A Hybrid of Lattice-reduction and Meet-LWE via Near-Collision on Babai's Plane

Minki Hhan[1†], Jiseung Kim[2†], Changmin Lee[3†], Yongha Son[4*]

[1]University of Texas at Austin, USA.
[2]Jeonbuk National University, South Korea.
[3]Korea University, South Korea.
[4]Sungshin Women's University, South Korea.

*Corresponding author(s). E-mail(s): yongha.son@sungshin.ac.kr;
Contributing authors: minki.hhan@austin.utexas.edu;
jiseungkim@jbnu.ac.kr; changminlee@korea.ac.kr;
†These authors contributed equally to this work.

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 the 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, 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.

We finally estimate the concrete cost of our attack for sparse LWE keys. For some currently deployed parameter sets in fully homomorphic encryption libraries, our attack achieves up to **10** bits improvement than the previous primal hybrid attacks.

Keywords: Learning With Errors, Hybrid lattice attack, Meet-in-the-middle

1 Introduction

The (search) learning with errors (LWE) problem [1] asks to find the secret key s given $(A, b) \in \mathbb{Z}_q^{m\times(n+1)}$ such that $b = As+e \bmod q$ for a small error vector e. The theoretical hardness of the LWE problem is well established, 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. Practical LWE-based schemes uses other parameters far from the provable range for efficiency. In particular, the secret key size has a great effect on efficiency, so many LWE-based cryptosystems choose the binary/ternary secret keys [2–4] or even sparse keys [5–8].

As the parameters are outside the provable range, the concrete security is measured against known attacks that solves the underlying LWE parameters [9]. The use of small secret vector also opens new possibilities in the adversary's perspective, and the combinatorial attack is one of the natural approaches to exploit this feature. The Meet-in-the-middle (MitM) strategy due to Odlyzko [10] that finds the secret key in time $S^{0.5}$ has remained the best combinatorial algorithm for a long time, beyond the naive brute-force search that takes S time.

As a breakthrough on this line, May [11] proposes a new combinatorial attack called Meet-LWE based on the representation technique, originally introduced in the context of subset sum-type problems [12–14]. 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. The combinatorial attacks including Meet-LWE themselves cannot beat the lattice attacks (e.g., [15, 16]) in the practical parameter setting.

The *hybrid* attack refers to 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 [17], 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 [17–21]. As Meet-LWE improves over MitM, the hybrid attack that combines the lattice reduction algorithm and Meet-LWE is expected to have some implications on the concrete hardness of LWE. This combination was already considered in the original paper of May, but it appears that the direct application of Meet-LWE face some difficulties [11, Section 10]. The hybrid attack combining lattice reduction and Meet-LWE remains a central open problem in this landscape of LWE attacks, which is the main subject of this paper.

This work. We propose a hybrid attack of lattice reduction and Meet-LWE, resolving the open question of [11]. In a general hybrid attack, the lattice reduction phase converts the original LWE equation of the form $b = As + e \bmod q$ into another equation of the form $\mathsf{NP}_B(Ms') = e'$ where $\mathsf{NP}_B(\cdot)$ is a Babai's nearest plane algorithm. The

underlying algebraic structure of the latter equation becomes quite complicated compared to the original LWE equation over integers, which is the main technical difficulty of combining combinatorial strategy with lattice. This work, in this respect, identifies and resolves the technical challenges that occurs when adapting the original Meet-LWE's idea into this algebraic structure, whose overview is presented below in Section 3.

To examine the impact of this attack on the practical LWE-based cryptosystems, we also estimate the concrete cost of our new hybrid attack. From this, we observe that our attack beats all the previous attacks for some currently deployed parameters in FHE library [22, 23]; see Table 2 for the estimation results, and refer Section 8.3 for more discussions.

2 Preliminaries

For positive integers n, we denote the set $\{1, \ldots, n\}$ by $[n]$. For a real number a and integer q, we write by $[a]_q$ the unique real number in $[-q/2, q/2)$ such that q divides $a - [a]_q$. The uniform distribution over a set S is denoted by $\mathcal{U}(S)$. For any distribution χ, we denote a sampling from χ by $x \leftarrow \chi$, but in particular sampling from $\mathcal{U}(S)$ is simply denoted by $x \leftarrow S$. Let $\rho_c(x) := \exp(-\pi x^2/c^2)$. We denote the (centered) *continuous* Gaussian distribution with standard deviation σ by $\mathcal{G}_\sigma$ defined by the probability density function proportional to $\rho_{\sigma/\sqrt{2\pi}}(x)$. The (centered) *discrete* Gaussian distribution over $\mathbb{Z}$ is denoted by $\mathcal{D}_{\mathbb{Z},r}$, and defined by the probability mass function $\rho_r(x)/\rho_r(\mathbb{Z})$ for every $x \in \mathbb{Z}$.

A simpler formulation of LWE. Our main interest is in solving the LWE equation $b = As + e \bmod q$ where the error vector e is sampled from some discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z},r}$. We consider the following formulation in our paper: Given a matrix $M \in \mathbb{Z}_q^{m \times n}$, find $s \in \mathbb{Z}_q^n$ such that

$$Ms = e \bmod q$$

for an (unknown) error vector e. The LWE equation $b = As' + e \bmod q$ can be viewed as this problem, by defining $M = (A| - b)$ and $s = (s'|1)$.

2.1 Ternary Secrets

We write the set of n-dimensional ternary vectors with w nonzero entries by

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

The number of nonzero entries w is also called hamming weight. Each element $s \in \mathcal{T}^n(w)$ can be represented by a sum of lower hamming weight ternary vectors:

Definition 1 For a vector $s \in \mathcal{T}^n(w)$, we call a pair $(s_r, s'_r) \in \mathcal{T}^n(w') \times \mathcal{T}^n(w')$ such that $s = s_r - s'_r$ by a weight-w' *representation pair*, or a w'-rep pair of x.

[1] Our definition of $\mathcal{T}^n(w)$ is a bit different from [11]. It defined $\mathcal{T}^n(w)$ by the set of vectors having w numbers of 1 and w numbers of -1.

The following lemma computes the number of w-rep pairs of $s \in \mathcal{T}^n(w)$.

Lemma 1 (Adapted from [18, Lemma 4.1]) *Let w be an even integer and w' be an integer such that $w' \geq w/2$. For any $s \in \mathcal{T}^n(w)$, the number of w'-rep pairs of s is*

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

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

2.2 Meet-in-the-Middle and May's Meet-LWE

To solve $Ms = e \bmod q$ with $M \in \mathbb{Z}_q^{m \times n}$, Meet-in-the-middle (MitM) strategy considers a w'-rep pair $(s_1, s_2) \in \mathcal{T}^n(w')$ such that $s = s_1 - s_2$ and

$$Ms_1 - Ms_2 = Ms = e \bmod q.$$

Since e is small, this equation implies that Ms_1 and Ms_2 are *near-collision* modulo q. This enables one to recover s by finding near-collision in the set $L := \{Mx \bmod q \mid x \in \mathcal{T}^n(w')\}$. Instead of searching the entire $\mathcal{T}^n(w')$, the classic meet-in-the-middle [10] constructs a random subset $S \subset \mathcal{T}^n(w')$ of size about $|\mathcal{T}^n(w')|/\sqrt{R}$ for the number of w'-rep pairs R, so that $S \times S$ includes a single representation of s with high probability.

Meet-LWE [11] utilizes these R representations in a more novel way. It 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 $\pi_r([Mx]_q) = e$; precisely

$$S_i = \{x \in \mathcal{T}^n(w') \mid \pi_r(Mx \bmod q) = e_i\} \text{ for } i \in \{1, 2\} \tag{1}$$

where $e_1 \in \mathbb{Z}^{r/2} \times 0^{r/2}, e_2 \in 0^{r/2} \times \mathbb{Z}^{r/2}$ are proper guessing for $\pi_r(e)$ such that $\pi_r(e) = e_1 - e_2$. The projection dimension r is chosen so that a single representation pair survives in $S_1 \times S_2$, namely $r \approx \log_q(R)$. This yields $|S_i| \approx |\mathcal{T}^n(w')|/R$, which is smaller than $|\mathcal{T}^n(w')|/\sqrt{R}$ of the MitM approach.

It should be remarked that the attack succeeds only when the initial error guessing e_1, e_2 is correct, i.e., $e_1 - e_2 = \pi_r(e)$. The original Meet-LWE enumerates all candidates $\pi_r(e)$ to address this. As the projection dimension r is not so large, this enumeration step and the overall attack is asymptotically efficient. On the other hand, the concrete cost for enumeration is not negligibly small, and [11] indeed leaves it as an open question to avoid this error enumeration step. Later, [24] suggested an approach

that avoids the error enumeration step. However, we found their algorithm is flawed; see Section A for details.

2.3 Lattices and Basis

A lattice is a discrete subgroup of $\mathbb{R}^m$. For a (full-rank) matrix $B = (b_1|\dots|b_n) \in \mathbb{R}^{m\times n}$, 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 $\mathcal{L}(B)$.

Gram-Schmidt basis. Given a basis $B \in \mathbb{R}^{m\times n}$, we denote the Gram-Schmidt orthogonal matrix of B by $B^* = (b_1^*|\cdots|b_n^*) \in \mathbb{R}^{m\times m}$, precisely

$$b_1^* = b_1, \quad b_i^* = b_i - \sum_{1\le j<i} \frac{\langle b_i, b_j^*\rangle}{\|b_j^*\|^2} \cdot b_j^* \text{ for } 2 \le i \le n.$$

For the upper triangular matrix $R_B = \left[\langle b_i, b_j^*\rangle / \|b_j^*\|^2\right]_{i,j\in[n]}$, $B = B^* \cdot R_B$ holds. We also consider the normalized matrix of B^*, say $\bar{B}^* = (\bar{b}_1^*|\cdots|\bar{b}_n^*) \in \mathbb{R}^{n\times n}$ where $\bar{b}_i^* = b_i^*/\|b_i^*\|$. We write the basis change map from the standard coordinates to the Gram-Schmidt basis by $\zeta_{\bar{B}^*} : \mathbb{R}^n \to \mathbb{R}^n$. For $v = (v_1, \cdots, v_n)$ represented in the standard coordinate, $\zeta_{\bar{B}^*}(v)$ corresponds to one represented in the coordinate system determined by $\bar{B}^*$. The map $\zeta_{\bar{B}^*}$ can be computed as

$$\zeta_{\bar{B}^*}(v) = (\bar{B}^*)^{-1}v \quad \text{or} \quad \zeta_{\bar{B}^*}(v)_i = \langle v, \bar{b}_i^*\rangle. \tag{2}$$

Lattice reduction and Geometric Series Assumption (GSA). 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 [25] that is parameterized by a block size β, denoted by BKZ-β. The Geometric Series Assumption (GSA) predicts that the Gram-Schmidt norms $\|b_i^*\|$ of the basis B reduced by BKZ-β geometrically decreases, whose concrete value is estimated as follows $\|b_i^*\| = \delta_0^{-2(i-1)+n-1} \cdot \det(B)^{\frac{1}{n}}$ 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 nearest plane (Babai's NP) algorithm considers a representation of v with respect to the Gram-Schmidt orthonormal basis $\bar{B}^*$, and then reduces all components into $[-1/2, 1/2)$. By writing the fundamental parallelepiped

$$\mathcal{P}(B) := \{Bx \in \mathbb{R}^m \ : x \in [-1/2, 1/2)^n\},$$

Lemma 2 summarizes the classic fact of Babai's NP algorithm.

Lemma 2 *Given an input vector $t \in \mathbb{R}^n$ and basis $B \in \mathbb{R}^{n\times n}$, Babai's NP algorithm outputs a vector $\mathsf{NP}_B(t)$ in $\mathcal{P}(B^*)$. Further, $\mathsf{NP}_B(t)$ is the unique vector in $\mathcal{P}(B^*)$ such that $t - \mathsf{NP}_B(t) \in \mathcal{L}(B)$.*

2.4 Primal Hybrid Strategy

To solve the equation $Ms = e \bmod q$ with $M \in \mathbb{Z}_q^{m\times n}$ and $s \in \mathcal{T}^n(w)$, the primal lattice attack considers a lattice defined by the basis matrix $P = \begin{pmatrix} qI_m & M \\ & I_n, \end{pmatrix}$ which contains a short vector $(e|s) = P\cdot(x|s)$ for some integral vector $x \in \mathbb{Z}^m$. To hybrid this with combinatorial strategy, we consider some guessing dimension $d \le n$, and divide the matrix M into $M = (M_0|M_1)$ by the first $n-d$ and the last d columns, and the secret key s into $(s_0|s_1)$ according to M_0 and M_1. In this view, writing $m' = m+n-d$, we can re-write the basis matrix P by

$$P = \begin{pmatrix} B & M' \\ & I_d \end{pmatrix} \text{ where } M' = \begin{pmatrix} M_1 \\ O \end{pmatrix} \in \mathbb{Z}^{m'\times d} \text{ and } B = \begin{pmatrix} qI_m & M_0 \\ & I_{n-d} \end{pmatrix} \in \mathbb{Z}^{m'\times m'}. \quad (3)$$

Letting $e' = (e|s_0)$ and $k = (x|s_0)$, we can write $(e'|s_1) = P\cdot(x|s)$ into $B\cdot k + M'\cdot s_1 = e$. By applying the Babai's Nearest Plane (NP) algorithm denoted by $\mathsf{NP}_B(\cdot)$, we have $\mathsf{NP}_B(M's_1) = \mathsf{NP}_B(e')$. Assuming that the matrix B is sufficiently reduced (with respect to lattice reduction), we have a good chance of $\mathsf{NP}_B(e') = e'$. This finally yields a new goal that finds s_1 satisfying $\mathsf{NP}_B(M's_1) = e'$, and then we might apply combinatorial strategies to find s_1. We formally write this process by Algorithm 1 for later use.

