^{(i+1)}} \left(1 - \frac{|d_{j-r^{(i)}}|}{\|b_{m-j}^*\|}\right) \right]$$

for $0 \leq i \leq t-1$. *Let* $p_{ncf}(\ell; \chi)$ *be computed as in Proposition 1. Then the following holds for* $0 \leq i \leq t-1$*:*

$$\Pr[E_{ncf}^{(i)}] \geq \left(p_{nc}(\chi_{ncf}^{(i)}) \cdot p_{ncf}(\ell^{(i)}; \chi_{ncf}^{(i)})\right)^{2^i}.$$

Proof. See Appendix B.9. □

Complexity analysis. We then proceed to the cost analysis. First, we compute the cost of each i-th level of the main loop of Algorithm 3.

Lemma 7. *For any $0 < \epsilon < 1$, the time cost $T_{loop}^{(i)}$ of the i-th level of the main loop of Algorithm 3 is dominated by*

$$R_{lsh}^{(i)} \cdot \left(|\tilde{S}^{(i)}| \cdot T_{\mathsf{hash}}(r^{(i-1)}) + N_{col}^{(i)}(\epsilon) \cdot (T_{\mathsf{check}}(r^{(i-1)}) + 2w^{(i)}) \right)$$

with probability at least $1 - \epsilon$, where $N_{col}^{(i)}(\epsilon)$ is defined as Proposition 1.

Proof. See Appendix B.10 □

The complexity of the overall algorithm is the summation of the cost of each level, which is bounded well with probability $1 - t\epsilon$ by the union bound.

For the concrete value of $|\tilde{S}^{(i)}|$, we use a trivial bound $|S^{(i)}|$. The intermediate level size can be easily estimated by volume ratio:

$$|S^{(i)}| = |\mathcal{T}^d(w^{(i)})| \cdot \frac{\mathrm{vol}(\mathcal{C}_{\ell^{(i)}}^{r^{(i)}})}{\mathrm{vol}(\mathcal{P}(B_{r^{(i)}}^*))} = \binom{d}{w^{(i)}} 2^{w^{(i)}} \cdot \prod_{j \in [r^{(i)}]} \frac{2\ell^{(i)}}{\|b_{m-r^{(i)}+j}^*\|}. \quad (18)$$

As mentioned in top-level optimization, the top-level list size is fixed by $|S^{(t)}| = \frac{\sqrt{3}|\mathcal{T}^d(w^{(t)})|}{\sqrt{R^{(t-1)}}}$.

Full Theorem Statement. Putting everything together, we have the full theorem that analyzes Algorithm 3. Note that again the contents in this section so far is a proof for this theorem.

Theorem 1. *Let $M \in \mathbb{Z}^{m \times d}$ and $B \in \mathbb{Z}^{m \times m}$ be matrices, $\ell^{(0)} > 0$ be a real number, $w^{(0)}$ be a positive integer. Suppose that there exists a unique $s \in \mathcal{T}^d(w^{(0)})$ such that $\|[Ms]_B\|_\infty \le \ell^{(0)}$ holds. Let t be a positive integer, and let*

$$\{w^{(i)}\}_{i=1}^t, \{r^{(i)}\}_{i=0}^{t-1}, \{\ell^{(i)}\}_{i=0}^{t-1}, \{b_{lsh}^{(i)}\}_{i=1}^t, \{R_{lsh}^{(i)}\}_{i=1}^t$$

be the set of parameters satisfying the conditions in Algorithm 3. Assuming that Heuristics 1 to 4, the followings hold.

(Complexity) For any $0 < \epsilon < 1$, the running time of Algorithm 3 is dominated[9]

$$\sum_{i=0}^{t} R_{lsh}^{(i)} \cdot \left(|\tilde{S}^{(i)}| \cdot T_{\mathsf{hash}}(r^{(i-1)}) + N_{col}^{(i)}(|\tilde{S}^{(i)}|, \epsilon) \cdot (T_{\mathsf{check}}(r^{(i-1)}) + 2w^{(i)}) \right) \quad (19)$$

with probability at least $1 - t \cdot \epsilon$, where $N_{col}^{(i)}(|\tilde{S}^{(i)}|, \epsilon)$ is defined as Proposition 1 and

$$|\tilde{S}^{(i)}| \le \begin{cases} \frac{\sqrt{3}|\mathcal{T}^d(w^{(t)})|}{\sqrt{\binom{w^{(t)}}{w^{(t)}/2} \cdot \binom{d-w^{(t)}}{w^{(t-1)}-w^{(t)}/2} \cdot 2^{w-h/2}}} & \text{for } i = t, \\ \binom{d}{w^{(i)}} 2^{w^{(i)}} \cdot \prod_{j \in [r^{(i)}]} \frac{2\ell^{(i)}}{\|b_{m-r^{(i)}+j}^*\|} & \text{otherwise.} \end{cases}$$

[9] *Dominated by T* means that it is less than (say) $1.001T$ for practical parameters.

(Correctness) Let $p_{suc}^{meet} := \Pr[s \in \tilde{S}^{(0)}]$ *be the probability that Algorithm 3 finds the solution* s*. It holds that*

$$p_{suc}^{meet} \geq \prod_{i=0}^{t-1} \Pr[E_{sp}^{(i)}] \cdot \prod_{i=0}^{t-1} \Pr[E_{ncf}^{(i)}] \tag{20}$$

where the explicit formulas for $\Pr[E_{sp}^{(i)}]$ *and* $\Pr[E_{ncf}^{(i)}]$ *are given in Proposition 2 and Proposition 3, respectively.*

Remark 3 (Choice of bottom-level dimension $r^{(0)}$*).* Observe that when $r^{(0)}$ gets bigger, $N_{col}^{(1)}$ becomes smaller and $R_{lsh}^{(1)}$ becomes larger. Using this fact, we choose $r^{(0)}$ by the optimal one (between $r^{(1)}$ and m) that minimize $T_{loop}^{(1)}$.

Computation of $p_{sp}(\chi_{sp}^{(i)})$ **and the choice of** $r^{(i)}$**.** Note that $\Pr[E_{sp}^{(t-1)}]$ can be concisely computed by $0.95^{2^{t-1}}$ by fixing only the top-level set size $|S^{(t)}|$, as Lemma 6. Meanwhile, for the intermediate levels probability in Proposition 2, the exact computation of $p_{sp}(\chi_{sp}^{(i)}) := \mathbb{E}_{\chi_{sp}^{(i)}}\left[1-(1-p_{rep}(x))^{R^{(i)}/2}\right]$ involves too many parameters, so it is hard to expect such simplification. This is really cumbersome for the concrete attack cost estimation, so we consider a heuristic approach that enables a simple computation of $p_{sp}(\chi_{sp}^{(i)})$.

Observe that the shape of $p_{sp}(\chi_{sp}^{(i)})$ is almost similar to Eq. (4), so we again take a similar approach. Precisely, we choose the maximal $r^{(i+1)}(\leq r^{(i)})$ so that

$$\frac{R^{(i)}}{2} \geq \frac{C_{proj}}{\mathbb{E}_{\chi_{sp}^{(i)}}[p_{rep}(x)]} \tag{21}$$

for some constant $C_{proj} > 0$, since $\mathbb{E}_{\chi_{sp}^{(i)}}$ can be expressed in a closed-form again. Similar to Eq. (5), the underlying rationale of this choice is an expectation that

$$p_{sp}(\chi^{(i)}) := \mathbb{E}_{\chi_{sp}^{(i)}}[(1-(1-p_{rep}(x))^{R^{(i)}/2}] \approx 1-\left(1-\mathbb{E}_{\chi_{sp}^{(i)}}[p_{rep}(x)]\right)^{R^{(i)}/2}, \tag{22}$$

where our choice of $r^{(i)}$ lower bounds the right-hand side by $1-e^{-C_{proj}}$. Again, Eq. (22) is in fact false, so we experimentally verify that the probability $p_{sp}(\chi^{(i)})$ is sufficiently large for our choice of $r^{(i)}$ and $C_{proj} = 10$ in Appendix C.2, and use the values for the attack cost estimation in later Section 7.

6 Primal-Meet-LWE

This section proposes our new hybrid attack that combines the primal lattice conversion Algorithm 1) and Meet-LWE for matrix modulus (Algorithm 3). Note that, although we convert the LWE equation $b = As' + e' \bmod q$ into $Ms = e \bmod B$, we cannot know the exact hamming weight of the partial guessing secret s. In this regard, we introduce a guess weight parameter $w^{(0)}$ while expecting $HW(s) = w^{(0)}$. The formal description is given in Algorithm 4.

Algorithm 4: PRIMAL-MEET-LWE

Input: LWE instances $(A, b) \in \mathbb{Z}_q^{m' \times (n+1)}$ such that $b = As' + e$ where $e \leftarrow \mathcal{G}_\sigma$ and $s' \leftarrow \mathcal{T}^n(h)$, and a top-level parameter t

Params : β: the block size for the BKZ reduction
d: the guessing dimension
$\mathsf{meet\text{-}lwe\text{-}params} = \{(w^{(i)}, r^{(i)}, \ell^{(i)}, b_{lsh}^{(i)}, R_{lsh}^{(i)})\}_i$ satisfying the conditions specified in Algorithm 3

Output: The secret key $s' \in \mathcal{T}^n(h)$ such that $b = As' + e \bmod q$

1 $(M, B) \leftarrow$ *Algorithm 1*(A, b, d)
2 $B \leftarrow$BKZ-$\beta(B)$
3 $\tilde{L}^{(0)} \leftarrow$ *Algorithm 3*$(M, B, \ell^{(0)}, w^{(0)}, t; \mathsf{meet\text{-}lwe\text{-}params})$
4 **if** $\exists\ s_g \in \tilde{L}^{(0)}$ **then**
5 Parse $\mathsf{NP}_B(Ms_g)$ into $(e, \bar{s}) \in \mathbb{Z}^{m'} \times \mathbb{Z}^{n-d+1}$.
6 **return** $(\bar{s} \| s_g)$
7 **end**
8 **return** $\perp$

Theorem 2. *Let $(A, b) \in \mathbb{Z}_q^{m \times (n+1)}$ be an LWE instance with a secret vector $s \in \mathcal{T}^d(w)$ and noise vector $e \leftarrow \mathcal{G}_\sigma$. Then Algorithm 4 with* **Params** *successfully recovers the solution s with probability at least*

$$p_{suc} = p_{np} \cdot p_{hw} \cdot p_{suc}^{meet},$$

where p_{suc}^{meet} is the success probability of Algorithm 3 defined as Eq. (20),

$$p_{np} = \prod_{i=1}^{m+n-d+1} \operatorname{erf}\left(\frac{\|b_i^*\|\sqrt{2}}{\sigma}\right) \text{ and } p_{hw} = \binom{n-w}{d-w^{(0)}}\binom{w}{w^{(0)}} / \binom{n}{d}.$$

The execution time is estimated by $T_{bkz}(m, \beta) + T_{meet}$ where $T_{bkz}(m, \beta)$ is the running time of BKZ-β for m-dimensional lattice, and T_{meet} is the running time of Algorithm 3, computed as Eq. (19).

Proof. For the attack to succeed, we need the following events:

- $\mathsf{NP}_B(e) = e$, whose probability p_{np} is already discussed in Section 2.3.
- $HW(s) = w^{(0)}$, whose probability p_{hw} comes from simple combinatorics.
- $s_g \in \tilde{L}^{(0)}$, whose probability is analyzed in Theorem 1 by p_{suc}^{meet}.

Therefore, the success probability of Algorithm 4 is $p_{suc} = p_{np} \cdot p_{hw} \cdot p_{suc}^{meet}$.

The timing cost part is immediate: the computational cost is clearly dominated by the BKZ reduction (Line 4) and the execution of Algorithm 3. □

7 Concrete Estimations

Given input LWE parameters such as m, n, q, σ, w, there would be the optimal parameterizations of Algorithm 4 producing minimal cost

$$\frac{1}{p_{suc}} \left(T_{bkz}(m, \beta) + T_{meet}\right).$$

We have implemented a Python script that investigates the minimal cost and such optimal parameterization, which is available in Supplementary Material.

In particular, we consider two popular parameter regimes from the literature. The first one comes from the fully homomorphic encryption (FHE) literature, which employs huge $n \geq 2^{15}$ and $q \geq 2^{500}$. In particular, it usually uses (extremely) sparse secret key of hamming weight from 64 to 192, which is essential to have efficient bootstrapping procedures [16, 20, 31, 38].

We also consider the other popular parameter regime, say the post-quantum cryptography (PQC) literature, which have much smaller dimension n and modulus q than HE. In this regime, the secret key size still affects the efficiency, and it is also common to use *small* secrets. For estimations, we especially figure out the schemes based on sparse (or fixed hamming-weight) ternary secret [4, 18].[10]

Cost model and assumptions. We mainly follow the similar assumptions and cost model used in lattice-estimator [2], since the estimations for previous attacks (including non-hybrids) are obtained from that.

- We identify a single integer, a floating point number, and an element in $\mathbb{Z}_q$ as a unit *word*, and the unit cost of arithmetic operations over them by rop.
- For the hash table (or RAM) cost, we assume $T_{\mathsf{hash}}(r)$ takes r unit costs, and random access for the hash table takes a unit cost.
- We assume that BKZ-β algorithm on m-dimensional basis takes $5.46m \cdot 2^{0.296\beta+20.388}$ [8] unit cost, which is a default setting in lattice-estimator [2].
- We use Geometric Series Assumption (GSA) to simulate the Gram-Schmidt norm of BKZ output basis.

Remark 4. The cost model for the concrete security estimation reaches to the point of considering gate-level cost for each unit operation or consider more precise model for BKZ such as Z-shape model [25, 34]. We acknowledge that these kinds of details should be eventually examined for our attack as well, and leave it as a future work.

Restriction of parameter search range. For a quick estimation, we want to exclusively focus on the some specific search range. In particular, we want to use the heuristic suggestion of $r^{(i)}$ and $R_{lsh}^{(i)}$ (Eq. (21) and Eq. (5)). However, recall that this choice was based on (in fact, false) expectation of the form

$$\mathbb{E}_{\chi}[1-(1-p(x))^R] \approx 1-(1-\mathbb{E}_{\chi^{(i)}}[p(x)])^R],$$

so the target specific search range should resolve this issue anyhow. We manage to find out some appropriate choices parameters from some trial-and-error, which is summarized as Figure 6, and the relevant experimental details about the approximation will be presented in Appendix C.2.

Estimation results. Table 1 shows estimation results of several attack strategies for HE and PQC regimes, respectively, based on the parameter search

[10] As a subtle issue, the error distribution of our target PQC scheme is uniform ternary distribution. Our estimation is done by approximating it by Gaussian with the same variance.

Norm bounds $\ell^{(i)}$, torus LSH block-length $b_{lsh}^{(i)}$, NCF repetition numbers $R_{ncf}^{(i)}$, and projection dimensions $r^{(i)}$ are chosen as follows.

- $\ell^{(i)} = 6\sigma$ for $0 \le i \le t-1$
- $b_{lsh}^{(i)} = 12\sigma$ for $1 \le i \le t$
- $\chi^{(0)} = \mathcal{G}_\sigma$ and $\chi^{(i)} = \mathcal{U}_{\ell^{(i)}}$ for $1 \le i \le t$.
- $R_{ncf}^{(i)} = \left\lceil C_{lsh}/\mathbb{E}_{\chi^{(i-1)}}[p_{lsh}(e)] \right\rceil$ with $C_{lsh} = 10$ for $1 \le i \le t$ (See Eq. (5))
- $r^{(i)}$: the maximal one s.t. $R^{(i-1)} \ge 2C_{proj}/\mathbb{E}_{\chi^{(i-1)}}[p_{rep}(e)]$ with $C_{proj} = 10$, where $R^{(i-1)}$ is the number of $w^{(i)}$-reps of $w^{(i-1)}$-weight ternary vector, for $1 \le i < t$ (See Eq. (21))

The other parameters, such as guessing dimension d, BKZ blocksize β, and split weights $w^{(i)}$ are exhaustively searched.

Fig. 6: Parameter search range

range of Figure 6. The numbers for our attacks and the previous hybrid attack (HG) [34] are obtained from our script[11], and the others are obtained from lattice-estimator [2]. For the reader interested in the optimal parameterization itself as well as the optimal attack cost, the log files can be found also in Supplementary Material.

The consequence is quite expectable. Our attack beats the previous best attacks when the sparsity of the secret is severe, such as HE parameters or (currently inactive) IEEE standards for NTRU [4]. However, our attack falls behind the non-hybrid attack for larger hamming-weight, especially for the recent PQC scheme such as NTRU for NIST PQC standardization [18], which confirms again the recent trend of avoiding extremely sparse secrets. Nonetheless, our attack is better than Howgrave-Graham's hybrid attack [34] in every parameter set, which shows that our generalization anyhow makes some advance from the previous hybrid attack [34].

Separation of non-hybrids. One might be curious how hybrid attacks become worse than 'Non-hybs' in a certain regime, or why 'Non-hybs' is not treated as a special case of our hybrid attack with $d = 0$ and $w^{(0)} = 0$ in our estimation. This is because the post-processing combinatorial strategy requires many new constraints regarding modulo B, which are not required for the pure lattice attack ('Non-hybs'). In particular, the condition Eq. (13) on Gram-Schmidt norm $\|b_i^*\|$ critically affect such a drastic discrepancy.

[11] The lattice-estimator [2] also provide some estimation of the HG attack [34]. However, we rather use our own estimation, as we correctly deal with scaling factor Remark 1 and near-collision finding cost.

Consideration on the best level. For all our target parameters, our hybrid attack obtains the minimal cost at top level 2, and top level 3 sometimes even worse than top level 1 case; see $(2^{16}, 2^{1553}, 192)$ case. This is in contrast to the original Meet-LWE [42], which obtained the minimal attack cost at (up to) top level 4. We suspect the following reasoning for this. First, recall that the lattice reduction phase already significantly reduces the searching space. This makes the improvements in the exponent from Meet-LWE over Meet-in-the-middle marginal than the direct key search. Second, the higher-level iterations retains more heavier overhead in the time complexity and success probability than [42], particularly due to the near-collision finding algorithms.

(n, q, w)	Primal Strategy				Dual Strategy [5, 21]
	Non-hybs [6, 40]	HG [34] ($t = 1$)	**Ours**		
			$t = 2$	$t = 3$	
Homomorphic Encryption [16, 31, 37, 38]					
$(2^{15}, 2^{699}, 128)$	161	132.9	**127.1**	127.5	133
$(2^{15}, 2^{768}, 192)$	145	130.0	**125.9**	126.6	133
$(2^{16}, 2^{1553}, 192)$	145	129.5	**127.1**	138.1	136
$(2^{16}, 2^{1450}, 64)$	155	111.4	**105.1**	106.6	108
NTRU IEEE [4]					
(659,2048,76)	153	148.2	**135.4**	145.7	136
(761,2048,84)	176	166.5	**151.5**	162.2	154
(1087,2048,126)	255	241.8	**219.5**	233.5	221
NTRU Encrypt [18], NTRU Prime [13]					
(509,2048,254)	**131**	195.4	175.1	195.4	137
(677,2048,254)	**171**	242.2	216.2	238.2	175
(653,4621,288)	**153**	218.4	196.2	216.9	158

Table 1: Complexity estimations of LWE attacks. For all HE parameters, the error is (discrete) Gaussian of $\sigma = 3.2$. For other parameters, the error distribution is the uniform distribution over $[-1, 0, 1]$, and approximated by Gaussian distribution of the same variance. We remark that there is a significant concern [26] about the core assumption used for the dual strategy at the time of writing this paper, so the estimation result for dual strategy loses some reliability.