Algorithm 1: Primal hybrid conversion

Input: A matrix $M \in \mathbb{Z}_q^{m\times n}$ and a guessing dimension $0 < d \le n$

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 Split the matrix M into $(M_0|M_1) \in \mathbb{Z}_q^{m\times(n-d)} \times \mathbb{Z}_q^{m\times d}$

2 **return** $M' = \begin{pmatrix} M_1 \\ O \end{pmatrix} \in \mathbb{Z}^{m'\times d}$ and $B = \begin{pmatrix} qI_m & M_0 \\ & I_{n-d} \end{pmatrix} \in \mathbb{Z}_q^{m'\times m'}$

Normalizing the new error vector. Note that the new error vector $e' = (e|s_0)$ in the equation $\mathsf{NP}_B(M_1s_1) = e'$ is in a bit weird shape: the former part e' comes from a Gaussian distribution and the latter part s_0 is sparse ternary. This can be (heuristically) normalized as follows: Many primal attacks utilized *scaling factor* [26, 27] $\nu > 0$ to define $P_\nu = \begin{pmatrix} qI_{m'} & M_0 \\ & \nu I_n \end{pmatrix}$, which yields the error vector $e'_\nu = (e'|\nu s_0)$. With a proper choice of ν, [18] assumed that e'_ν with respect to the Gram-Schmidt basis $\bar{B}^*$, i.e., $\zeta_{\bar{B}^*}(e'_\nu)$ follows the input Gaussian distribution.[2]

The probability of $\mathsf{NP}_B(e') = e'$. It should be remarked that the event $\mathsf{NP}_B(e') = e'$, equivalently $e' \in \mathcal{P}(B^*)$ does not always happen, even if the basis B is highly reduced. Regarding this, many previous works [18, 19, 29] computed the probability $p_{NP} := \Pr[e' \in \mathcal{P}(B^*) \mid e' \leftarrow \mathcal{D}_{\mathbb{Z},\sqrt{2\pi}\sigma}]$, over an idealized model that assumes the

[2] This assumption is (implicitly) used when estimating previous hybrid attack [17] in lattice-estimator [28].

discrete Gaussian $\mathcal{D}_{\mathbb{Z},\sqrt{2\pi}\sigma}$ (where the error e is sampled) behaves like the continuous Gaussian $\mathcal{G}_\sigma$, as follows

$$p_{NP} = \prod_{i=1}^{m} \mathrm{erf}\left(\frac{\|b_i^*\|}{2\sqrt{2}\sigma}\right). \tag{4}$$

3 Technical Overview

This section presents how to apply Meet-LWE to the equation $\mathsf{NP}_B(Ms) = e$ that is the main goal of the primal hybrid attack as reviewed in Section 2.4.

3.1 Projection for Primal Hybrid

The core idea of Meet-LWE is to utilize the projection map π_r to reduce the searching space. However, the fundamental reason that [11] leaves the application of Meet-LWE to primal hybrid as open is an absence of appropriate projection π_r that is compatible to $\mathsf{NP}_B(Ms) = e$ as noted in [11, Section 10]. Our starting point is to design a proper projection map π_r compatible with NP_B equation.

As the first step, since $\mathsf{NP}_B(v)$ lies in the fundamental parallelepiped $\mathcal{P}(B^*)$, it would be natural to switch our view to the coordinate system determined by $\bar{B}^*$, a Gram-Schmidt orthonormal basis. We have a natural projection map with respect to some coordinates in the Gram-Schmidt basis. However, an arbitrary r coordinate projection over $\bar{B}^*$-coordinates is generally incompatible with $\mathsf{NP}_B(\cdot)$, due to the non-orthogonality of B; see Figure 1. Because of this problem, it is unclear how to *write* the projection constraint as in Eq. (1).



(a) NP_B then project (b) Project then NP_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 $\mathsf{NP}_B(\cdot)$ does not commute.

Fortunately, the projection to the *last* r coordinates yields a well-defined projection map compatible with NP_B, say

$$\mathsf{NP}_{B,r}(v) := \pi_{\bar{B}^*,r}(\mathsf{NP}_B(v)) \in \mathcal{P}(B_r^*)$$

where $\pi_{\bar{B}^*,r}$ is the projection onto the last r coordinate in the Gram-Schmidt basis. This comes from the property of Gram-Schmidt process that R-factor of the last r vectors of basis B is identical to the projection of the full basis B; See Figure 2 for a visual explanation.

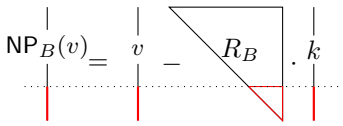

Fig. 2: $\mathsf{NP}_B(v)$ represented over $\bar{B}^*$-coordinate system. LHS shows NP_B then last r coordinate projection, and RHS shows last r coordinate then NP_B.

The formal description for the above idea can be found in Section 4, together with some further observations on this projection $\mathsf{NP}_{B,r}(\cdot)$ which are used for concrete attack cost estimations.

3.2 Meet-LWE with NP Projection

Equipped with the concept of $\mathsf{NP}_{B,r}$ above, we can now *define* the projection-based constraint. As the first attempt toward Meet-LWE, one can naively mimic the original Meet-LWE by considering two sets

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

and guessing e_1, e_2 such that $\mathsf{NP}_{B,r^{(1)}}(e) = e_1 - e_2$ with $w^{(1)} \geq w^{(0)}/2$. At this point, however, another problem arises. 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 $r^{(1)}$. In our case, 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 3, which makes this error enumeration infeasible.

We solve this problem by considering *approximate constraints* of the form $\|\mathsf{NP}_{B,r^{(1)}}(Mx)\|_\infty \approx 0$; more precisely, we define a set

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

for some small $r^{(1)}$ and $\ell^{(1)}$.

Recall that the projection dimension r in Meet-LWE was set so that there is only a single representation pair. Similarly, we set the parameters $r^{(1)}$ and $\ell^{(1)}$ so that a single representation $(s_1^{(1)}, s_2^{(1)})$ of s is in $S^{(1)} \times S^{(1)}$; Figure 4 graphically explains this idea. The relevant computations to concrete choices of $r^{(1)}$ and $\ell^{(1)}$ are presented in the main body Section 6.

With such proper choices of $r^{(1)}$ and $\ell^{(1)}$, we can expect a single representation pair $(s_1^{(1)}, s_2^{(1)})$ in $\mathcal{S}^{(1)}$. Clearly it holds that $\mathsf{NP}_B(Ms_1^{(1)}) = \mathsf{NP}_B(Ms_2^{(1)} + e)$, and hence we can expect $\mathsf{NP}_B(Ms_1^{(1)}) - \mathsf{NP}_B(Ms_2^{(1)}) = e$ with a good chance.[3] Therefore, one can recover the solution s by computing $L^{(1)} := \{\mathsf{NP}_B(Mx) : x \in S^{(1)}\}$ and finding a *near-collision* in $L^{(1)} \times L^{(1)}$.

[3] In fact, this is not obvious because of the non-orthogonality of B. Therefore, we need to correctly account for the relevant probability. A similar issue was already the main technical hurdle of the seminal primal hybrid attack [17], considered under the name *admissiblity* [17, 19, 20]; we also correctly reflect this in the analysis.

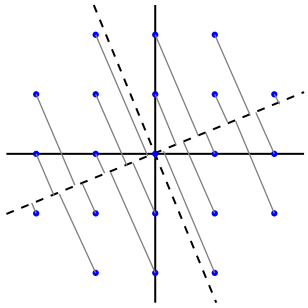

Fig. 3: The blue points denote the integer coordinate points with respect to the standard basis. The dotted line represents the coordinate system determined from $\tilde{B}^*$, and the gray lines show the projection $\tau_{B,r}$.

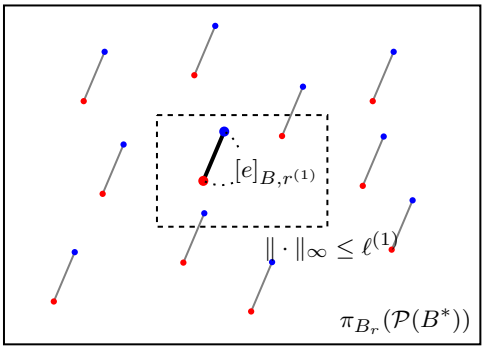

Fig. 4: The blue and red points represent $\mathsf{NP}_{B,r^{(1)}}(Ms_r)$ and $\mathsf{NP}_{B,r^{(1)}}(Ms'_r)$ respectively, which are at distance $\pi_{\bar{B}^*,r}(e)$. We need to set $r^{(1)}$ and $\ell^{(1)}$ so that a representation $(s_1^{(1)}, s_2^{(1)})$ that makes both points have infinity norm less than $\ell^{(1)}$.

Recursion to higher levels. Our proposed attack comes from recursion of this idea to higher level, whose informal description is given in Algorithm 2. Observe that the original problem to solve $\mathsf{NP}_B(Ms) = e$ is almost equivalent to construct

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

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

$$S^{(i)} := \{x \in \mathcal{T}^d(w^{(i)}) \mid \|\mathsf{NP}_{B,r^{(i)}}(Mx)\|_\infty \le \ell^{(i)}\}, \tag{5}$$

where the top level t is chosen so that $|S^{(t)}|$ becomes sufficiently small and it is recoverable by exhaustive search.

3.3 Near-Collision Finding from Locality Sensitive Hashing

Our attack needs to find near-collisions to recover a list $S^{(i-1)}$ from a list $S^{(i)}$. Any near-collision finding algorithm can be exploited, but the concrete efficiency of the deployed algorithm greatly affects the total attack cost. We adapt the near-collision finding algorithm based on locality sensitive hashing (LSH) introduced in the boosted hybrid attack on NTRU [20]. The analysis of the LSH algorithm in our context is nontrivial because of the following two differencesf from [20].

The first point is in the shape of error vector; our interest problem LWE considers the closeness defined by Gaussian, rather than fixed weight ternary vectors (of

Algorithm 2: Meet-LWE for matrix modulus (Informal)

Input: Matrices $M \in \mathbb{Z}^{m\times d}$ and $B \in \mathbb{Z}^{m\times m}$
Params : List parameters $(w^{(i)}, r^{(i)}, \ell^{(i)})$
Output: The solution s such that $\|[Ms]_B\| \le \ell_0$.

Construct the top-level set $S^{(t)} := \{s^{(t)} \in \mathcal{T}^d(w^{(t)}) \mid \|\mathsf{NP}_{B,r^{(t)}}(Ms^{(t)})\|_\infty \le \ell^{(t)}\}$
for $i = t$ down to $i = 1$ **do**
  Compute $L^{(i)} = \{\mathsf{NP}_{B,r^{(i-1)}}(Ms^{(i)}) \mid s^{(i)} \in S^{(i)}\}$
  Construct $S^{(i-1)} = \{s^{(i-1)} \in \mathcal{T}^d(w^{(i-1)}) \mid \|\mathsf{NP}_{B,r^{(i-1)}}(Ms^{(i-1)})\|_\infty \le \ell^{(i-1)}\}$ by investigating near-collisions in $L^{(i)} \times L^{(i)}$
end
return $S^{(0)}$

NTRU). Thus we revise the analysis of [20] with respect to Gaussian distribution. In particular, we extensively use the classic technique [30] that uses multiple LSH functions to amplify the success probability.[4] Combined with Gaussian distribution, the analysis of the amplification becomes even much complicated. We are not aware of relevant arguments in the literature.

Second, the target situation where we apply LSH is different. [20] considered a domain $\mathcal{P}(Q)$ with an orthogonal basis Q (in fact, $q \cdot I$ for I identity matrix). This enables to understand any near-collision pair in a torus manner. That is, any near-collision pair (v, w) in $\mathcal{P}(Q)$ satisfies $\|v - w \bmod q\|_\infty$ is small, which provides many convenience in analysis. For our attack, we need to consider more general cases where the given basis B is not orthogonal anymore, and we cannot use the torus interpretation. More precisely, as our target domain is $\mathcal{P}(\mathcal{B}^*)$ determined by the Gram-Schmidt basis $\mathcal{B}^*$, the non-orthogonality of B makes the near-collision pair should be strictly near, that is, $||v - w||_\infty$ should be small. To correctly deal with this, we revise the previous torus-case analysis to our case. More details and the resulting analysis can be found in Section 5.

4 Matrix Modulus and Projection

In this section, we introduce a matrix modulus notation 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 [11] into the primal hybrid case.

Matrix modulus and representatives. For the sake of readability, we introduce a matrix modulus notation. Given any basis $B \in \mathbb{R}^{m\times m}$, we can define the equivalence relation $a = b \bmod \mathcal{L}(B)$ since $\mathcal{L}(B)$ is a subgroup of $\mathbb{R}^m$, which leads to a natural definition of modulo *matrix* B. Furthermore, by rephrasing Lemma 2, we can set the representative element $[v]_B$ of each coset $v + \mathcal{L}(B)$ by the output of Babai's NP algorithm:

[4] [20] also mentioned this technique and implicitly used it. However, there was no need to seriously discuss this in their case.

Definition 2 For a matrix $B \in \mathbb{R}^{m \times m}$ and vectors $v, w \in \mathbb{R}^m$, we define modulo B equivalence relation by $v = w \bmod B \iff v - w \in \mathcal{L}(B)$. Further, we define the representative of v modulo B by $v \bmod B = [v]_B := \mathsf{NP}_B(v) \in \mathcal{P}(B^*)$, where $\mathsf{NP}_B(v)$ is the output of Babai's NP algorithm on input B and v.

Gram-Schmidt coordinate system. As mentioned in Section 3.1, we switch our view to the coordinate system determined by $\bar{B}^*$. To be more rigorous, let $B_{\mathsf{std}} \in \mathbb{Z}^{m \times m}$ be a basis whose column vectors are represented in the standard coordinate. Recall that from Eq. (2) for a vector $v \in \mathbb{R}^m$ represented in the standard coordinates, $\zeta_{\bar{B}^*}(v) = (\bar{B}^*)^{-1}(v)$ represents the coordinates of the same vector v in the $\bar{B}^*$-coordinates. In particular, considering the QR-decomposition $B_{\mathsf{std}} = \bar{B}^* \cdot R$ obtained by Gram-Schmidt process, (each column of) B_{std} represented in the $\bar{B}^*$-coordinate corresponds to (each column of) the upper triangular matrix $R \in \mathbb{R}^{m \times m}$. In other words, considering a (dense) basis B_{std} and then switching to $\bar{B}^*$-coordinates is actually equivalent to considering an upper triangular basis $B \in \mathbb{R}^{m \times m}$. Thus we may consider *an upper triangular* basis without loss of generality, so our projection concept for the matrix modulus below also focuses on an upper triangular basis.

Remark 1 In fact, it is possible to write every content of this section for a general matrix B_{std}, using the decomposition $B_{\mathsf{std}} = \bar{B}^* \cdot R$ and some projected lattice concept. However, our restriction for upper triangle matrix is not just for the sake of simplicity, but it really suffices for our main attack description in Section 6, since it considers every vector in $\bar{B}^*$-coordinates.

Projection of matrix modulus. We define the projection map onto the last coordinates, to make it compatible with modulo B operation.

$$\pi_r : (x_1, \ldots, x_m) \mapsto (x_{m-r+1}, \ldots, x_m).$$

Additionally, we also define $\pi_r(M)$ for a matrix $M = (M_1 | \cdots | M_d) \in \mathbb{R}^{m \times d}$ by the column-wise application, i.e. $\pi_r(M) := (\pi_r(M_1) | \cdots | \pi_r(M_d)) \in \mathbb{R}^{r \times d}$. Then we have (a sort of) commutativity between the modulo B and π_r.

Lemma 3 *Let $B \in \mathbb{R}^{m \times m}$ be a full-rank upper-triangular matrix, and B_r be the lower-right submatrix of B of size $r \times r$. Then it holds that $\pi_r([v]_B) = [\pi_r(v)]_{B_r}$.*

Proof By definition, $[v]_B$ is in $\mathcal{P}(B^*)$, which is equals to $\prod_{i=1}^m \left[-\frac{\|b_i^*\|}{2}, \frac{\|b_i^*\|}{2}\right)$ since B is upper-triangular. Then it holds that $\pi_r(\mathcal{P}(B^*)) = \mathcal{P}(B_r^*)$, and hence we have $\pi_r([v]_B) \in \pi_r(\mathcal{P}(B^*)) = \mathcal{P}(B_r^*)$. Meanwhile, $[v]_B = v - B \cdot k$ for some integral vector $k \in \mathbb{Z}^m$, and thanks to the linearity of π_r, we have

$$\pi_r([v]_B) = \pi_r(v - B \cdot k) = \pi_r(v) - \pi_r(B) \cdot k = \pi_r(v) - B_r \cdot \pi_r(k)$$

where the last equality holds because B is upper-triangular. From the rightmost term, we know that $\pi_r([v]_B) = \pi_r(v) \bmod B_r$. Since $[\pi_r(v)]_{B_r} = \mathsf{NP}_{B_r}(\pi_r(v))$ is defined as the unique element in $\mathcal{P}(B_r^*)$ such that equals to $\pi_r(v)$ modulo B_r, we conclude $\pi_r([v]_B) = [\pi_r(v)]_{B_r}$. □

Thanks to Lemma 3, we have a well-defined projected matrix modulus.

Definition 3 Let $B \in \mathbb{R}^{m\times m}$ be a full-rank upper-triangular matrix, and B_r be the lower-right submatrix of B of size $r \times r$. We define the projected matrix modulus by

$$[v]_{B,r} := \pi_r([v]_B) = [\pi_r(v)]_{B_r} \in \mathcal{P}(B_r^*).$$

Splitting matrix modulus. The hybrid attack often deals with elements of the form $[v+e]_B$ for some small e such that $[e]_B = e$. It is expected that $[v+e]_B = [v]_B + e$ thanks to such smallness of e, but this is not generally true due to the non-orthogonality of B. The previous hybrid attacks [17, 19] resolved this by enforcing the addition of e incurs no modulo B reduction, *i.e.* $[v]_B + e \in \mathcal{P}(B^*)$, which is dubbed as *admissiblity* event [17, Def. 3] ([19, Def. 4.2], resp). This condition can be slightly mitigated for $[v+e]_{B,r}$.

Lemma 4 *Let $B \in \mathbb{R}^{m\times m}$ be a full-rank upper-triangular matrix, and let $v, e \in \mathbb{R}^m$ such that $[e]_{B,r} = \pi_r(e)$. If $[v]_{B,r} + \pi_r(e) \in \mathcal{P}(B_r^*)$, it holds that*

$$[v+e]_{B,r} = [v]_{B,r} + \pi_r(e).$$

Proof Writing $[v]_B = (v_1, \ldots, v_m)$ and $e = (e_1, \ldots, e_m)$, then the given condition is written as $|v_i + e_i| \le \|b_i^*\|/2$ for every $i \in [m-r+1, m]$. In general, it holds that $[v+e]_B = [v]_B + [e]_B + \sum_{i=1}^m c_i \cdot b_i^*$ for $c_i \in \mathbb{Z}$. Here, the assumption implies that $c_i = 0$ for $i \in [m-r+1, m]$, and we have $[v+e]_B = [v]_B + [e]_B + \sum_{i=1}^{m-r} c_i \cdot b_i^*$. From the linear property of $[\cdot]_{B,r}$ and an identity $[[v]_B]_{B,r} = [v]_{B,r}$, it holds that $[v+e]_{B,r} = [v]_{B,r} + [e]_{B,r} + \sum_{i=1}^{m-r} c_i \cdot [b_i^*]_{B,r}$. Since $[b_i^*]_{B,r} = 0$ for every $i \in \{1, \ldots, m-r\}$, this gives the desired result. □

5 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 [20, 31, 32]. This algorithm is called the *near-collision finding (NCF) algorithm*, and will be used as a subroutine in our main attack in Section 6. Throughout this section, we fix $\mathcal{Q}$ by a zero-centered r-dimensional hypercube having each coordinate length q_i, namely $\mathcal{Q} := \prod_{i=1}^r [-q_i/2, q_i/2)$. We define the near-collision in $\mathcal{Q}$ as follows:

Definition 4 We say that an (ordered) pair $(y, y') \in \mathcal{Q} \times \mathcal{Q}$ is an $(\mathcal{Q}, \ell)$-*near-collision* if $\|y - y'\|_\infty \le \ell$ and $y \ne y'$. When the domain $\mathcal{Q}$ is clear from the context, we simply call it by an ℓ-*near-collision*.

Remark 2 For the readers familiar with [20], note that our target domain $\mathcal{Q}$ and the corresponding definition of near-collision are not defined in a torus-coordinate manner, unlike [20] that considered coordinate-wise (discrete) *torus* over $\mathbb{Z}_q^n$. This is because we will apply this NCF algorithm on $\mathcal{P}(B_r^*)$ in Section 6 where the non-orthogonality of basis B prevents us from considering $\mathcal{P}(B_r^*)$ as a coordinate-wise torus.

Torus-LSH. Our NCF algorithm is based on a variant of *the torus LSH* functions adapted from [20]. The basic idea is to divide each coordinate interval $[-q_i/2, q_i/2)$ into the equal-length bin of length b, while understanding each coordinate interval as a torus and the interval division is randomized with different starting points denoted by $c \in \mathcal{Q}$.

Definition 5 Let $b \in \mathbb{N}$, $n_i := \max\{1, \lfloor q_i/b \rfloor\}$, and $b_i := q_i/n_i$ for each $i \leq [r]$. Then for a block-length parameter b, the family $\mathcal{H}(\mathcal{Q}; b)$ of torus LSH functions $h_c : \mathcal{Q} \to \prod_{i\in[r]} \mathbb{Z}_{n_i}$ is defined as follows:

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

Our NCF algorithm. A torus-LSH function h_c can be used to find near-collision by labeling each element y in L by $h_c(y)$, and then checking whether each pair of elements with the same label is ℓ-near-collision. In fact, our NCF algorithm described as Algorithm 3 further utilizes the classic technique [30] that uses many hash functions, in order to amplify the success probability. More precisely, the label-then-check procedure is repeated R times with different choices of h_c to find more near-collisions.

Algorithm 3: LSH-based near-collision finding algorithm

Input: A domain $\mathcal{Q}$, a list $L \subset \mathcal{Q}$, 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 $\mathcal{H} = \mathcal{H}(\mathcal{Q}; b)$
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, y') in $T(\mathsf{addr}) \times T(\mathsf{addr})$ **do**
      if $\|y - y'\|_\infty \leq \ell$, **then** $C \leftarrow C \cup \{(y, y')\}$
    end
  end
end
return C

Models for input list. For a rigorous analysis of our NCF algorithm, we need to clarify the shape of the input a list L and the target near-collision pairs that we want to find from L.

First, consider the near-collision pair $([Ms_1]_B, [Ms_2]_B)$ stemming from the equation $[Ms]_B = e$ and $s = s_1 - s_2$. This near-collision pair would be unusually close one in the list $\{[Ms]_B \mid s \in \mathcal{T}_d(w)\}$. Regarding this, we introduce a model where the input list L has such unusual near-collision pair; in particular the distance of the pair is sampled from small (continuous) Gaussian $\mathcal{G}_\sigma$.

Model 1 (One special $\mathcal{G}_\sigma$-near-collision model). *The list L is defined by the union $L' \cup \{y, y'\}$ where a pair $(y, y') \in \mathcal{Q}$ is sampled as $y \leftarrow \mathcal{U}(\mathcal{Q})$, and $y = y' + e$ with $e \leftarrow \mathcal{G}_\sigma$, and each element of L' is uniformly sampled from $\mathcal{U}(\mathcal{Q})$ and independent of any other element in L'. We call the pair (y, y') by the special $\mathcal{G}_\sigma$-near-collision. When L in this model is given as an input of NCF algorithm, we say the success probability as the probability of finding the pair (y, y').*

Second, we also need to consider the case where the elements in L are uniformly and independently distributed in the domain $\mathcal{Q}$, which will correspond to the higher level case for our Meet-LWE generalization in Section 6. Note that this model is analogous to the well-distributed assumptions used in [12–14]. In this case, we have no special target element unlike Model 1, so any *random* ℓ-near-collision is our interest.

Model 2 (Purely random model). *Each element in L is uniformly sampled from $\mathcal{U}(\mathcal{Q})$ and independent of any other element in L. When L in this model is given as an input of NCF algorithm, we say the success probability as the probability of finding a randomly chosen ℓ-near-collision pair (y, y').*

Analysis. First, the following lemma can be easily shown from the definition.