References

1. HEAAN. https://github.com/snucrypto/HEAAN. Accessed: 2023-10-06.
2. Lattice-Estimator. https://github.com/malb/lattice-estimator. Accessed: 2023-10-06.
3. Lattigo. https://github.com/tuneinsight/lattigo. Accessed: 2023-10-06.
4. IEEE Standard Specification for Public Key Cryptographic Techniques Based on Hard Problems over Lattices. IEEE Std 1363.1-2008, 2008. Accessed: 2022-05-27.
5. M. R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 103–129. Springer, 2017.
6. M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 297–322. Springer, 2017.
7. M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.
8. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key {Exchange—A} new hope. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 327–343, 2016.
9. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009.
10. H. Baan, S. Bhattacharya, S. Fluhrer, O. Garcia-Morchon, T. Laarhoven, R. Rietman, M.-J. O. Saarinen, L. Tolhuizen, and Z. Zhang. Round5: Compact and fast post-quantum public-key encryption. In *International Conference on Post-Quantum Cryptography*, pages 83–102. Springer, 2019.
11. S. Bai and S. D. Galbraith. Lattice decoding attacks on binary LWE. In *Australasian Conference on Information Security and Privacy*, pages 322–337. Springer, 2014.
12. A. Becker, J.-S. Coron, and A. Joux. Improved generic algorithms for hard knapsacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 364–385. Springer, 2011.
13. D. J. Bernstein, B. B. Brumley, M.-S. Chen, C. Chuengsatiansup, T. Lange, A. Marotzke, B.-Y. Peng, N. Tuveri, C. van Vredendaal, and B.-Y. Yang. Ntru prime: round-3 updates. 2020.
14. L. Bi, X. Lu, J. Luo, and K. Wang. Hybrid dual and meet-lwe attack. In *Information Security and Privacy: 27th Australasian Conference, ACISP 2022, Wollongong, NSW, Australia, November 28–30, 2022, Proceedings*, pages 168–188. Springer, 2022.
15. X. Bonnetain, R. Bricout, A. Schrottenloher, and Y. Shen. Improved classical and quantum algorithms for subset-sum. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 633–666. Springer, 2020.
16. J.-P. Bossuat, C. Mouchet, J. Troncoso-Pastoriza, and J.-P. Hubaux. Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 587–617. Springer, 2021.

17. J. Buchmann, F. Göpfert, R. Player, and T. 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.
18. C. Chen, O. Danba, J. Hoffstein, A. Hülsing, J. Rijneveld, J. M. Schanck, T. Saito, P. Schwabe, W. Whyte, K. Xagawa, T. Yamakawa, and Z. Zhang. PQC round-3 candidate: NTRU. Technical report, 2020. Accessed: 2022-05-20.
19. Y. Chen and P. 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.
20. J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song. Bootstrapping for approximate homomorphic encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 360–384. Springer, 2018.
21. J. H. Cheon, M. Hhan, S. Hong, and Y. Son. A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret LWE. *IEEE Access*, 7:89497–89506, 2019.
22. B. R. Curtis and R. Player. On the feasibility and impact of standardising sparse-secret LWE parameter sets for homomorphic encryption. In *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 1–10, 2019.
23. M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004.
24. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal gaussians. In *Annual Cryptology Conference*, pages 40–56. Springer, 2013.
25. L. Ducas, T. Espitau, and E. W. Postlethwaite. Finding short integer solutions when the modulus is small. In *Annual International Cryptology Conference*, pages 150–176. Springer, 2023.
26. L. Ducas and L. N. Pulles. Does the dual-sieve attack on learning with errors even work? In *Annual International Cryptology Conference*, pages 37–69. Springer, 2023.
27. T. Espitau, A. Joux, and N. Kharchenko. On a dual/hybrid approach to small secret LWE. In *International Conference on Cryptology in India*, pages 440–462. Springer, 2020.
28. T. Glaser and A. May. How to enumerate LWE keys as narrow as in Kyber/Dilithium. *Cryptology ePrint Archive*, 2022.
29. T. Güneysu, V. Lyubashevsky, and T. Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 530–547. Springer, 2012.
30. Q. Guo and T. Johansson. Faster dual lattice attacks for solving LWE with applications to CRYSTALS. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 33–62. Springer, 2021.
31. K. Han and D. Ki. Better bootstrapping for approximate homomorphic encryption. In *Cryptographers' Track at the RSA Conference*, pages 364–390. Springer, 2020.
32. S. Har-Peled, P. Indyk, and R. Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of computing*, 8(1):321–350, 2012.
33. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium*, pages 267–288. Springer, 1998.
34. N. Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In *Annual International Cryptology Conference*, pages 150–169. Springer, 2007.

35. N. Howgrave-Graham and A. Joux. New generic algorithms for hard knapsacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 235–256. Springer, 2010.
36. E. Kirshanova and A. May. How to find ternary LWE keys using locality sensitive hashing. In *IMA International Conference on Cryptography and Coding*, pages 247–264. Springer, 2021.
37. J.-W. Lee, E. Lee, Y. Lee, Y.-S. Kim, and J.-S. No. High-precision bootstrapping of RNS-CKKS homomorphic encryption using optimal minimax polynomial approximation and inverse sine function. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 618–647. Springer, 2021.
38. Y. Lee, J.-W. Lee, Y.-S. Kim, Y. Kim, J.-S. No, and H. Kang. High-precision bootstrapping for approximate homomorphic encryption by error variance minimization. In *Advances in Cryptology–EUROCRYPT 2022: 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30–June 3, 2022, Proceedings, Part I*, pages 551–580. Springer, 2022.
39. R. Lindner and C. Peikert. Better key sizes (and attacks) for lwe-based encryption. In *Cryptographers' Track at the RSA Conference*, pages 319–339. Springer, 2011.
40. M. Liu and P. Q. Nguyen. Solving bdd by enumeration: An update. In *Cryptographers' Track at the RSA Conference*, pages 293–309. Springer, 2013.
41. MATZOV. Report on the Security of LWE: Improved Dual Lattice Attack. https://zenodo.org/record/6493704, 2022. Accessed: 2023-10-06.
42. A. May. How to meet ternary LWE keys. In *Annual International Cryptology Conference*, pages 701–731. Springer, 2021.
43. P. Q. Nguyen. Boosting the hybrid attack on ntru: Torus LSH, Permuted HNF and Boxed Sphere. *NIST Third PQC Standardization Conference*, 2021.
44. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.
45. Y. Son and J. H. Cheon. Revisiting the hybrid attack on sparse secret LWE and application to HE parameters. In *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 11–20, 2019.
46. I. van Hoof, E. Kirshanova, and A. May. Quantum key search for ternary LWE. In *International Conference on Post-Quantum Cryptography*, pages 117–132. Springer, 2021.
47. T. Wunderer. A detailed analysis of the hybrid lattice-reduction and meet-in-the-middle attack. *Journal of Mathematical Cryptology*, 13(1):1–26, 2019.

A Flaw in the analysis of [36]

As noted in the introduction, Kirshanova and May suggested the improved Meet-LWE algorithm based on the locality-sensitive hashing, called LSH-Meet-LWE [36]. We briefly describe some steps of the LSH-Meet-LWE algorithm and the flaw in the analysis, which was confirmed by the personal conversation with the authors.

Suppose that we are given a $Ms = e \bmod q$ with the secret key $s \in \mathcal{T}^n(w)$ and the ternary error vector e. The following sets are the level-1 lists in the LSH-Meet-LWE algorithm

$$L_1^{(1)} = \left\{ s_1^{(1)} \in \mathcal{T}^n(w/2) : Ms_1^{(1)} \in \mathbb{Z}_q^{n-r} \times \{0\}^{r/2} \times \{\pm 1, 0\}^{r/2} \right\}$$
$$L_2^{(1)} = \left\{ s_2^{(1)} \in \mathcal{T}^n(w/2) : Ms_2^{(1)} \in \mathbb{Z}_q^{n-r} \times \{\pm 1, 0\}^{r/2} \times \{0\}^{r/2} \right\}.$$

These two lists have the different constraints on the last r coordinates; half of the r coordinates are fixed to 0, and the other half is bounded by 1.

The number of representations $(s_1^{(1)}, s_2^{(1)}) \in \mathcal{T}^n(w/2) \times \mathcal{T}^n(w/2)$ such that $s_1^{(1)} - s_2^{(1)} = s$ is $R = \binom{w}{w/2}^2$. The authors claim that the choice of r such that

$$R \approx q^{r/2} \cdot (q/3)^{r/2} \tag{23}$$

ensures that there is a representation $(s_1^{(1)}, s_2^{(1)}) \in L_1^{(1)} \times L_2^{(1)}$ on expectation. However, this claim turns out to be false.