Lemma 5 *Let $y, y' \in \mathcal{Q}$ and $e = (e_1, \cdots, e_r) := y - y'$ such that $\|e\|_\infty \leq b_i$. Then the probability that (y, y') collides by a random torus-LSH function is*

$$p_{lsh}(\mathcal{Q}, e, b) := \Pr_{h_c \leftarrow \mathcal{H}(\mathcal{Q};b)}[h_c(y) = h_c(y')] = \prod_{\substack{1 \leq i \leq r \\ n_i > 1}} \left(1 - \frac{|e_i|}{b_i}\right).$$

Proof It suffices to see each i-th coordinate of y and y'. They are mapped into the same image by h_c with the probability $1 - |e_i|/b_i$ if $n_i > 1$ (and 1 if $n_i = 1$ because it always mapped into 0). □

Then the following proposition provides the analysis of Algorithm 3 under the models we established, where the full proposition and the detailed proof can be found in Section B.

Proposition 6 (Simplified) *Let $\mathcal{Q} = \prod_{i=1}^{r} [-q_i/2, q_i/2)$ be an r-dimensional domain, $L \subset \mathcal{Q}$ 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 4. 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. Then, Algorithm 3 terminates in time*

$$R \cdot \left(|L| \cdot T_{\mathsf{hash}}(r) + \frac{|L|^2}{2} \cdot \left(\prod_{1 \leq i \leq r} \frac{1}{n_i} \right) \cdot T_{\mathsf{check}}(r) \right), \tag{6}$$

using $|L|$ words of space for hash function outputs with overwhelming probability.

Writing

$$p_{ncf}(\mathcal{Q}, \chi, \ell, b) := \Pr_{e \leftarrow \chi}[\|e\|_\infty \leq \ell] \cdot \mathop{\mathbb{E}}_{e \leftarrow \chi}\left[1 - (1 - p_{lsh}(\mathcal{Q}, e, b))^R\right] \tag{7}$$

for some r-dimensional distribution χ, where $p_{lsh}(\mathcal{Q}, e, b)$ defined by Lemma 5, the success probability of this NCF algorithm is

- *$p_{ncf}(\mathcal{Q}, \mathcal{G}_\sigma, \ell, b)$ when L follows Model 1, and*
- *$p_{ncf}(\mathcal{Q}, \mathcal{U}([-\ell, \ell]^r), \ell, b)$ when L follows Model 2.*

Sketch For the time complexity, $|L| \cdot T_{\mathsf{hash}}(r)$ is the time for the for loop in Line 6. The expected number of collisions found by the hash function in Line 8 is less than $\frac{|L|^2}{2} \cdot \left(\prod_{1 \le i \le r} \frac{1}{n_i}\right)$, which corresponds to the second term. As in the standard LSH complexity, the time to check the false positive is relatively small.

For the success probability, suppose that the interested pair (the special near collision for Model 1 and a randomly chosen ℓ-near collision for Model 2) has the difference e. Then, a single iteration finds the target pair with probability $p_{lsh}(\mathcal{Q}, e, b)$, and R iteration finds the target pair with probability at least $1 - (1 - p_{lsh}(\mathcal{Q}, e, b))^R$. By carefully taking the expectation, we obtain the success probability as in the proposition.

□

6 Meet-LWE for Matrix Modulus

Recall again from Section 2.4, primal hybrid attacks are given two matrices $M \in \mathbb{Z}^{m \times d}$ and $B \in \mathbb{Z}^{m \times m}$, and aim to find $s \in \mathcal{T}^d(w^{(0)})$ such that

$$\mathsf{NP}_B(Ms) = e \text{ or } [Ms]_B = e^{5}$$

where e is sampled from Gaussian distribution $\mathcal{G}_\sigma$.

This section describes a generalization of Meet-LWE [11] to matrix modulus that solves $[Ms]_B = e$. Before start, as the projection $[\cdot]_{B,r}$ defined as Definition 3 regards the $\bar{B}^*$-coordinates, we will work on the representation of M, B and e with respect to the Gram-Schmidt orthonormal basis $\bar{B}^*$, rather than the standard coordinates. To clarify more, writing the matrices obtained from Algorithm 1 by M_{std}, B_{std}, and the corresponding error vector e_{std} to emphasize that they are represented in the standard coordinates, this section considers the matrices $M := \zeta_{\bar{B}^*}(M_{\mathsf{std}})$, $B := \zeta_{\bar{B}^*}(B_{\mathsf{std}})$ and the error vector $e := \zeta_{\bar{B}^*}(e_{\mathsf{std}})$, where $\zeta_{\bar{B}^*}$ is the coordinate change map from the standard coordinates to the $\bar{B}^*$-coordinates defined as Eq. (2). Also note that B is now upper-triangle, so we can use the lemmas regarding projected matrix modulus in Section 4.

The Algorithm 4 describes our attack, which is a full-detailed version of Algorithm 2. Some explanations for notable differences with Algorithm 2 follows below.

Distinction of $\tilde{S}^{(i)}$ and $S^{(i)}$. Unlike the informal version Algorithm 2, we distinguish the set obtained during the attack by $\tilde{S}^{(i)}$ from the full set $S^{(i)} = \{x \in \mathcal{T}^d(w^{(i)}) \mid \|[Mx]_{B,r^{(i)}}\| \le \ell^{(i)}\}$. This is because it is extremely hard to fully recover the set $S^{(i)}$ during the attack process, due to several failure events such as near-collision finding. Thus, the attack usually finds some proper subset of $S^{(i)}$, which we denote by $\tilde{S}^{(i)}$. Nonetheless, our attack can recover the desired solution s even if we cannot fully construct $S^{(i)}$, which is in fact the key argument of the success probability analysis in Section 6.2.

No constraint on the top-level set. Although the core idea of this attack is to construct a lower-level set by running NCF algorithm on the upper-level set, the top-level set $S^{(t)}$ has no further upper level set, so that $S^{(t)}$ should be constructed directly. Then, even though $S^{(t)}$ is defined with some constraint, say $\|[Mx]_{B,r^{(t)}}\|_\infty \le \ell^{(t)}$ for some $r^{(t)}$ and $\ell^{(t)}$, one needs to check whether every element $x \in \mathcal{T}^d(w^{(i)})$ satisfies the constraint. That is, the construction of $S^{(t)}$ takes $|\mathcal{T}^d(w^{(i)})|$, which means that the constraint provides no benefit on complexity. Therefore, the top-level set $S^{(t)}$ is particularly defined by a random subset of $\mathcal{T}^d(w^{(i)})$ of fixed size N_t that is given as a parameter, which can be found at Line 1 of Algorithm 4.

[5] This implicitly assumes $[e]_B = e$. The probability of this event is already discussed in Section 2.4, and we assume this holds throughout this section.

Algorithm 4: Meet-LWE for matrix modulus with the top level t

Input: Matrices $M \in \mathbb{R}^{m \times d}$ and $B \in \mathbb{R}^{m \times m}$
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 $w^{(1)}, \cdots, w^{(t)}$,
Constraint parameters $(r^{(1)}, \ell^{(1)}), \cdots, (r^{(t-1)}, \ell^{(t-1)})$,
Top-level set size N_t,
LSH block-lengths $b_{lsh}^{(1)}, \cdots, b_{lsh}^{(t)}$,
NCF repetition numbers $R_{lsh}^{(1)}, \cdots, R_{lsh}^{(t)}$

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 $\mathcal{T}^d(w^{(t)})$ of size N_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 $\tilde{L}^{(i)} = \{[Mx]_{B,r^{(i-1)}} \mid x \in \tilde{S}^{(i)}\}$
4 Invoke LSH-NCF$(\mathcal{P}(B^*_{r^{(i-1)}}), L^{(i)}, \ell^{(i-1)}; b_{lsh}^{(i)}, R_{lsh}^{(i)})$ to find a set of $\ell^{(i-1)}$-near-collisions in $\tilde{L}^{(i)}$, say $\mathsf{NearCols}^{(i)}$
  `// Check the weight condition for each near-collision`
5 Set $\tilde{S}^{(i-1)} = \emptyset$
6 **for** each pair $([Ms_r]_{B,r^{(i-1)}}, [Ms'_r]_{B,r^{(i-1)}}) \in \mathsf{NearCols}^{(i)}$ **do**
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** $\tilde{S}^{(0)}$

Remark 3 (Relation with the previous hybrid attack) When applying this top-level optimization, our attack with top-level $t = 1$ exactly corresponds to the original Howgrave-Graham's hybrid attack [17, 19]: Both attacks consider a random subset of $\mathcal{T}^d(w^{(1)})$ and aim to find one representation pair $(s_1, s_2) \in \mathcal{T}^d(w^{(1)}) \times \mathcal{T}^d(w^{(1)})$ of the desired solution s such that $[Ms_1]_B$ and $[Ms_2]_B$ are close enough.

6.1 Timing Cost Analysis

The timing cost of each i-th level of the main loop of Algorithm 4 consists of the following steps:

- Line 3: Computing $L^{(i)} := \{[Mx]_{B,r^{(i-1)}} \mid x \in \tilde{S}^{(i)}\}$
- Line 4: Running LSH-NCF over $L^{(i)}$, with LSH repetition number $R_{lsh}^{(i)}$ and LSH block-size $b_{lsh}^{(i)}$ on dimension $r^{(i-1)}$ domain.
- Line 5-8: Checking hamming weight for each near-collision pair.

Clearly, the second item clearly dominates the i-th loop's timing cost, and hence we conclude the overall timing cost of Algorithm 4 as

$$T_{meet} = \sum_{i=1}^{t} T_{ncf}(R_{lsh}^{(i)}, |\tilde{S}^{(i)}|, r^{(i-1)}, b_{lsh}^{(i)}) \tag{8}$$

where $T_{ncf}(\cdot)$ is the timing cost for running NCF, whose concrete computation is given as Eq. (6). To estimate the size of $\tilde{S}^{(i)}$, the top level set size is given as a parameter N_t, and for the intermediate level set, we use a trivial bound $|S^{(i)}|$, which can be easily estimated using volume ratio; e.g. $|S^{(1)}| = |\mathcal{T}^d(w^{(1)})| \cdot (2\ell^{(1)})^{r^{(1)}}/\mathrm{vol}(\mathcal{P}(B^*_{r^{(1)}}))$.

6.2 Success Probability Analysis

This section provides a success probability analysis of Algorithm 4. First of all, considering the definition of $\tilde{S}^{(0)}$, the success probability is simply written as

$$p_{meet} := \Pr[s \in \tilde{S}^{(0)}].$$

The main idea of analysis is to recursively lift the probability p_{meet} to a higher level, which is a formalization of the high-level story in Section 3.2.

Remark 4 This section would be the most technical and complicated part of this paper. In the main body, we mainly focuses on the top level $t = 2$ case, which would be even better on understanding the core idea. Full analysis with an arbitrary top level t can be found in Section C.

6.2.1 Lifting from Level-0 to Level-1.

The starting point is splitting the event $s \in \tilde{S}^{(0)}$ into the following two events:

$E^{(1)}_{sp}$: There exists a $w^{(1)}$-rep pair of $s^{(0)}_1$ in $\tilde{S}^{(1)}$, say $(s^{(1)}_1, s^{(1)}_2)$.

$E^{(1)}_{nc}$: NCF succeeds to find $([Ms^{(1)}_1]_B, [Ms^{(1)}_2]_B)$,

from which we have

$$\Pr[s^{(0)}_1 \in \tilde{S}^{(0)}] \geq \Pr[E^{(1)}_{sp} \wedge E^{(1)}_{nc}] = \Pr[E^{(1)}_{sp}] \cdot \Pr[E^{(1)}_{nc} \mid E^{(1)}_{sp}] \tag{9}$$

The first event $E^{(1)}_{sp}$ might seem to correspond to the situation what Figure 4 depicts, but it should be more rigorous, since the set $\tilde{S}^{(1)}$ could be different with $S^{(1)}$. To deal with this, we further split the event $E^{(1)}_{sp}$ as:

$E^{(1)}_{ex}$: There exists a $w^{(1)}$-pair of $s^{(0)}_1$ in $S^{(1)}$, say $(s^{(1)}_1, s^{(1)}_2)$.

$E^{(1)}_{suc}$: Both elements $s^{(1)}_1$ and $s^{(1)}_2$ are found in the algorithm process, by the successive set construction down-to level-1.

which gives

$$\Pr[E^{(1)}_{sp}] = \Pr[E^{(1)}_{ex} \wedge E^{(1)}_{suc}] = \Pr[E^{(1)}_{ex}] \cdot \Pr[E^{(1)}_{suc} \mid E^{(1)}_{ex}] \tag{10}$$

By the definition of $\tilde{S}^{(1)}$, the probability $\Pr[E^{(1)}_{suc} \mid E^{(1)}_{ex}]$ can be written as

$$\Pr[E^{(1)}_{suc} \mid E^{(1)}_{ex}] = \Pr[s^{(1)}_1 \in \tilde{S}^{(1)} \text{ and } s^{(1)}_2 \in \tilde{S}^{(1)}]. \tag{11}$$

Putting Eqs. (9) to (11) together, we have

$$\Pr[s^{(0)}_1 \in \tilde{S}^{(0)}] \geq \Pr[E^{(1)}_{nc} | E^{(1)}_{sp}] \cdot \Pr[E^{(1)}_{ex}] \cdot \Pr[s^{(1)}_1 \in \tilde{S}^{(1)} \text{ and } s^{(1)}_2 \in \tilde{S}^{(1)}],$$

where

- $\Pr[E^{(1)}_{nc} | E^{(1)}_{sp}]$ is the NCF success probability, computed using Proposition 6.
- $\Pr[E^{(1)}_{ex}]$ corresponds to what Figure 4 shows, computed in below Lemma 7.