Let us clarify the details of the problem. Since the list $L_1^{(1)}$ has $3^{r/2}$ possible entries in the last r coordinates, the choice of Eq. (23) indeed ensures that there is a representation $(s_1^{(1)}, s_2^{(1)})$ such that $s_1^{(1)} \in L_1^{(1)}$ on expectation. In the original Meet-LWE attack (with the corresponding list definitions therein), $s_1^{(1)} \in L_1^{(1)}$ automatically implies the event $s_2^{(1)} \in L_2^{(1)}$ [42, Section 5]. This is because the algorithm essentially enumerates all the possible r coordinates of the error vector.

This automatic implication is not the case for LSH-Meet-LWE, which does not enumerate the partial error vector. The condition $s_1^{(1)} \in L_1^{(1)}$ says that

$$\pi_r(Ms_1^{(1)}) \in \{0\}^{r/2} \times \{\pm 1, 0\}^{r/2}.$$

From the (simplified) LWE identity $As = e \bmod q$, we have

$$\pi_r(Ms_2^{(1)}) = \pi_r(e) - \pi_r(Ms_1^{(1)}).$$

To make $\pi_r(Ms_2^{(1)}) \in \{\pm 1, 0\}^{r/2} \times \{0\}^{r/2}$ to be true, it must hold that

$$\pi_{r/2}(Ms_1^{(1)}) = \pi_{r/2}(e)$$

which only happens with probability $1/3^{r/2}$ (assuming e is sampled from the uniform random ternary vector.) This probability is neglected in the original analysis, and the algorithm becomes impractical if this analysis is included.

B Missing Proofs

B.1 Proof of Lemma 1

Observe that in order to $x'_r = x - x_r$ be ternary, x and x_r should agree on every position where x and x_r are both nonzero; if not x'_r has some entry 2 or -2. Writing ℓ be the number of such positions, $x - x_r$ should have $h + w - 2\ell$ nonzero positions. Since $x'_r = x - x_r$ also has weight w, we know $\ell = h/2$. Therefore, we may choose $\ell = h/2$ nonzero positions of x_r among h nonzero positions of x, which accounts for $\binom{h}{h/2}$ term. The remaining $w - h/2$ nonzero positions of x_r can be both ± 1 over $d - h$ positions where x is zero, which accounts for $\binom{d-h}{w-h/2} \cdot 2^{w-h/2}$ term. □

B.2 Proof of Proposition 1

The complexity part. Under Model 1, the probability that a random pair (y_1, y_2) in $L \times L$ have the same image under for a fixed h_c is

$$p_{col} = \prod_{1 \le i \le r} \frac{1}{n_i}. \tag{24}$$

To prove the claim, we instead show the following: for any $t > 0$, the algorithm terminates in time

$$R|L| \cdot T_{\mathsf{hash}} + R(|L|^2 p_{col}/2 + t) \cdot T_{\mathsf{check}}$$

with probability at least $1 - R|L|^2 p_{col}/2t^2$, which clearly yields the claim by taking $\epsilon = R|L|^2 p_{col}/2t^2$.

First, the number of hash evaluations is clearly $R \cdot |L|$ hash evaluations (lines 6-8) and consumes $|L|$ words of space for hash table T.

The remaining part is dominated by the time for checking if the collision pairs (y, z) in L such that $h_c(y) = h_c(z)$ is ℓ-near-collision. We count such pairs for a fixed hash function h_c. Let $|L| = N$ and write $L = \{y_1, ..., y_N\}$ and $M := 1/p_{col} = \prod_{1 \le i \le r} n_i$. Let $I_{i,j}$ be a random variable that equals 1 if $h_c(y_i) = h_c(y_j)$ and otherwise $I_{i,j} = 0$. Note that $\mathbb{E}[I_{i,j}] = 1/M$ over the randomness of L. We will compute the value $X := \sum_{i<j} I_{i,j}$, whose expectation is $\mathbb{E}[X] = N(N-1)/2M$. The variance of X is computed as follows.

$$\begin{aligned}
\mathrm{Var}[X] = \mathrm{Var}\left[\sum_{i<j} I_{i,j}\right] &= \mathbb{E}\left[\left(\sum_{i<j} I_{i,j} - \mathbb{E}[I_{i,j}]\right)^2\right] \\
&= \left(\sum_{i<j}\sum_{k<\ell} \mathbb{E}[I_{i,j} I_{k,\ell}]\right) - \frac{N^2(N-1)^2}{4M^2}.
\end{aligned}$$

By considering the following three cases: 1) $(i,j) = (k,\ell)$, 2) $|\{i,j\} \cap \{k,\ell\}| = 1$, and 3) otherwise, this quantity is computed by

$$\mathrm{Var}[X] = \frac{(M-1)N(N-1)}{2M^2} \leq \mathbb{E}[X].$$

Applying Chebyshev's inequality, we have

$$\Pr\left[X \geq \mathbb{E}[X] + t\right] \leq \frac{\mathrm{Var}[X]}{t^2} \leq \frac{\mathbb{E}[X]}{t^2}$$

for any $t > 0$. In other words, with probability at least $1 - \mathbb{E}[X]/t^2$, a single iteration with h_c suffices to check if $\mathbb{E}[X] + t$ pairs are ℓ-near-collision.

The full algorithm with R repetitions is expected to check whether less than

$$R \cdot (\mathbb{E}[X] + t) \leq R \cdot (|L|^2 p_{col}/2 + t)$$

pairs are ℓ-near-collision or not with probability at least $1 - R \cdot |L|^2 p_{col}/2t^2$ by the union bound.

For Model 2, the above complexity analysis can be applied in almost the same way since this model only differs from the purely random model by one special near-collision pair, say (y_1, y_2). Regarding the high collision probability of y_1 and y_2, the term $+1$ is added. The analysis for Model 1 applies to the other pair. In other words, the claim almost similarly holds when assuming Model 2. □

The correctness part. We start from the probability that a fixed ℓ-near-collision $(y_1, y_2) \in \mathcal{B} \times \mathcal{B}$ collides by a random h_c. Writing $e = (e_1, \cdots, e_r) := y_1 - y_2$, the i-th coordinate of y_1 and y_2 are mapped into the same image by h_c with the probability $1 - |e_i|/b_i$ if $n_i > 1$, and always mapped into 0 if $n_i = 1$. Thus the probability that y and $y + e$ in $\mathcal{B}$ have the same image under h_c is

$$p_{lsh}(e) = \prod_{\substack{1 \leq i \leq r \\ n_i > 1}} \left(1 - \frac{|e_i|}{b_i}\right). \tag{25}$$

We first show the correctness of the algorithm under Model 1, which can be rephrased as follows.

Lemma 8. *Assume the input list L follows Model 1. Then Algorithm 2 finds a random ℓ-near-collision pair with a probability at least*

$$p_{ncf}(\ell; \mathcal{U}([-\ell,\ell]^r)) := \mathop{\mathbb{E}}_{e \leftarrow \mathcal{U}([-\ell,\ell]^r)} \left[1 - (1 - p_{lsh}(e))^R\right]$$

Proof. Let (y_1, y_2) be a random ℓ-near-collision pair. The probability of one iteration succeeding to find this pair is $p_{lsh}(e)$ of Eq. (25), which only depends on the difference $d = y_1 - y_2$. Since we use R independent hash functions and it suffices to succeed only one of them, the algorithm succeeds in finding (y_1, y_2) with probability $1 - (1 - p_{lsh}(e))^R$.

Now let $\mathcal{E}$ be the probabilistic variable of the difference $e = y_1 - y_2$ of the random ℓ-near-collision pair (y_1, y_2) in $\mathcal{B}$, and then the success probability is represented by $\mathbb{E}_{e\leftarrow\mathcal{E}}\left[1-(1-p_{lsh}(e))^R\right]$. However, the distribution $\mathcal{E}$ of e is slightly different from the uniform distribution over $[-\ell,\ell]^r$; $\Pr[\mathcal{E}=e]$ is proportional to $\prod_{i\in[r]}\left(1-\frac{|e_i|}{q_i}\right)$ for $\|e\|_\infty \le \ell$. So it remains to prove that

$$\mathop{\mathbb{E}}_{e\leftarrow\mathcal{E}}\left[1-(1-p_{lsh}(e))^R\right] \ge \mathop{\mathbb{E}}_{e\leftarrow[-\ell,\ell]^r}\left[1-(1-p_{lsh}(e))^R\right]. \tag{26}$$

This can be proven by the probabilistic version of Chebyshev's sum inequality

$$\mathbb{E}_{e\leftarrow[-\ell,\ell]}[f\cdot g] \ge \mathbb{E}_{e\leftarrow[-\ell,\ell]}[f]\cdot\mathbb{E}_{e\leftarrow[-\ell,\ell]}[g] \tag{27}$$

when f, g are both non-increasing functions or satisfy $(f(x)-f(y))(g(x)-g(y)) \ge 0$ for any $x, y \in [-\ell,\ell]$ in general.

We sketch the rough idea. The left-hand side of Eq. (26) can be written as

$$\left(\int_{[-\ell,\ell]^r}\left(1-(1-p_{lsh}(e))^R\right)\cdot\left(1-\frac{|e_i|}{q_i}\right)de\right)\Bigg/\left(\int_{[-\ell,\ell]^r}\left(1-\frac{|e_i|}{q_i}\right)de\right).$$

For each $i\in[r]$, fix all coordinates of e except $x = e_i$. Let $g(x) = C(1-|x|/q_i)$ and $f(x) = 1-(1-D(1-|x|/b_i))^R$ for some $C, D > 0$. Then f, g are both even functions and decreasing for $[0,\ell]$, so that $(f(x)-f(y))(g(x)-g(y)) \ge 0$ for any $x, y \in [-\ell,\ell]$. Thus, we can apply Eq. (27) for each e_i for $i\in[r]$ to obtain the desired result. □

The correctness for Model 2 is almost analog of Lemma 8, where the probability that the infinity norm of the error e sampled from $\mathcal{D}$ is bounded by ℓ is additionally taken into account.

Lemma 9. *Assume the input list L follows Model 2 where the special near-collision pair is sampled from $\mathcal{D}$. Then Algorithm 2 finds the special near-collision pair with a probability at least*

$$p_{ncf}(\ell;\mathcal{D}) := \Pr_{e\leftarrow\mathcal{D}}[\|e\|_\infty \le \ell]\cdot\mathop{\mathbb{E}}_{e\leftarrow\mathcal{D}}\left[1-(1-p_{lsh}(e))^R\right].$$

Lemma 8 and Lemma 9 completes the correctness part of Proposition 1. □

B.3 Justification of Heuristic 1

Let $e_*^{(j)} := [Ms_*^{(j)}]_{B,r^{(j+1)}}$. First, observe that the following equation holds

$$Ms_k^{(i)} = Ms_{2k-1}^{(i+1)} - Ms_{2k}^{(i+1)} \bmod B_{r^{(i+1)}}$$

and from Figure 5, the size of $e_k^{(i)} = [Ms_k^{(i)}]_{B,r^{(i+1)}}$ is a crucial factor for $E_{sp,k}^{(i)}$; when $e_k^{(i)}$ gets smaller, it is more likely to $[Ms_{2k-1}^{(i+1)}]_{B,r^{(i+1)}}$ and $[Ms_{2k}^{(i+1)}]_{B,r^{(i+1)}}$ both lie in $\mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}}$. Thus, it is natural to consider the effect of Eq. (16) on $e_k^{(i)}$.

In this regard, observe that Eq. (16) with $j < i-1$ is essentially about the difference of $e_*^{(j+1)}$; for example, $j = i-2$ implies

$$e_\ell^{(i-2)} = e_{2\ell-1}^{(i-1)} - e_{2\ell}^{(i-1)} \bmod B_{r^{(i-1)}}.$$

We expect it has a rather indirect effect on $e_k^{(i)}$, and establish the following argument.

Argument 1. It holds that

$$\Pr[E_{sp}^{(i)}] \approx \Pr[\bigwedge_{k\in[2^i]} E_{sp,k}^{(i)}] = \prod_{k\in[2^{i-1}]} \Pr[E_{sp,2k-1}^{(i)} \wedge E_{sp,2k}^{(i)}].$$

Meanwhile, Eq. (16) with $j = i-1$ that directly affects to $e_k^{(i)}$ by the equation $e_k^{(i-1)} = e_{2k-1}^{(i)} - e_{2k}^{(i)} \bmod B_{r^{(i)}}$, and this would really make the probability $\Pr[E_{sp,2k-1}^{(i)} \wedge E_{sp,2k}^{(i)}]$ quite different from $\Pr[E_{sp,2k-1}^{(i)}] \cdot \Pr[E_{sp,2k}^{(i)}]$. Fortunately, we managed to argue the following lower bound.

Argument 2. It holds that for any $k \in [2^{i-1}]$,

$$\Pr[E_{sp,2k-1}^{(i)} \wedge E_{sp,2k}^{(i)}] \geq \Pr[E_{sp,2k-1}^{(i)}] \cdot \Pr[E_{sp,2k}^{(i)}].$$

We explain the rationale behind Argument 2. Write $d_k = [Ms_k^{(i-1)}]_{B,r^{(i+1)}}$, $e_{2k-1} = [Ms_{2k-1}^{(i)}]_{B,r^{(i+1)}}$ and $e_{2k} = [Ms_{2k}^{(i)}]_{B,r^{(i+1)}}$ by dropping superscripts for a simpler description. Since $d_k \in \mathcal{C}_{\ell^{(i-1)}}^{r^{(i+1)}}$ and $e_{2k-1}, e_{2k} \in \mathcal{C}_{\ell^{(i+1)}}^{r^{(i)}}$ from the constraint, it holds that

$$d_k = e_{2k-1} - e_{2k}$$

without modulo B thanks to the condition on $\ell^{(i)}$. Then the left-hand side of Argument 2 expects that $E_{sp,2k-1}^{(i)}$ and $E_{sp,2k}^{(i)}$ occur under the relation $d_k = e_{2k-1} - e_{2k}$, and the right-hand side of Argument 2 expects a similar event but for independent e_{2k-1} and e_{2k}.

The relation $d_k = e_{2k-1} - e_{2k}$ implies some restriction on the area where (e_{2k-1}, e_{2k}) can lie. More precisely, considering that $e_{2k-1}, e_{2k} \in \mathcal{C}_{\ell^{(i+1)}}^{r^{(i)}}$, the relation restricts e_{2k-1} and e_{2k} so that they never lie in some corner of $\mathcal{C}_{\ell^{(i+1)}}^{r^{(i)}}$. On the other hand, on the right-hand side, there is no restriction on each e_{2k-1} and e_{2k}. This implies that the "expected size" of e_{2k-1} and e_{2k} is smaller in the left-hand side. Finally, observing that the chance of $E_{sp,*}^{(i)}$ increases as the size of e_* decreases, we may expect that the chance of $E_{sp,2k-1}^{(i)} \wedge E_{sp,2k}^{(i)}$ is larger in the left-hand side on average. □

B.4 Justification of Heuristic 2

Basically, we model each element of $[Ms_r]_{B,r^{(i+1)}}$ as a uniformly sample x from $\mathcal{P}(B_{r^{(i+1)}}^*)$ that are independent from each others. The insertion of $[-x-e]_{B_{r^{(i+1)}}}$ comes from the symmetric nature of rep pairs: If (s_r, s_r') is a rep pair, so is $(-s_r', -s_r)$. □

B.5 Proof for Proposition 2

Recall that $E^{(i)}_{sp,k}$ means that at least one pair in $P(s^{(i)}_k)$ lies in $\mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}}$, and the pair $([Ms_r]_{B,r^{(i+1)}}, [Ms'_r]_{B,r^{(i+1)}}) \in P(s^{(i)}_k)$ is represented by $(x, [x+e]_{B,r^{(i+1)}})$ where $x = [Ms_r]_{B,r^{(i+1)}}$ and $e = [Ms^{(i)}_k]_{B,r^{(i+1)}}$.

Under Heuristic 2, for each first element x of the pair in $P(s^{(i)}_k)$, the probability that the corresponding second element is included in $\mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}}$ is

$$p_{rep}(e) = \Pr\left[x \in \mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}}, [x+e]_{B,r^{(i+1)}} \in \mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}} \mid x \leftarrow \mathcal{P}(B^*_{r^{(i+1)}})\right]. \tag{28}$$

Since each first element is independent and by the symmetry of $P(s^{(i)}_k)$, the number of pairs in $\mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}}$ follows $2 \cdot B(R^{(i)}/2, p_{rep}(e))$[12] where $e = [Ms^{(i)}_k]_{B,r^{(i+1)}}$, and $B(\cdot,\cdot)$ is a binomial distribution.

We proceed to explain the distribution $\chi^{(i)}_{sp}$ where e is sampled. For $i = 0$, our problem setting ensures that $x = [Ms^{(0)}_1]_B$ follows Gaussian distribution $\mathcal{G}_\sigma$ so that $\chi^{(0)}_{sp} = \mathcal{G}^{r^{(1)}}_\sigma$.

For $i \geq 1$, we use the fact that each level-i target element $s^{(i)}_k \in S^{(i)}$ defining $e = [Ms^{(i)}_k]_{B,r^{(i+1)}}$ is one element of a rep pair of some $s^{(i-1)}_* \in S^{(i-1)}$. When $s^{(i)}_k$ is the first element of the rep pair of $s^{(i-1)}_*$, Heuristic 2 for $s^{(i-1)}_*$ immediately implies the uniformity of e. If $s^{(i)}_k$ is the second element, $[Ms^{(i)}_k]_{B,r^{(i+1)}}$ follows a $[Ms^{(i-1)}_*]_{B,r^{(i+1)}}$-shift of uniform distribution, and hence still uniform. At the same time, the constraint condition of $S^{(i)}$ enforces that $e^{(i)}_k$ is included in $\mathcal{C}^{r^{(i)}}_{\ell^{(i+1)}}$, so we have $e^{(i)}_k \leftarrow \mathcal{U}^{r^{(i)}}_{\ell^{(i+1)}}$.

Finally, it holds that

$$\begin{aligned}
\Pr[E^{(i)}_{sp,k}] &= \Pr_{e \leftarrow \chi^{(i)}_{sp}}[2 \cdot B(R^{(i)}/2, p_{rep}(e)) > 0] \\
&= \int \Pr[\chi^{(i)}_{sp} = e] \left(1 - (1 - p_{rep}(e))^{R^{(i)}/2}\right) de \\
&= \mathop{\mathbb{E}}_{e \leftarrow \chi^{(i)}_{sp}} \left[1 - (1 - p_{rep}(e))^{R^{(i)}/2}\right]
\end{aligned}$$

where the second equality holds from the law of total probability.

It remains to show the $p_{rep}(x)$ computation. For that, we split $p_{rep}(e)$ into

$$\Pr\left[x \in \mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}} \mid x \leftarrow \mathcal{P}(B^*_{r^{(i+1)}})\right] \cdot \Pr\left[[x+e]_{B_{r^{(i+1)}}} \in \mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}} \mid x \leftarrow \mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}}\right].$$

The first probability is immediately computed as the volume ratio

$$\frac{\mathrm{vol}(\mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}})}{\mathrm{vol}(\mathcal{P}(B^*_{r^{(i+1)}}))} = \prod_{i \in [r^{(i+1)}]} \frac{2\ell^{(i+1)}}{\|b^*_{m-r^{(i+1)}+j}\|}.$$

[12] We stress that it is not $B(R^{(i)}, p_{rep}(e))$, which does not take into account the symmetry.