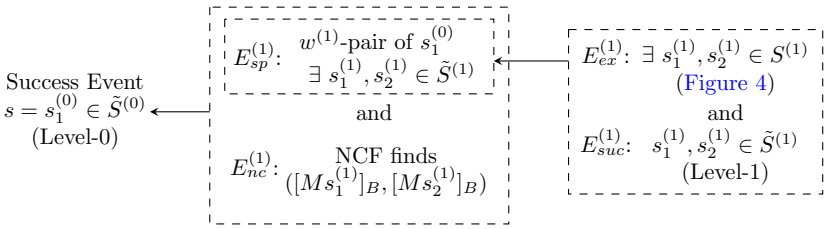

Fig. 5: High-level flow of level-0 to level-1 reduction

- $\Pr[s_1^{(1)} \in \tilde{S}^{(1)} \text{ and } s_2^{(1)} \in \tilde{S}^{(1)}]$ resembles to our starting point $\Pr[s_1^{(0)} \in \tilde{S}^{(0)}]$, which enables to apply the same lifting to higher level.

The whole process is summarized in Figure 5.

$E_{ex}^{(1)}$: Existence of a rep-pair in $S^{(1)}$. By recalling the definition of level-1 constraint $\|[Mx]_{B,r^{(1)}}\| \le \ell^{(1)}$, this event is precisely what Figure 4 exactly depicts; there exists $s_1^{(1)}, s_2^{(1)} \in \mathcal{T}^d(w^{(1)})$ such that $s_1^{(0)} = s_1^{(1)} - s_2^{(1)}$ and $[Ms_1^{(1)}]_{B,r^{(1)}}, [Ms_2^{(1)}]_{B,r^{(1)}} \in [-\ell^{(1)}, \ell^{(1)}]^{r^{(1)}}$. The following lemma computes this probability, while assuming $[Ms]_{B,r^{(1)}}$ as a uniform sample x from $\mathcal{P}(B^*_{r^{(1)}})$ for each $w^{(1)}$-pair (s, s') of $s_1^{(0)}$; note that the other one $[Ms']_{B,r^{(1)}}$ is automatically determined as $[x+e]_{B,r^{(1)}}$.

Lemma 7 *Assume that the pair $([Ms]_{B,r^{(1)}}, [Ms']_{B,r^{(1)}})$ is modelled as $(x, [x+e]_{B,r^{(1)}})$ with $x \leftarrow \mathcal{P}(B^*_{r^{(1)}})$, for a random $w^{(1)}$-rep-pair (s, s') of $s_1^{(0)}$. Then the probability of $E_{ex}^{(1)}$ is*

$$p_{ex}^{(1)} = \mathop{\mathbb{E}}_{e \leftarrow \mathcal{G}_\sigma} \left[1 - (1 - p_{rep}(e))^{R^{(1)}}\right] \tag{12}$$

where $p_{rep}(e) := \dfrac{(2\ell^{(1)})^{r^{(1)}}}{\mathrm{vol}(\mathcal{P}(B^*_{r^{(1)}}))} \cdot \prod_{i \in [r^{(1)}]} \left(1 - \dfrac{|e_i|}{2\ell^{(1)}}\right)$ and $R^{(1)}$ is the number of $w^{(1)}$-rep pairs of $s_1^{(0)} \in \mathcal{T}^d(w^{(1)})$.

Proof We temporarily omit the superscript in $r^{(1)}, \ell^{(1)}, R^{(1)}$ in the proof. For a w-pair (s, s') of $s_1^{(0)}$, we let p_{rep} be the probability of $[Ms]_{B,r}, [Ms']_{B,r} \in \mathcal{C}_\ell^r$. Thanks to our assumption on $([Ms]_{B,r}, [Ms']_{B,r})$, this probability only depends on the error term e, and can be computed as

$$\begin{aligned} p_{rep}(e) &= \Pr_{x \leftarrow \mathcal{P}(B^*_r)} \left[x, [x+e]_{B,r} \in \mathcal{C}_\ell^r\right] \\ &= \Pr_{x \leftarrow \mathcal{P}(B^*_r)} \left[x \in \mathcal{C}_\ell^r\right] \cdot \Pr_{x \leftarrow \mathcal{P}(B^*_r)} \left[[x+e]_{B_r} \in \mathcal{C}_\ell^r \mid x \in \mathcal{C}_\ell^r\right]. \end{aligned}$$

The first probability is clearly $\mathrm{vol}(\mathcal{C}_\ell^r)/\mathrm{vol}(\mathcal{P}(B_r^*)) = (2\ell)^r/\mathrm{vol}(\mathcal{P}(B_r^*))$. The second probability expects that a random element $x \in \mathcal{C}_\ell^r$ still remains in $\mathcal{C}_\ell^r$ after adding a vector $e = (e_1, \ldots, e_r)$. Looking at each i-th coordinate, we expect that $x_i + e_i$ lies in $[-\ell, \ell)$ where $x_i \leftarrow [-\ell, \ell)$, whose probability is exactly $1 - \frac{|e_i|}{2\ell}$, and the second probability becomes $\prod_{i \in [r]} \left(1 - \frac{|e_i|}{2\ell}\right)$.

As there are R numbers of w-pair in total, the probability of (at least) one w-pair satisfies the level-1 constraint is $1-(1-p_{rep}(e))^R$, for a fixed error e. Finally, since the error e is sampled from $\mathcal{G}_\sigma$, we have

$$p_{ex}^{(1)} = \int \Pr[\chi_{sp}^{(i)} = e]\left(1-(1-p_{rep}(e))^{R^{(i)}}\right) de = \mathop{\mathbb{E}}_{e\leftarrow\chi_{sp}^{(i)}}\left[1-(1-p_{rep}(e))^{R^{(i)}}\right]$$

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

$E_{nc}^{(1)}$: Near-collision finding. By the event $E_{sp}^{(1)}$, we guarantee that $\tilde{S}^{(1)}$ has at least one $w^{(1)}$-rep pair, say $(s_1^{(1)}, s_2^{(1)})$. Here we proceed to the probability that the algorithm actually finds them in a near-collision finding, which is exactly what $\Pr[E_{nc}^{(1)} \mid E_{sp}^{(1)}]$ means.

At first glance, this might seem to correspond exactly to the LSH-NCF success probability of Proposition 6. However, it requires a bit more rigor: Recall that our NCF algorithm only regards a coordinate-wise near pair; see Definition 4. Meanwhile, although $s_1^{(0)} = s_1^{(1)} - s_2^{(1)}$ implies $Ms_1^{(1)} - Ms_2^{(1)} = Ms_1^{(0)} = e \bmod B$, the modulo B pair $([Ms_1^{(1)}]_B, [Ms_2^{(1)}]_B)$ could not be coordinate-wisely near due to the non-orthogonality of B. Thus, we actually need to ensure that $[Ms_1^{(1)}]_B = [Ms_2^{(1)}]_B + e$ without modulo B so that $([Ms_1^{(1)}]_B, [Ms_2^{(1)}]_B)$ becomes a near-collision. To summarize, the event $([Ms_1^{(1)}]_B, [Ms_2^{(1)}]_B) \in \mathsf{NearCols}^{(1)}$ is split into:

(a) $(y_1^{(1)}, y_2^{(1)}) = ([Ms_1^{(1)}]_B, [Ms_2^{(1)}]_B)$ is coordinate-wisely near.

(b) NCF algorithm (in line 4) succeeds to find the near-collision $(y_1^{(1)}, y_2^{(1)})$.

The first event (a) is identical to what the previous hybrid works [17, 19] considered under the name *admissibile* probability, say

$$p_{adm}(\mathcal{Q}, \chi) := \Pr_{x\leftarrow\mathcal{Q}, e\leftarrow\chi}[x+e \in \mathcal{Q}].$$

As the distance e is assumed to follow (continuous) Gaussian, the probability of this event can be computed as $p_{adm}(\mathcal{P}(B^*), \mathcal{G}_\sigma)$, assuming that $[Ms_1^{(1)}]_B$ is uniformly distributed over $\mathcal{P}(B^*)$ whose computation is known as [18, Lemma 4.2]: For any real (closed) interval I_b of length b,

$$p_{adm}\left(I_b, \mathcal{G}_\sigma\right) = \mathrm{erf}(X) + \frac{e^{-X^2}-1}{\sqrt{\pi}X} \text{ where } X = \frac{b}{\sqrt{2}\sigma}. \tag{13}$$

For the second event (b), given that $(y_1^{(1)}, y_2^{(1)})$ is indeed a (coordinate-wise) near-collision in $\mathcal{P}(B^*)$, we can directly utilize the analysis for LSH-NCF (Proposition 6), with a proper modelling of the target list $\tilde{L}^{(1)} = \{[Mx]_B \mid x \in \tilde{S}^{(1)}\}$: Since we are interested in only one near-collision $(y_1^{(1)}, y_2^{(1)})$ of distance e from (continuous) Gaussian, one special $\mathcal{G}_\sigma$-near-collision model (Model 1) makes sense.

$$p_{nc}^{(1)} = p_{adm}(\mathcal{P}(B^*), \mathcal{G}_\sigma) \cdot p_{ncf}(\mathcal{P}(B^*), \mathcal{G}_\sigma, \ell^{(0)}, b_{lsh}^{(1)}). \tag{14}$$

6.2.2 Lifting from Level-1 to Level-2.

So far, we show the following inequality from Eq. (9) and Eq. (10)

$$p_{meet} \geq p_{ex}^{(1)} \cdot p_{suc}^{(1)} \cdot p_{nc}^{(1)} \tag{15}$$

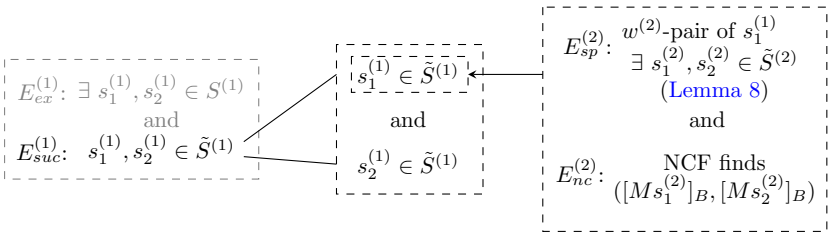

Fig. 6: High-level flow of level-1 to level-2 (top level) reduction

where $p^{(1)}_{ex}$ and $p^{(1)}_{nc}$ can be concretely computed by Lemma 7 and Eq. (14), respectively. We then proceed to lift the remaining probability

$$p^{(1)}_{suc} = \Pr[s^{(1)}_1 \in \tilde{S}^{(1)} \text{ and } s^{(1)}_2 \in \tilde{S}^{(1)} \mid E^{(1)}_{sp}]$$

to level-2, and Figure 6 shows the overview. For a simplicity, we omit the condition $E^{(1)}_{sp}$ when describing the probabilities, which ensures the existence of the target rep pair $s^{(1)}_1, s^{(1)}_2$ satisfying level-1 constraint.

We first take a similar approach to the level-0 to level-1 lifting on each probability $\Pr[s^{(1)}_1 \in \tilde{S}^{(1)}]$ (and also $\Pr[s^{(1)}_2 \in \tilde{S}^{(1)}]$). Specifically, we have

$$\Pr[s^{(1)}_1 \in \tilde{S}^{(1)}] \geq \Pr[E^{(2)}_{sp}] \cdot \Pr[E^{(2)}_{nc} \mid E^{(2)}_{sp}] =: p^{(2)}_{sp} \cdot p^{(2)}_{nc}$$

where

$E^{(2)}_{sp}$: There exists a $w^{(2)}$-rep pair of $s^{(1)}_1$ in $\tilde{S}^{(1)}$, say $(s^{(1)}_1, s^{(1)}_2)$.

$E^{(2)}_{nc}$: The $\ell^{(1)}$-near-collision pair $([Ms^{(2)}_1]_B, [Ms^{(2)}_2]_B)$ is in $\mathsf{NearCols}^{(2)}$.

For the first probability $p^{(2)}_{sp}$, note that level-2 is now the top level, so that the set $\tilde{S}^{(2)}$ is just a random subset of $\mathcal{T}^d(w^{(2)})$ without any constraint. In this case, the computation of $p^{(2)}_{sp}$ is just a counting argument.

Lemma 8 $\Pr[E^{(2)}_{sp}] \geq 1 - \exp(-R^{(2)} \cdot \left(N_2/|\mathcal{T}^d(w^{(2)})|\right)^2)$ *where* $R^{(2)}$ *is the number of* $w^{(2)}$*-rep pairs of* $s^{(1)}_1 \in \mathcal{T}^d(w^{(1)})$.