The second probability expects that a random element $x \in \mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}}$ still remains in $\mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}}$ after adding a vector $e = (e_1, \cdots, e_{r^{(i+1)}})$. Looking at each j-th coordinate, we expect that $x_j + e_j$ lies in $[-\ell^{(i+1)}, \ell^{(i+1)})$ where $x_j \leftarrow [-\ell^{(i+1)}, \ell^{(i+1)})$, whose probability is exactly $1 - \frac{|e_j|}{2\ell^{(i+1)}}$. Thus the second probability is $\prod_{j \in [r^{(i+1)}]} \left(1 - \frac{|e_j|}{2\ell^{(i+1)}}\right)$. □

B.6 Proof for Lemma 6

A level-$(t-1)$ target element $s_k^{(t-1)}$ successfully splits into $\tilde{S}^{(t)} = S^{(t)} \subset \mathcal{T}^d(w^{(t)})$ if $S^{(t)}$ includes a $w^{(t)}$-rep pair (s_r, s'_r) (of $s_k^{(t-1)}$). The probability of both of s_r, s'_r are included in $S^{(t)}$ is

$$p = (|S^{(t)}|/|\mathcal{T}^d(w^{(t)})|)^2.$$

Since we have $R^{(t-1)} := R(d, w^{(t)}, w^{(t-1)})$ numbers of rep pairs, the number of rep pairs in $S^{(t)}$ follows the distribution $B(R^{(t-1)}, p)$, and hence at least one rep pair of $s_k^{(i)}$ is in $S^{(t)}$ with probability $1 - (1-p)^{R^{(t-1)}}$.

Recall we choose $|S^{(t)}| = \frac{\sqrt{3}|\mathcal{T}^d(w^{(t)})|}{R^{(t-1)}}$. It gives

$$1 - (1-p)^{R^{(t-1)}} \geq 1 - e^{-R^{(t-1)}p} \geq 1 - e^{-3} \geq 0.95.$$

Since there are total 2^{t-1} numbers of level-$(t-1)$ target elements, we conclude that

$$\Pr[E_{sp}^{(t-1)}] \geq \left(1 - e^{-3}\right)^{2^{t-1}} \geq 0.95^{2^{t-1}}.$$

□

B.7 Justification of Heuristic 3

This heuristic can be justified almost similarly to Heuristic 1, as the situation is conceptually identical: The size of $e_k^{(i)} = [Ms_k^{(i)}]_{B,r^{(i)}}$ is a crucial factor for the event $E_{ncf,k}^{(i)}$ that NCF on $\tilde{S}^{(i+1)}$ finds $(s_{2k-1}^{(i+1)}, s_{2k}^{(i+1)})$. □

B.8 Justification of Heuristic 4

The heuristic is based on the reasonable assumption that the elements of $L^{(i+1)}$ are uniformly and independently distributed. The level-1 list $L^{(1)}$ stands as an exception to reflect the fact that the target near-collision we want to compute through this list is unusually short. □

Remark 5. This heuristic does not reflect the symmetry of $L^{(i)}$ described in Heuristic 2. We remark that ignoring symmetry does not significantly affect the analysis here because our NCF algorithm does not utilize the symmetry of the underlying domain.

B.9 Proof for Proposition 3

Let $e_j = [Ms_j^{(i+1)}]_{B,r^{(i)}}$ for $j = 2k-1, 2k$. We argue that (e_{2k-1}, e_{2k}) becomes an $\ell^{(i)}$-near-collision in $L^{(i+1)}$ with a certain probability.

Let $d_k = [Ms_k^{(i)}]_{B,r^{(i)}}$ so that

$$e_{2k} = e_{2k-1} + d_k \bmod B_{r^{(i)}}.$$

Since $s_k^{(i)} \in S^{(i)}$, we have $\|d_k\|_\infty \le \ell^{(i)}$ and the above equation implies what we desired except the existence of the modulus $\bmod B_{r^{(i)}}$. We need to show that

$$e_{2k} = e_{2k-1} + d_k, \text{ or } e_{2k,j} = e_{2k-1,j} + d_{k,j} \text{ for every } j \in [r^{(i)}] \tag{29}$$

to say that (e_{2k-1}, e_{2k}) is an $\ell^{(i)}$-near-collision.

Observe that the last $r^{(i+1)}$ coordinates of both e_{2k} and e_{2k-1} are already less than $\ell^{(i+1)}$ thanks to level-$(i+1)$ constraint and the condition of Eq. (13). This implies that Eq. (29) holds for $j \in [r^{(i)} - r^{(i+1)}, r^{(i)}]$.

In the remaining index $j \in [r^{(i)} - r^{(i+1)}]$, considering Lemma 5, Eq. (29) translates into

$$|e_{2k-1,j} + d_{k,j}| \le \|b^*_{m-r^{(i)}+j}\|/2. \tag{30}$$

According to the heuristic, each coordinate $e_{2k-1,j}$ is distributed uniformly over

$$\left[-\|b^*_{m-r^{(i)}+j}\|/2, \|b^*_{m-r^{(i)}}\|/2\right),$$

and hence the probability of Eq. (30) for $j \in [r^{(i)} - r^{(i+1)}]$ is $1 - \frac{|d_{k,j}|}{\|b^*_{m-r^{(i)}+j}\|}$.

Since $d_k = [Ms_k^{(i)}]_{B,r^{(i)}} \leftarrow \chi^{(i)}_{ncf}$, the probability of Eq. (29) is

$$p_{nc}(\chi^{(i)}_{ncf}) := \mathop{\mathbb{E}}_{d \leftarrow \chi^{(i)}_{ncf}} \left[\prod_{j \in [r^{(i)} - r^{(i+1)}]} \left(1 - \frac{|d_j|}{\|b^*_{m-r^{(i)}+j}\|}\right)\right],$$

meaning that (e_{2k-1}, e_{2k}) becomes $\ell^{(i)}$-near-collision in $\mathcal{B}^{(i)}$ with probability $p_{nc}(\chi^{(i)}_{ncf})$. The probability that Algorithm 2 successfully finds the $\ell^{(i)}$-near-collision pair (e_{2k-1}, e_{2k}) is $p_{ncf}(\ell^{(i)}; \chi^{(i)}_{ncf})$ as analyzed in Proposition 1. For $i = 0$, the target pair is the special near-collision with the distance $e = [Ms]_B$ following the distribution $\mathcal{G}_\sigma$. If $i \ge 1$, the distance between the target pair follows $\mathcal{U}^{r^{(i)}}_{\ell^{(i)}}$. □

Remark 6. Note that the shape of $p_{nc}(\chi_{ncf})$ is almost identical with $\mathbb{E}_\chi[p_{rep}(x)]$ of Eq. (21) and $\mathbb{E}_\chi[p_{lsh}(x)]$ of Eq. (5) as

$$\mathbb{E}_{x \leftarrow \chi}\left[\prod\left(1 - \frac{|x_j|}{\ell_j}\right)\right],$$

and the computation is also similar. Furthermore, when the Gram-Schmidt norm $\|b^*_j\|$ is sufficiently large, we can ignore the term $p_{nc}(\chi^{(i)}_{ncf})$ because it is almost

1, and hence $\Pr[E^{(i)}_{ncf}] \gtrsim \left(p_{ncf}(\ell^{(i)}; \chi^{(i)}_{ncf})\right)^{2^i}$. This particularly happens when applying this attack for homomorphic encryption parameters, whose modulus q size is enormous.

B.10 Proof for Lemma 7

The computation parts of the i-th level consist of the following steps.

- Line 3: Computing $L^{(i)} := \{[Mx]_{B,r^{(i-1)}} \mid x \in \tilde{S}^{(i)}\}$
- Line 4: Running Algorithm 2 over $L^{(i)}$
- Line 6-11: Checking hamming weight

In the first item (Line 3), for $M = (M_1|...|M_d)$, we first compute $\pi_{B,r^{(i-1)}}(M_i)$ (recall Eqs. (7) and (11)) for $1 \le i \le d$, which is negligibly small compared to the other costs. For $x = (x_1, ..., x_d) \in S^{(i)} \subset \mathcal{T}^d(w^{(i)})$, it holds that

$$\pi_{B,r^{(i-1)}}(Mx) = \sum_{i \in [d]} x_i \cdot \pi_{B,r^{(i-1)}}(M_i)$$

which can be computed by using $r^{(i-1)} \cdot w^{(i)}$ flops using the fact that there are at most $w^{(i)}$ nonzero coordinates in x. The computation of $[v]_{B,r^{(i-1)}}$ requires $(r^{(i-1)})^2$ operations from Lemma 4. The overall cost is much smaller than others.

The cost for the second item (Line 4) directly comes from Proposition 1 as $R^{(i)}_{lsh}\left(|\tilde{S}^{(i)}| \cdot T_{\mathsf{hash}}(r^{(i-1)}) + N^{(i)}_{col}(\epsilon) \cdot T_{\mathsf{check}}(r^{(i-1)})\right)$.

The number of Hamming weight checks for the third item (Line $6-11$) is trivially bounded by the number of near-collisions $R^{(i)}_{lsh} \cdot N^{(i)}_{col}$ detected by Algorithm 2, and checking Hamming weight is done with $2w^{(i)}$ operations by storing the nonzero coordinates of each $x \in S^{(i)}$. □

C Experimental Validations

We briefly recall the flow of the analysis of Algorithm 3 in Section 5.2.

- The main components of success probability are $\Pr[E^{(i)}_{sp}]$ and $\Pr[E^{(i)}_{ncf}]$. Proposition 2 and Proposition 3 equipped with Heuristic 1 and Heuristic 3 lower bound them as

$$\Pr[E^{(i)}_{sp}] \ge \left(p_{sp}(\chi^{(i)}_{sp})\right)^{2^i} \text{ and } \Pr[E^{(i)}_{ncf}] \ge \left(p_{nc}(\chi^{(i)}_{ncf}) \cdot p_{ncf}(\ell^{(i)}; \chi^{(i)}_{ncf})\right)^{2^i}$$

  with some further elementary probability p_{sp} and p_{ncf}.
- Under uniformity and independence assumptions Heuristic 2 and Heuristic 4, the probabilities p_{sp} and p_{ncf} are further represented by Eq. (17) and Eq. (3) as

$$p_{sp}(\chi^{(i)}_{sp}) = \mathbb{E}_{\chi^{(i)}_{sp}}\left[1 - (1 - p_{rep}(e))^{R^{(i)}/2}\right],$$
$$p_{ncf}(\ell^{(i)}; \chi^{(i)}_{ncf}) \approx \mathbb{E}_{\chi^{(i)}_{ncf}}\left[1 - (1 - p_{lsh}(e))^{R^{(i)}_{lsh}}\right].$$

– The exact computation of $\mathbb{E}_\chi[1-(1-p(x))^R]$ requires excessively complicated integral. We then instead consider $1-(1-\mathbb{E}_{\chi^{(i)}}[p(x)])^R]$ for the relevant parameter settings, while expecting

$$\mathbb{E}_\chi[1-(1-p(x))^R] \approx 1-(1-\mathbb{E}_\chi[p(x)])^R]. \tag{31}$$

This section experimentally validates these arguments, over our parameter search range in Figure 6. First, we validate the uniformity heuristics in Appendix C.1. Then then argue that our parameter setting based on Eq. (31) still yields sufficiently large success probability in Appendix C.2. We then check the lower bound argument regarding the value of p_{sp} (and p_{ncf}) in Appendix C.3. All scripts for experiment can be found in Supplementary Material.

We also want to remark that the projection dimensions $r^{(i)}$ are set at most 100 for the parameter sets where our attack shows the best performance in Table 1. Thus we focus on $r^{(i)} \leq 100$ range in this section.

Simulation of the modulus matrix. In our main attack description, the modulus matrix $B \in \mathbb{R}^{m\times m}$ is assumed to be processed by BKZ with some β. For our experiment, instead of directly running BKZ, we generate B by simulating its Gram-Schmidt norm with GSA, and consider its representation under the Gram-Schmidt orthonormal basis $\tilde{B}^*$. More precisely, given modulus q and BKZ block-size β, we generate B by defining diagonal entries $B_{i,i}$ as i-th Gram-Schmidt norm determined by Eq. (1), and sampling $B_{i,j}$ from the range $[-B_{i,i}/2, B_{i,i}/2)$ for $j > i$.

C.1 Uniformity Modelling

In this section, we consider Heuristic 2 and Heuristic 4, which commonly assumes some discrete set S is uniformly and independently distributed in some domain D. We verify this by checking for any random area $\mathcal{A}$, it holds that

$$\mathcal{R}_{ideal} := \frac{\mathrm{vol}(D \cap \mathcal{A})}{\mathrm{vol}(D)} \approx \mathcal{R}_{real} := \frac{|S \cap \mathcal{A}|}{|S|}.$$

More precisely, our experiments generate the set $P_1(s_k^{(1)})$ and $L^{(1)}$ that corresponds to each heuristic, and plot the point $(\mathcal{R}_{ideal}, \mathcal{R}_{real})$ while randomly choosing the area $\mathcal{A}$.[13] Figure 7 shows the result, where the parameters defining each set are taken from the actual cost estimation in Section 7.

C.2 Exchange of Expectation

Recall that the choices of $R_{lsh}^{(i)}$ and $r^{(i)}$ in Figure 6 were for $1-(1-\mathbb{E}_\chi[p(x)])^R \approx 1-e^{-10}$. However, p_{ncf} and p_{sp} correspond to $\mathbb{E}_\chi[1-(1-p(x))^R]$, but we can only prove from Jensen's inequality

$$\mathbb{E}_\chi[1-(1-p(x))^R] \leq 1-(1-\mathbb{E}_\chi[p(x)])^R.$$

[13] More precisely, a random *cube* having nonempty intersection with the domain D

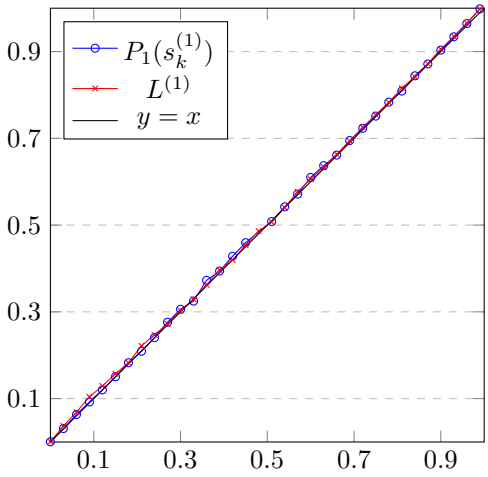


Fig. 7: Plots of $(\mathcal{R}_{ideal}, \mathcal{R}_{real})$ for each target set $P_1(s_k^{(1)})$ and $L^{(1)}$ of Heuristic 2 and Heuristic 4, with parameters $d = 8488, r^{(0)} = 46, r^{(1)} = 19, w^{(0)} = 20, w^{(1)} = 14, q = 2^{699}$ and $\beta = 300$.

Thus, our desired left-hand side can possibly be much smaller than $1 - e^{-10}$, so we cannot say p_{ncf} and p_{sp} are close to $1 - e^{-10}$ yet.

To resolve this issue, we experimentally measure $\mathbb{E}_{\chi}[1 - (1 - p(x))^R]$ to argue the value is also sufficiently large. More precisely, recall the inner functions $p_{rep}(x)$ and $p_{lsh}(x)$ in p_{sp} and p_{ncf} are of the form

$$p_{rep}(x) \propto \prod_{j=1}^{r^{(i+1)}} \left(1 - \frac{|x_j|}{2\ell^{(i+1)}}\right) \text{ and } p_{lsh}(x) \propto \prod_{j=1}^{r^{(i)}} \left(1 - \frac{|x_i|}{b_{lsh}^{(i+1)}}\right).$$

Furthermore, the corresponding distribution $\chi^{(i)}$ of each coordinate of x is as follows:

$$\chi^{(i)} = \begin{cases} \mathcal{G}_\sigma & \text{if } i = 0 \\ \mathcal{U}[-\ell^{(i)}, \ell^{(i)}] & \text{if } i \geq 1. \end{cases}$$

Considering our choice of $\ell^{(i)} = 6\sigma$ and $b_{lsh}^{(i)} = 12\sigma$, two inner functions $p_{rep}(x)$ and $p_{lsh}(x)$ can be unified as

$$p_r(x) := \prod_{j=1}^{r} \left(1 - \frac{|x_j|}{12\sigma}\right).$$

Finally, Figure 8 shows the values of $\mathbb{E}_{\chi^{(i)}}[1-(1-p_r(x))^R]$ with $R \approx 10/\mathbb{E}_{\chi^{(i)}}[p_r(x)]$ for some range of r, from which we could say at least p_{ncf} and p_{sp} is larger than at least 0.6 for $r \leq 100$. In fact, we use those values for p_{sp} and p_{ncf} to compute the success probability for cost estimation in Section 7.

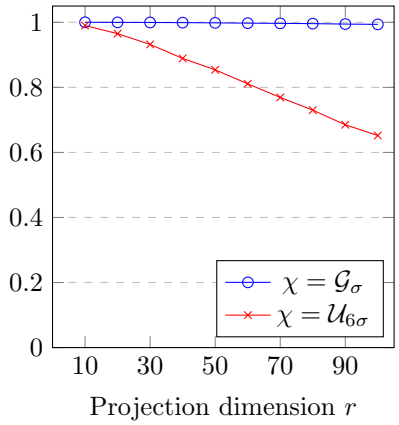


Fig. 8: Values of $\mathbb{E}_{x \leftarrow \chi}[1 - (1 - p_r(x))^R]$ where $R \approx 10/\mathbb{E}_{\chi^{(i)}}[p_r(x)]$. We use this value for the estimations in Section 7. The probabilities could become smaller for larger r, especially for the $\chi = \mathcal{U}_{6\sigma}$ case. However, the projection dimension $r^{(i)}$ is indeed set ≤ 100 in the optimal parameterization in Section 7 and hence this experiment faithfully shows what we need to validate.

C.3 Lower Bound to Product by Neglecting Relation

We now proceed to the validation of Heuristic 1 (Heuristic 3, resp) that lower bound the probability $\Pr[E^{(i)}_{sp}]$ ($\Pr[E^{(i)}_{ncf}]$, resp) by the product of $p_{sp}(\chi^{(i)}_{sp})$ ($p_{ncf}(\chi^{(i)}_{ncf})$, resp). We especially focus on Heuristic 1 that regards $E^{(i)}_{sp}$ below, since our choice $b^{(i)}_{lsh} = 2\ell^{(i-1)}$ makes the validation for $E^{(i)}_{ncf}$ is almost identical to $E^{(i)}_{sp}$.