Proof Let (s, s') be a $w^{(2)}$-pair of $s^{(1)}_1$. Recalling that the top-level set $\tilde{S}^{(t)}$ is defined as a random subset of $\mathcal{T}^d(w^{(2)})$ of fixed size N_2, the probability of both (s, s') are included in $\tilde{S}^{(2)}$ is $p = \left(N_2/|\mathcal{T}^d(w^{(2)})|\right)^2$. Since we have $R^{(2)} := R(d, w^{(2)}, w^{(1)})$ numbers of rep pairs, we conclude the desired probability is $1 - (1-p)^{R^{(2)}}$. Finally, $(1-p)^{R^{(2)}} \leq e^{-R^{(2)} \cdot p}$ for $p \in [0, 1]$ ends proof. □

For the second probability $p^{(2)}_{nc}$, the overall story is similar to the level-0 to level-1 case. We first consider the admissible probability that $[Ms^{(2)}_1]_{B,r^{(1)}} = [Ms^{(2)}_2]_{B,r^{(1)}} + e^{(1)} \bmod B$ where $e^{(1)} = [Ms^{(1)}_1]_{B,r^{(1)}}$, and then apply the analysis result of LSH-NCF (Proposition 6).

However, the distribution of the error term $e^{(1)} = [Ms_1^{(1)}]_{B,r^{(1)}}$ now becomes the uniform distribution over $\mathcal{C}_{\ell^{(1)}}^{r^{(1)}} := [-\ell^{(1)}, \ell^{(1)}]^{r^{(1)}}$, which makes the admissible probability by $p_{adm}(\mathcal{P}(B^*), \mathcal{U}(\mathcal{C}_{\ell^{(1)}}^{r^{(1)}}))$, whose computation is quite immediate:

$$p_{adm}\left(I_b, \mathcal{U}([-\ell, \ell])\right) = 1 - \ell/(2b) \tag{16}$$

for any real (closed) interval I_b of length b.

Furthermore, as our target near-collision pair is not anymore the unusually short one, $\tilde{L}^{(2)} = \{[Mx]_{B,r^{(1)}} : x \in \tilde{S}^{(2)}\}$ is now regarded as the purely random model (Model 2), and hence finally we have

$$p_{nc}^{(2)} = p_{adm}(\mathcal{P}(B^*_{r^{(1)}}), \mathcal{U}(\mathcal{C}_{\ell^{(1)}}^{r^{(1)}}) \cdot p_{ncf}(\mathcal{P}(B^*_{r^{(1)}}), \mathcal{U}(\mathcal{C}_{\ell^{(1)}}^{r^{(1)}}), \ell^{(1)}, b_{lsh}^{(2)}) \tag{17}$$

Dependency of $s_1^{(1)} \in \tilde{S}^{(1)}$ and $s_2^{(1)} \in \tilde{S}^{(1)}$. The above argument can also be applied to $\Pr[s_2^{(1)} \in \tilde{S}^{(1)}]$ with the exactly same computation, so that we have

$$\Pr[s_1^{(1)} \in \tilde{S}^{(1)}] = \Pr[s_2^{(1)} \in \tilde{S}^{(1)}] \geq p_{sp}^{(2)} \cdot p_{nc}^{(2)}.$$

Here, it tempts to compute $\Pr[s_1^{(1)} \in \tilde{S}^{(1)}$ and $s_2^{(1)} \in \tilde{S}^{(1)}]$ by $\Pr[s_1^{(1)} \in \tilde{S}^{(1)}] \cdot \Pr[s_2^{(1)} \in \tilde{S}^{(1)}]$, but this is not true. More precisely, write

$E_{sp,2}^{(2)}$: There exists a $w^{(2)}$-rep pair of $s_2^{(1)}$ in $\tilde{S}^{(1)}$, say $(s_3^{(1)}, s_4^{(1)})$.

$E_{nc,2}^{(2)}$: The $\ell^{(1)}$-near-collision pair $([Ms_3^{(2)}]_B, [Ms_4^{(2)}]_B)$ is in $\mathsf{NearCols}^{(2)}$.

Then the independence of $E_{sp,1}^{(2)}$ and $E_{sp,2}^{(2)}$ can be easily checked, considering that $\tilde{S}^{(2)}$ is defined as a random subset of $\mathcal{T}^d(w^{(2)})$. However, the event $E_{nc,1}^{(2)}$ and $E_{nc,2}^{(2)}$ are definitely dependent, so that we cannot compute our goal probability as $\Pr[s_1^{(1)} \in \tilde{S}^{(1)}] \cdot \Pr[s_2^{(1)} \in \tilde{S}^{(1)}]$. Regarding this issue, we establish the following heuristic that is sufficient to lower bound the success probability, which finally yields Eq. (C13) combined with Eq. (15). Below we provide a brief theoretic justification of this heuristic.

Heuristic 1. $\Pr[E_{nc,1}^{(2)} \wedge E_{nc,2}^{(2)}] \geq \Pr[E_{nc,1}^{(2)}] \cdot \Pr[E_{nc,2}^{(2)}] = \left(p_{nc}^{(2)}\right)^2$.

Justification. Let $e_1 = [Ms_1^{(1)}]_{B,r^{(1)}}$, $e_2 = [Ms_1^{(2)}]_{B,r^{(1)}}$, which are ensured to live in $\mathcal{C}_{\ell^{(1)}}^{r^{(1)}}$ due to the level-1 constraint. Here, e_1 and e_2 are bound by the relation $e = e_1 - e_2$ in the left-hand side, and the relation forces e_1 (and e_2) can never lie in some corner of $\mathcal{C}_{\ell^{(1)}}^{r^{(1)}}$. Meanwhile, e_1 and e_2 can lie independently in everywhere $\mathcal{C}_{\ell^{(1)}}^{r^{(1)}}$. This difference makes the "expected size" of e_1 and e_2 is smaller in the left-hand side. It finally explains the inequality, by observing that smaller e_* increases the chance of $s_*^{(1)} \in \tilde{S}^{(1)}$.

Finally, putting everything together, we have the following summarization theorem.

Theorem 9 *Given input matrices $M \in \mathbb{R}^{m \times d}$ and $B \in \mathbb{R}^{m \times m}$ such that $e := [Ms]_B$ is sampled from Gaussian distribution $\mathcal{G}_\sigma^m$ for some $s \in \mathcal{T}^d(w^{(0)})$, the success probability $p_{meet} = \Pr[s \in \tilde{S}^{(0)}]$ of Algorithm 4 with the attack specification parameters*

- *Split weights $w^{(1)}$ and $w^{(2)}$.*
- *Constraint dimension $r^{(1)}$ and norm $\ell^{(1)}$,*

- *LSH block-lengths* $b_{lsh}^{(1)}, b_{lsh}^{(2)}$ *and LSH repetition numbers* $R_{lsh}^{(1)}, R_{lsh}^{(2)}$,
- *Top-level set size* N_2,

is lower bounded by

$$p_{meet} \geq p_{ex}^{(1)} \cdot p_{adm}^{(1)} \cdot p_{ncf}^{(1)} \cdot \left(p_{sp}^{(2)} \cdot p_{adm}^{(2)} \cdot p_{ncf}^{(2)}\right)^2, \tag{18}$$

where each probability has the following meaning

- $p_{ex}^{(1)}$*: Among all* $w^{(1)}$*-rep pairs of* s*, there exists one pair* $(s_1^{(1)}, s_2^{(1)})$ *satisfying* $\|[Ms_*^{(1)}]_{B,r^{(1)}}\| \leq \ell^{(1)}$*. The computation formula is given by Lemma 7.*
- $p_{adm}^{(1)}$*: For such* $(s_1^{(1)}, s_2^{(1)})$*, the pair* $\left([Ms_1^{(1)}]_B, [Ms_2^{(1)}]_B\right)$ *becomes a near-collision pair in* $\mathcal{P}(B)$*, that is, the pair elements are coordinate-wisely close. The computation formula is given by Eq. (13).*
- $p_{ncf}^{(1)}$*: The NCF algorithm on* $L^{(1)}$ *succeeds to find the near-collision pair* $\left([Ms_1^{(1)}]_B, [Ms_2^{(1)}]_B\right)$*. The computation formula is given by Eq. (7).*
- $p_{sp}^{(2)}$*: Among all* $w^{(2)}$*-rep pair of* $s_1^{(1)}$*, there exists one pair* $(s_1^{(2)}, s_2^{(2)})$ *in* $\tilde{S}^{(2)} \times \tilde{S}^{(2)}$*. The computation formula is given by Lemma 8, which also applies for* $s_2^{(1)}$ *and its* $w^{(2)}$*-rep pair* $(s_3^{(2)}, s_4^{(2)})$*.*
- $p_{adm}^{(2)}$*: For such* $(s_1^{(2)}, s_2^{(2)})$*, the pair* $\left([Ms_1^{(2)}]_{B,r^{(1)}}, [Ms_2^{(2)}]_{B,r^{(1)}}\right)$ *becomes a near-collision pair in* $\mathcal{P}(B_{r^{(1)}})$*. The computation formula is given by Eq. (16), which also applies for* $(s_3^{(2)}, s_4^{(2)})$*.*
- $p_{ncf}^{(2)}$*: The NCF algorithm on* $L^{(2)}$ *succeeds to find the near-collision pair* $\left([Ms_1^{(2)}]_{B,r^{(1)}}, [Ms_2^{(2)}]_{B,r^{(1)}}\right)$*. The computation formula is given by Eq. (7), which also applies for* $(s_3^{(2)}, s_4^{(2)})$*.*

7 Primal-Meet-LWE

Finally, this section proposes the new hybrid attack that combines the primal lattice strategy and Meet-LWE for matrix modulus (Algorithm 4). The formal description is given in Algorithm 5. As one subtle issue, 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. Thus, we guess a weight parameter $w^{(0)}$ and solve Meet-LWE for matrix modulus while expecting $HW(s) = w^{(0)}$, and the probability of $HW(s) = w^{(0)}$ should be taken into account in the total attack success probability.

Theorem 10 *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* $e \leftarrow \mathcal{D}_{\mathbb{Z},\sqrt{2\pi}\sigma}$*. Then Algorithm 5 with* **Params** *successfully recovers the solution* s *with probability at least* $p_{suc} = p_{np} \cdot p_{hw} \cdot p_{meet}$*, where* p_{meet} *is the success probability of Algorithm 4, and*

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

Algorithm 5: PRIMAL-MEET-LWE

Input: LWE instances $(A, b) \in \mathbb{Z}_q^{m' \times (n+1)}$ such that $b = As' + e$ where $e \leftarrow \mathcal{D}_{\mathbb{Z}, \sqrt{2\pi}\sigma}$ 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
$\mathsf{meet\text{-}lwe\text{-}params} = \{(w^{(i)}, r^{(i)}, \ell^{(i)}, b_{lsh}^{(i)}, R_{lsh}^{(i)})\}_i$

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

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 4, computed as Eq. (8).

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 2.4.
- $HW(s) = w^{(0)}$, whose probability p_{hw} comes from simple combinatorics.
- $s_g \in \tilde{L}^{(0)}$, whose probability is p_{meet} by definition.

Therefore, the success probability of Algorithm 5 is $p_{suc} = p_{np} \cdot p_{hw} \cdot p_{meet}$.

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

8 Estimations and Validations

Given input LWE parameters such as m, n, q, σ, w, the concrete complexity of our attack comes from the optimal attack-specific parameterizations (such as $r^{(i)}$, $R_{lsh}^{(i)}$) of Algorithm 5 that make the following term from Theorem 10 minimal:

$$\frac{1}{p_{np} \cdot p_{hw} \cdot p_{meet}} \left(T_{bkz}(m, \beta) + T_{meet}\right). \tag{19}$$

This section provides several arguments related to the cost estimation, together with some experimental validation. We first illustrate the estimation strategy with validation of our attack, and present the estimation result in Section 8.3. All codes related to this section can be found at https://github.com/yonghaason/PrimalMeetLWE.

8.1 Computation of Concrete Values of p_{meet}

The fundamental strategy to find the optimal parameterizations for Eq. (19) is to sweep through all possible parameter choices and pick the one that minimizes the total cost. Unsurprisingly, this exhaustive approach is extremely inefficient, and the dominant bottleneck for each computation of Eq. (19) is p_{meet}: We use the lower bound of p_{meet} established in Theorem 9, namely

$$p_{meet} \geq p_{ex}^{(1)} \cdot p_{adm}^{(1)} \cdot p_{ncf}^{(1)} \cdot \left(p_{sp}^{(2)} \cdot p_{adm}^{(2)} \cdot p_{ncf}^{(2)}\right)^2.$$

Here the computation of p_{ex} (Eq. (12)) and p_{ncf} (Eq. (7)) involves the following integral computations which cannot be represented in a closed form:

$$\mathop{\mathbb{E}}_{x \leftarrow \mathcal{G}_\sigma}\left[1-(1-p(x))^R\right] = \int \rho_\sigma(x) \cdot \left(1-(1-p(x))^R\right) dx.$$

For a rapid search over the parameter space, we restrict the reasonable search ranges for some parameters that yields more efficient optimization. Further details can be found in Section D due to the space limit.

8.2 Experimental Validation of p_{meet}

Most of the terms in the total cost formula Eq. (19) are easily computed. The only exception is p_{meet}, for which we provide a lower bound in Theorem 9 with extremely complicated argument. To ensure the reliability of this component, we conducted the following two experimental validations.

Full implementation of Algorithm 4. In this experiments, we directly examine whether the lower bound holds. More precisely, we randomly generates matrices $M \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{m \times d}$ such that $[Ms]_B = e$ for some $s \in \mathcal{T}^d(w^{(0)})$ and $e \leftarrow \mathcal{G}_\sigma^m$, and compute the empirical probability (denoted 'Real') that Algorithm 4 finds the solution $s \in \mathcal{T}^d(w^{(0)})$.

Table 1 shows the results for various and secret dimension d and searching spaces (namely $\binom{d}{w^{(0)}} \cdot 2^{w^{(0)}}$, denoted by $|S|$), where the modulo matrix B is modeled via GSA so that $\|b_i^*\| = q_l \cdot \delta_0^{2\cdot(n-i)}$. The attack-specific parameters (such as LSH parameters) are chosen following the above rapid search; more precisely, the guideline of Section D with $C_{proj} = C_{lsh} = 2$.

As a side note, the experiments also indicate that the theoretical lower bound, as well as our estimation, could be quite conservative. We leave a tighter analysis of p_{meet} and its implication to the estimation as open.

Validation of Heuristic 1. One could be doubtful on the behavior of lower bounds when extrapolated to real-world scale large parameters given small input parameters in Table 1. We provide another experiment that exclusively focuses on Heuristic 1, which is the only heuristic that Theorem 9 assumes:

$$\Pr[E_{nc,1}^{(2)} \wedge E_{nc,2}^{(2)}] \geq \Pr[E_{nc,1}^{(2)}] \cdot \Pr[E_{nc,2}^{(2)}] = \left(p_{nc}^{(2)}\right)^2$$

Considering Proposition 6 again, the probability of NCF with repetition number R_{lsh} finding a near-collision pair of distance e is $f(e) := 1-(1-p_{lsh}(e))^{R_{lsh}}$. Therefore, our argument boils down to the comparison of the expected value of $f(e_1^{(1)}) \cdot f(e_2^{(1)})$ for the following cases:

Exp_1: $e_1^{(1)}, e_2^{(1)} \in [-\ell^{(1)}, \ell^{(1)}]^{r^{(1)}}$ satisfying $e^{(0)} = e_1^{(1)} - e_2^{(1)}$ where $e^{(0)} \leftarrow \mathcal{G}_\sigma$;

Exp_2: $e_1^{(1)}, e_2^{(1)}$ are independently sampled in $[-\ell^{(1)}, \ell^{(1)}]^{r^{(1)}}$.

Table 1: 'Real' column shows the success counts averaged from 100 experiments, and 'Theory' column shows the RHS of Eq. (18). For every case, the standard deviation for $[Ms]_B = e$ is fixed by $\sigma = 1$. All theoretical lower bounds are much smaller than experimental probability, making our estimation conservative.

Input Params.					Attack-specific Params.					p_{meet}	
m	d	$w^{(0)}$	q_l	$\lvert S\rvert$	$w^{(1)}$	$w^{(2)}$	$r^{(1)}$	$R_{lsh}^{(1)}$	$R_{lsh}^{(2)}$	Real	Theory
20	20	10	25	$\approx 2^{27}$	6	3	6	5	9	0.90	0.113
30	30	10	12.5	$\approx 2^{34}$	6	3	12	7	7	0.13	0.007
40	40	10	25	$\approx 2^{40}$	6	3	7	19	11	0.75	0.094
40	40	14	30	$\approx 2^{48}$	8	4	8	17	15	0.60	0.112
40	80	10	40	$\approx 2^{51}$	8	4	10	16	26	0.67	0.140
30	120	8	40	$\approx 2^{57}$	6	3	9	8	19	0.61	0.108

As this experiment does not depend on the size of searching space, it is easy to scale to larger parameters. Figure 7 shows the results while varying the projection dimension $r^{(1)}$, where the other parameters are also chosen by the guideline of Section D with $C_{proj} = 10$. It shows not only Heuristic 1 consistently holds, but also the gap between our estimation and real behavior becomes larger. This trend would be convincing evidence that our computation of p_{meet} remains robust for the large-parameter regime in Section 8.3.

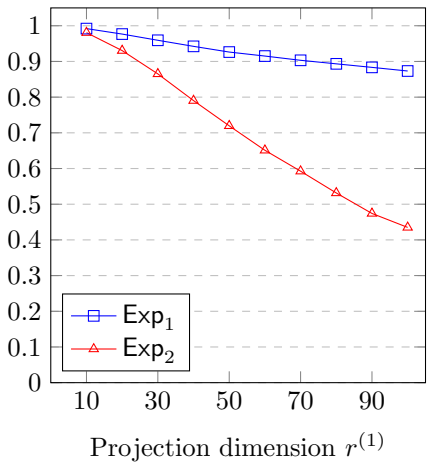

Fig. 7: Experimental validation of Heuristic 1: Exp_1 shows $\Pr[E_{nc,1}^{(1)} \wedge E_{nc,2}^{(1)}]$ and Exp_2 shows $\Pr[E_{nc,1}^{(1)}] \cdot \Pr[E_{nc,2}^{(1)}]$

8.3 Concrete Estimations

We mainly consider the fully homomorphic encryption (FHE) literature for concrete estimations that actively uses (extremely) sparse secret key of hamming weight from 64 to 256, which is essential for efficient bootstrapping [6, 8, 33, 34].

Remark 5 Another interesting question would be the impact on PQC. First, our attack provides no improvement for NIST PQC finalists such as Kyber [35] or Dilithium [36], since they are not based on sparse secrets. Other PQC schemes use sparse (or fixed hamming-weight) ternary secret keys for efficiency, especially NTRU [4, 37]. However, NTRU parameters have a critical difference from FHE parameters in that the error distribution is a *uniform ternary distribution*, not a (discrete) Gaussian. Some previous analyses, including lattice-estimator [28], estimate the attack cost by approximating the error distribution into the (discrete) Gaussian with the same variance. However, we recall that [20] exactly pointed out that such an approximation from ternary distribution to Gaussian could be inadequate. For this reason, we decide not to provide any estimation for NTRU parameters, to prevent any potential misuse.

Cost metric and assumptions. We mainly follow the similar assumptions and cost metric used in lattice-estimator [28] for the fair comparisons between the different attacks.

- We identify a real number and an element in $\mathbb{Z}_q$ as a unit *word*, and assume that any arithmetic operation between units by one unit cost.
- 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 $5.38m \cdot 2^{0.296\beta+20.388}$ [38] unit costs.
- We use Geometric Series Assumption (GSA) to simulate the Gram-Schmidt norm of BKZ output basis.

Remark 6 (More accurate cost metric) The cost metrics for the concrete security estimation reach the point of considering the gate-level cost for each real number operation, especially in the NIST PQC standardization. We acknowledge that these kinds of details should eventually be examined for our attack as well and left as future work.

Estimation results. Table 2 showed the estimation results based on the parameter search range of Figure D1. The numbers for other attacks except for Meet-LWE[6] are from lattice-estimator [28] (at commit `5ba00f5`), and the numbers for ours are from our estimation scripts. For the consistency check of two scripts, we check that both scripts output almost identical estimations for Howgrave-Graham's hybrid attack [17] (corresponds to $t = 1$ of our attack) within one bit difference. The consequence is quite expectable. Our attack beats the previous best attacks, thanks to the severe sparsity of the secret vectors.

Non-hybrid attacks. One might be curious how hybrid attacks can be worse than 'Non-hybs', 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').

[6] We use a very optimistic estimation for Meet-LWE: $\mathcal{S}^{0.25}$ time for the key space size $\mathcal{S}$. A non-asymptotic complexity of Meet-LWE is around $\mathcal{S}^{0.3}$ or larger.

Table 2: Attack cost estimations for some deployed FHE parameters [6–8, 34, 40], where the standard deviation of error is $\sigma = 3.2$ for all cases.

(n, q, w)	Primal Strategy				Dual Strategy[1] [27, 39]	Meet-LWE [11]
	Non-hybs [15, 16]	HG [17] $t=1$	**Ours** $t=2$	$t=3$		
$(2^{16}, 2^{1450}, 64)$	155	114.6	**105.3**	108.5	110.2	198
$(2^{15}, 2^{699}, 128)$	161	135.9	**128.9**	129.6	133.5	332.9
$(2^{15}, 2^{768}, 192)$	145	134.2	**129.3**	129.3	140.5	471.7
$(2^{16}, 2^{1553}, 192)$	145	134.0	**129.8**	141.8	134.3	519.8
$(2^{14}, 2^{426}, 256)$	128.7	125.3	**120.9**	121.2	129.2	538.3
$(2^{16}, 2^{1568}, 256)$	143.2	137.3	**132.9**	138.6	144.0	666.9

[1]We remark that the numbers for Dual Strategy are from hybrid dual attacks, but they are in doubt due to a concern [41] about the core assumption used for the dual strategy. Although there is a following report [42] that partially resolves the concern, it is only applicable to rather simple versions of dual attack, not *dual hybrid*.

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. This is in contrast to the original Meet-LWE [11], 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 [11], particularly due to the near-collision finding algorithms.

Appendix A Flaw in the analysis of [24]

As noted in the introduction, Kirshanova and May suggested the improved Meet-LWE algorithm based on the locality-sensitive hashing, called LSH-Meet-LWE [24]. 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

$$R \approx q^{r/2} \cdot (q/3)^{r/2} \tag{A1}$$

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. (A1) 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 $s_2^{(1)} \in L_2^{(1)}$ [11, 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.

Appendix B Full Version of Proposition 6

Proposition 11 *Let $\mathcal{Q} = \prod_{i=1}^{r}[-q_i/2, q_i/2)$ be an r-dimensional domain, $L \subset \mathcal{Q}$ 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 4.*

(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 either Model 2 or Model 1, for any $0 < \epsilon < 1$, Algorithm 3 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

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

for any r-dimensional distribution χ, and $p_{lsh}(\mathcal{Q}, e, b)$ defined by Lemma 5. Then

- *Assume that L follows Model 1 (one special $\mathcal{G}_\sigma$-near-collision model). The output C of Algorithm 3 contains the special $\mathcal{G}_\sigma$-near-collision with probability at least $p_{ncf}(\mathcal{Q}, \mathcal{G}_\sigma, \ell, b)$.*
- *Assume that L follows Model 2 (purely random model). For a random ℓ-near-collision pair, the output C of Algorithm 3 contains it with probability at least $p_{ncf}(\mathcal{Q}, \mathcal{U}([-\ell, \ell]^r), \ell, b)$.*

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

Proof We divide the proof into two parts.

The complexity part. Under Model 2, 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

$$p_{col} = \prod_{1 \le i \le r} \frac{1}{n_i}. \tag{B2}$$

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

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

Applying Chebyshev's inequality, we have

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

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

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

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 1, 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 2 applies to the other pair. In other words, the claim almost similarly holds when assuming Model 1.

The correctness part. We first show the correctness of the algorithm under Model 2, which can be rephrased as follows.

Lemma 12 *Assume the input list L follows Model 2. Then Algorithm 3 finds a random ℓ-near-collision pair with a probability at least*

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

of Lemma 12 Let (y_1, y_2) be a random ℓ-near-collision pair. The probability of one iteration succeeding to find this pair is $p_{lsh}(e, b)$ of Lemma 5, 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 $1 - (1 - p_{lsh}(e, b))^R$.

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 $\mathcal{Q}$, and then the success probability is represented by $\mathbb{E}_{e\leftarrow\mathcal{E}}\left[1-(1-p_{lsh}(e,b))^R\right]$. 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

$$\mathop{\mathbb{E}}_{e\leftarrow\mathcal{E}}\left[1-(1-p_{lsh}(e,b))^R\right] \ge \mathop{\mathbb{E}}_{e\leftarrow[-\ell,\ell]^r}\left[1-(1-p_{lsh}(e,b))^R\right]. \tag{B3}$$

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

$$\mathbb{E}_{e\leftarrow[-\ell,\ell]}[f\cdot g] \ge \mathbb{E}_{e\leftarrow[-\ell,\ell]}[f]\cdot\mathbb{E}_{e\leftarrow[-\ell,\ell]}[g] \tag{B4}$$

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. (B3) can be written as

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

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. (B4) for each e_i for $i \in [r]$ to obtain the desired result. □

The correctness for Model 1 is almost analog of Lemma 12, 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 13 *Assume the input list L follows Model 1 where the special near-collision pair is sampled from $\mathcal{D}$. Then Algorithm 3 finds the special near-collision pair with a probability at least*

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

Lemma 12 and Lemma 13 completes the correctness part of Proposition 6. □

Appendix C Full Analysis of Meet-LWE

This section presents the analysis of Algorithm 4, which is probably the most involved part of this paper. The resulting statement is presented in the end of this section as Theorem 18, and this entire section is a proof for that theorem.