First recall the event $E^{(i)}_{sp,k}$ that there is a $w^{(i+1)}$-rep pair (s_r, s'_r) for a target element $s^{(i)}_k$ such that

$$\begin{aligned}&([Ms^{(i+1)}_r]_{B,r^{(i+1)}}, [Ms'^{(i+1)}_r]_{B,r^{(i+1)}}) \in \mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}} \times \mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}} \\ &\Leftrightarrow ([Ms^{(i+1)}_r]_{B,r^{(i+1)}}, [Ms^{(i+1)}_r + e^{(i)}_k]_{B,r^{(i+1)}}) \in \mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}} \times \mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}}\end{aligned}$$

where $C^{(i+1)} := [-\ell^{(i+1)}, \ell^{(i+1)}]^{r^{(i+1)}}$ and $e^{(i)}_k = [Ms^{(i)}_k]_{B,r^{(i+1)}}$. Under Heuristic 2 that assumes the uniformity of $[Ms^{(i+1)}_r]$, we can show that

$$\Pr[E^{(i)}_{sp,k}] = \mathbb{E}_{e^{(i)}_k \leftarrow \chi^{(i)}_{sp}}[f(e^{(i)}_k)] \quad \text{where} \quad f(e^{(i)}_k) := 1 - (1 - p_{rep}(e^{(i)}_k))^{R^{(i)}/2}.$$

However, we cannot ensure

$$\Pr[E^{(i)}_{sp}] = \Pr\left[\bigwedge_{k \in [2^i]} E^{(i)}_{sp,k}\right] = \prod_{k \in [2^i]} \Pr[E^{(i)}_{sp,k}].$$

because every $e^{(i)}_k$ are mutually dependent due to the representation equation (Eq. (16)). In this regard, Heuristic 1 says that we can at least lower bound the probability, which we validate below.

For simplicity, we focus on only $i \leq 2$ cases since our estimation finds the best performance with $t \leq 3$. However, we believe the experiments also provide some intuitions for higher level, along with our justification of Heuristic 1.

The case of $i = 1$. In this case, we need to claim the following.

$$\Pr[E^{(1)}_{sp,1} \wedge E^{(1)}_{sp,2}] \geq \Pr[E^{(1)}_{sp,1}] \cdot \Pr[E^{(1)}_{sp,2}].$$

For that, we investigate the expected value of $\prod_{k=1}^{2} f(e^{(1)}_k)$ for the following cases.

- $\mathsf{Exp}_1(1)$: Two vectors $e^{(1)}_1, e^{(1)}_2 \in \mathcal{C}^{(1)}$ have a relation $e^{(0)} = e^{(1)}_1 - e^{(1)}_2$ where $e^{(0)} \leftarrow \mathcal{G}_\sigma$.
- $\mathsf{Exp}_1(2)$: Two vectors $e^{(1)}_1, e^{(1)}_2$ are independently sampled in $\mathcal{C}^{(1)}$.

For the actual experiment, we simulate the parameter search range of Figure 6. Precisely, we fix every norm bound $\ell^{(i)} = 6\sigma$. Then for each dimension $r^{(1)}$, we choose the exponent $R^{(1)}$ so that $R^{(1)}/2 \approx 10\mathbb{E}_{\chi^{(1)}}[p_r(x)]$. The resulting graph is given by Figure 9a, and the fact $\mathsf{Exp}_1(1) \geq \mathsf{Exp}_1(2)$ validates Heuristic 1.

The case of $i = 2$. In this case, we need to validate the following inequality about the four events

$$\Pr[E^{(2)}_{sp}] \geq \Pr[E^{(2)}_{sp,1}] \cdot \Pr[E^{(2)}_{sp,2}] \cdot \Pr[E^{(2)}_{sp,3}] \cdot \Pr[E^{(2)}_{sp,4}].$$

We investigate the expected value of $\prod_{k=1}^{4} \left(1 - (1 - p_{rep}(e^{(2)}_k))^{R^{(2)}/2}\right)$ for the following cases.

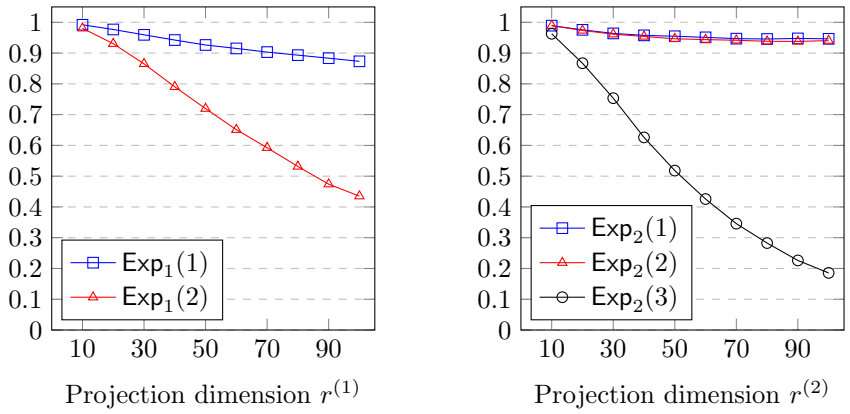
- $\mathsf{Exp}_2(1)$: Four vectors $e^{(2)}_k \in \mathcal{C}^{(2)}$ have relations
$$e^{(0)} = e^{(1)}_1 - e^{(1)}_2, e^{(1)}_1 = e^{(2)}_1 - e^{(2)}_2 \text{ and } e^{(1)}_2 = e^{(2)}_3 - e^{(2)}_4$$
where $e^{(0)} \leftarrow \mathcal{G}_\sigma$, and $e^{(1)}_1, e^{(1)}_2 \in \mathcal{C}^{(1)}$.
- $\mathsf{Exp}_2(2)$: Identical with $\mathsf{Exp}_2(1)$ except that $e^{(1)}_1$ and $e^{(1)}_2$ are independently sampled from $\mathcal{C}^{(1)}$.
- $\mathsf{Exp}_2(3)$: Four vectors $e^{(2)}_k$ are independently sampled from $\mathcal{C}^{(2)}$.

For the actual experiment, we again simulate the parameter search range of Figure 6. Precisely, we fix every norm bound $\ell^{(i)} = 6\sigma$. Then for each dimension $r^{(2)}$, we choose the exponent $R^{(2)}$ so that $R^{(2)}/2 \approx 10\mathbb{E}_{\chi^{(2)}}[p_r(x)]$.

The result is given as Figure 9b. It can be directly checked that $\mathsf{Exp}_2(3) \geq \mathsf{Exp}_2(1)$ validates Heuristic 1. More interestingly, recall that our justification of Heuristic 1 (Appendix B.8) consists of two sub-arguments

- Argument 1: $\Pr[\wedge_{k\in[2^i]} E^{(i)}_{sp,k}] = \prod_{k\in[2^{i-1}]} \Pr[E^{(i)}_{sp,2k-1} \wedge E^{(i)}_{sp,2k}]$
- Argument 2: $\Pr[E^{(i)}_{sp,2k-1} \wedge E^{(i)}_{sp,2k}] \geq \Pr[E^{(i)}_{sp,2k-1}] \cdot \Pr[E^{(i)}_{sp,2k}]$.

In this regard, $\mathsf{Exp}_2(1) \approx \mathsf{Exp}_2(2)$ and $\mathsf{Exp}_2(2) \geq \mathsf{Exp}_2(3)$ validates Argument 1 and Argument 2 respectively, which makes our justification of Heuristic 1 more reliable.



(a) $i = 1$ case

(b) $i = 2$ case

Fig. 9: Validation of $\Pr[E_{sp}^{(i)}] \geq \prod_{k\in[2^i]} \Pr[E_{sp,k}^{(i)}]$ (Heuristic 1). The real situation corresponds to $\mathsf{Exp}_1(1)$ ($\mathsf{Exp}_2(1)$, respectively), and theoretic lower bound used for the cost estimation in Section 7 corresponds to $\mathsf{Exp}_1(2)$ ($\mathsf{Exp}_2(3)$, respectively)

C.4 Full Implementation of Meet-LWE

The most involved part of this paper is definitely the description of Algorithm 3 and corresponding analysis in Section 5.2. To convince the correctness and analysis, we implement a proof-of-concept level of that algorithm. To recall, let $M \in \mathbb{Z}^{m\times m}$ and $B \in \mathbb{Z}^{m\times m}$ such that there exists a solution $s \in \mathcal{T}^d(h)$ satisfying $Ms = e \bmod B$ for some small e sampled from some $\mathcal{G}_\sigma$. Then the goal of Algorithm 3 is to find the solution $s \in \mathcal{T}^d(h)$.

We examine our implementation with the following toy parameters: $m = 25, d = 20, h = 12, \sigma = 1$, and the modulus matrix B is determined by the last Gram-Schmidt norm $q_l = 30$ with root-Hermite factor $\delta_0 = 1.05$. We further choose the split weights $(w^{(1)}, w^{(2)}) = (8, 4)$ for level $t = 2$, and $(w^{(1)}, w^{(2)}, w^{(3)}) = (10, 6, 3)$ for level $t = 3$, and the other attack parameters are chosen almost identically to Figure 6, except $C_{proj} = C_{lsh} = 2$.

Running 100 executions for level $t = 2$ and $t = 3$, we observe 82 and 81 successes for $t = 2$ and $t = 3$ experiments respectively. This fact immediately indicates that our proposed algorithm is indeed work. Meanwhile, computed from our theoretic lower bound of success probability of Theorem 1, we have 0.06 and 0.002 for each case, which means that the lower bound is quite loose. In particular, as the numbers in Table 1 obtained from the theoretic lower bound estimation, we expect that more refined analysis will reduce the cost estimation, and leave it as a future work.