C.1 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 $i \in [t-1]$.

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

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

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

Then we have the following inequality

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

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

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

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

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

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.

C.2 Computation of Splitting Probability.

The goal of this part is Proposition 14 and Lemma 15 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

$$([Ms_{2k-1}^{(i+1)}]_{B,r^{(i+1)}}, [Ms_{2k}^{(i+1)}]_{B,r^{(i+1)}}) \in \mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}} \times \mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}} \tag{C5}$$

See again Figure 4 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,

$$s_k^{(i-2)} = (s_{4k-3}^{(i)} - s_{4k-2}^{(i)}) - (s_{4k-1}^{(i)} - s_{4k}^{(i)}), \tag{C6}$$

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

Proof 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 4, 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. (C6) on $e_k^{(i)}$.

In this regard, observe that Eq. (C6) 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. (C6) 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 $d_k \in \mathcal{C}^{r^{(i+1)}}_{\ell^{(i-1)}}$ and $e_{2k-1}, e_{2k} \in \mathcal{C}^{r^{(i)}}_{\ell^{(i+1)}}$ 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 $E^{(i)}_{sp,2k-1}$ and $E^{(i)}_{sp,2k}$ 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 $e_{2k-1}, e_{2k} \in \mathcal{C}^{r^{(i)}}_{\ell^{(i+1)}}$, the relation restricts e_{2k-1} and e_{2k} so that they never lie in some corner of $\mathcal{C}^{r^{(i)}}_{\ell^{(i+1)}}$. 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 $E^{(i)}_{sp,*}$ increases as the size of e_* decreases, we may expect that the chance of $E^{(i)}_{sp,2k-1} \wedge E^{(i)}_{sp,2k}$ is larger in the left-hand side on average. □

We now proceed to compute each probability $\Pr[E^{(i)}_{sp,k}]$ 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 $E^{(i)}_{sp,k}$ means that at least one pair in $P(s_k^{(i)})$ lies in $\mathcal{C}^{r^{(i+1)}}_{\ell^{(i+1)}}$. To analyze the probability, we need to specify the distribution of $P(s_k^{(i)})$. Precisely, we model the set

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

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 $P(s_k^{(i)})$.

The underlying intuition of this model is to consider each element of $[Ms_r]_{B,r^{(i+1)}}$ as a uniform sample x from $\mathcal{P}(B^*_{r^{(i+1)}})$ independent from each other. 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 $e = [Ms_k^{(i)}]_{B,r^{(i+1)}}$, 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 $(-s'_r, -s_r)$.

Now we can compute $\Pr[E^{(i)}_{sp,k}]$ and then $\Pr[E^{(i)}_{sp}]$ as follows.

Proposition 14 *Assume Heuristic 2 holds for $1 \le i < t-1$, and let*

$$\chi^{(i)}_{sp} = \begin{cases} \mathcal{G}^{r^{(1)}}_{\sigma} & \text{if } i = 0 \\ \mathcal{U}^{r^{(i+1)}}_{\ell^{(i)}} & \text{otherwise,} \end{cases} \quad \text{and } p_{rep}(e) = \prod_{j \in [r^{(i+1)}]} \frac{2\ell^{(i+1)}}{\|b^*_{m-r^{(i+1)}+j}\|} \cdot \left(1 - \frac{|e_j|}{2\ell^{(i+1)}}\right).$$

Then for $0 \le i < t-1$,

$$\Pr[E^{(i)}_{sp,k}] = p_{sp}(\chi^{(i)}_{sp}) := \mathop{\mathbb{E}}_{e \leftarrow \chi^{(i)}_{sp}} \left[1 - (1 - p_{rep}(e))^{R^{(i)}/2}\right]. \tag{C7}$$

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

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

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

$$p_{rep}(e) = \Pr\left[x \in \mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}}, [x+e]_{B,r^{(i+1)}} \in \mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}} \mid x \leftarrow \mathcal{P}(B^*_{r^{(i+1)}})\right]. \tag{C8}$$

Since each first element is independent and by the symmetry of $P(s_k^{(i)})$, the number of pairs in $\mathcal{C}_{\ell^{(i+1)}}^{r^{(i+1)}}$ follows $2 \cdot B(R^{(i)}/2, p_{rep}(e))$[8] where $e = [Ms_k^{(i)}]_{B,r^{(i+1)}}$, and $B(\cdot,\cdot)$ is a binomial distribution.

We proceed to explain the distribution $\chi_{sp}^{(i)}$ where e is sampled. For $i = 0$, our problem setting ensures that $x = [Ms_1^{(0)}]_B$ follows Gaussian distribution $\mathcal{G}_\sigma$ so that $\chi_{sp}^{(0)} = \mathcal{G}_\sigma^{r^{(1)}}$.

For $i \geq 1$, we use the fact that each level-i target element $s_k^{(i)} \in S^{(i)}$ defining $e = [Ms_k^{(i)}]_{B,r^{(i+1)}}$ is one element of a rep pair of some $s_*^{(i-1)} \in S^{(i-1)}$. When $s_k^{(i)}$ is the first element of the rep pair of $s_*^{(i-1)}$, our modelling for $s_*^{(i-1)}$ immediately implies the uniformity of e. If $s_k^{(i)}$ is the second element, $[Ms_k^{(i)}]_{B,r^{(i+1)}}$ 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 $e_k^{(i)}$ is included in $\mathcal{C}_{\ell^{(i+1)}}^{r^{(i)}}$, so we have $e_k^{(i)} \leftarrow \mathcal{U}_{\ell^{(i+1)}}^{r^{(i)}}$.

Finally, it holds that

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

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

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

The first probability is immediately computed as the volume ratio

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

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 $e = (e_1, \ldots, e_{r^{(i+1)}})$. 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 $\prod_{j \in [r^{(i+1)}]} \left(1 - \frac{|e_j|}{2\ell^{(i+1)}}\right)$. □

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

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

Lemma 15 $\Pr[E_{sp}^{(t-1)}] \geq \left(1 - e^{-R^{(t-1)} \cdot |S^{(t)}|^2 / |\mathcal{T}^d(w^{(t)})|^2}\right)^{2^{t-1}}$. *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 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

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

□

C.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.

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

Proof This heuristic can be justified almost similarly to Heuristic 2, 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 $(s_{2k-1}^{(i+1)}, s_{2k}^{(i+1)})$. □

To compute $\Pr[E_{ncf,k}^{(i)}]$ and $\Pr[E_{ncf}^{(i)}]$, we model the list $L^{(i+1)} = \{[Mx]_{B,r^{(i)}} \mid x \in \tilde{S}^{(i+1)}\} \subset \mathcal{Q}^{(i+1)}$ as follows:

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

whose rationale is to consider each element 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.

Proposition 16 *Assume Heuristic 3. Let*

$$\chi_{ncf}^{(i)} = \begin{cases} \mathcal{G}_\sigma^{r^{(0)}} & \text{if } i = 0 \\ \mathcal{U}_{\ell^{(i)}}^{r^{(i)}} & \text{otherwise,} \end{cases} \quad \text{and } \mathcal{B}^{(i)} := \prod_{j=r^{(i)}+1}^{r^{(i+1)}} \left[-\frac{\|b_{m-j}^*\|}{2}, \frac{\|b_{m-j}^*\|}{2} \right]$$

for $0 \le i \le t-1$. Let $p_{ncf}(\ell, b_{lsh}^{(i+1)}; \chi)$ be computed as in Proposition 6. Then the following holds for $0 \le i \le t-1$:

$$\Pr[E_{ncf}^{(i)}] \ge \left(p_{adm}(\mathcal{B}^{(i)}, \chi_{ncf}^{(i)}) \cdot p_{ncf}(\ell^{(i)}, b_{lsh}^{(i+1)}; \chi_{ncf}^{(i)}) \right)^{2^i}.$$

Proof 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

$$e_{2k} = e_{2k-1} + d_k, \text{ or } e_{2k,j} = e_{2k-1,j} + d_{k,j} \text{ for every } j \in [r^{(i)}] \tag{C9}$$

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. This implies that Eq. (C9) holds for $j \in [r^{(i)} - r^{(i+1)}, r^{(i)}]$.

In the remaining index $j \in [r^{(i)} - r^{(i+1)}]$, considering Lemma 4, Eq. (C9) translates into

$$|e_{2k-1,j} + d_{k,j}| \le \|b_{m-r^{(i)}+j}^*\|/2. \tag{C10}$$

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. (C10) for $j \in [r^{(i)} - r^{(i+1)}]$ is $1 - \frac{|d_{k,j}|}{\|b_{m-r^{(i)}+j}^*\|}$.

Since $d_k = [Ms_k^{(i)}]_{B,r^{(i)}} \leftarrow \chi_{ncf}^{(i)}$, the probability of Eq. (C9) is

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

meaning that (e_{2k-1}, e_{2k}) becomes $\ell^{(i)}$-near-collision in $\mathcal{Q}^{(i)}$ with probability $p_{nc}(\chi_{ncf}^{(i)})$. The probability that Algorithm 3 successfully finds the $\ell^{(i)}$-near-collision pair (e_{2k-1}, e_{2k}) is $p_{ncf}(\ell^{(i)}, b_{lsh}^{(i+1)}; \chi_{ncf}^{(i)})$ as analyzed in Proposition 6. 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 $\mathcal{U}_{\ell^{(i)}}^{r^{(i)}}$. □

C.4 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 4.

Lemma 17 *For any $0 < \epsilon < 1$, the time cost $T^{(i)}_{loop}$ of the i-th level of the main loop of Algorithm 4 is dominated by*

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

with probability at least $1 - \epsilon$, where $N^{(i)}_{col}(|\tilde{S}^{(i)}|, \epsilon)$ is defined as Proposition 11.

Proof 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 3 over $L^{(i)}$
- Line 6-11: Checking hamming weight

It can be easily derived that the first item (Line 3) takes $O\left(|\tilde{S}^{(i)}| \cdot (r^{(i-1)}w^{(i)} + (r^{(i-1)})^2)\right)$ operations. However, since $R^{(i)}_{lsh} \gg 1$, this cost is much smaller than others.

The cost for the second item (Line 4) directly comes from Proposition 6 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 3, and checking Hamming weight is done with $2w^{(i)}$ operations by storing the nonzero coordinates of each $x \in S^{(i)}$. □

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:

$$|S^{(i)}| = |\mathcal{T}^d(w^{(i)})| \cdot \frac{\mathrm{vol}(\mathcal{C}^{r^{(i)}}_{\ell^{(i)}})}{\mathrm{vol}(\mathcal{P}(B^*_{r^{(i)}}))} = \binom{d}{w^{(i)}} 2^{w^{(i)}} \cdot \prod_{j \in [r^{(i)}]} \frac{2\ell^{(i)}}{\|b^*_{m-r^{(i)}+j}\|}. \tag{C11}$$

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

C.5 Full Theorem Statement.

Putting everything together, we have the full theorem that analyzes Algorithm 4. Note that again the contents in this section so far is a proof for this theorem.

Theorem 18 *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*

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

be the set of parameters satisfying the conditions in Algorithm 4. Assuming that Heuristics 2 and 3, the followings hold.

(Complexity) For any $0 < \epsilon < 1$, the running time of Algorithm 4 is dominated[9]

$$\sum_{i=0}^{t} R_{lsh}^{(i)} \cdot \left(|\tilde{S}^{(i)}| \cdot T_{\mathsf{hash}}(r^{(i-1)}) + N_{col}^{(i)}(|\tilde{S}^{(i)}|, \epsilon) \cdot (T_{\mathsf{check}}(r^{(i-1)}) + 2w^{(i)}) \right) \tag{C12}$$

with probability at least $1 - t \cdot \epsilon$, where $N_{col}^{(i)}(|\tilde{S}^{(i)}|, \epsilon)$ is defined as Proposition 6 and

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

(Correctness) Let $p_{suc}^{meet} := \Pr[s \in \tilde{S}^{(0)}]$ be the probability that Algorithm 4 finds the solution s. It holds that

$$p_{suc}^{meet} \geq \prod_{i=0}^{t-1} \Pr[E_{sp}^{(i)}] \cdot \prod_{i=0}^{t-1} \Pr[E_{ncf}^{(i)}] \tag{C13}$$

where the explicit formulas for $\Pr[E_{sp}^{(i)}]$ and $\Pr[E_{ncf}^{(i)}]$ are given in Proposition 14 and Proposition 16, respectively.

Appendix D Restriction of parameter search range

This section details the specific parameter range that we used to estimate the attack costs of Algorithm 4. Note that these choices are not seriously optimized, and one could find other range of parameters that lead to a better attack cost estimation. We also assume here the top level is $t = 2$.

Constraint parameters $(r^{(1)}, \ell^{(1)})$. First of all, we fix the norm bound of $\ell^{(0)}$ by 6σ, and also the constraint length parameter $\ell^{(1)}$ by 6σ. Then, for the optimal choice of $r^{(1)}$, it is correct to investigate for every possible dimension parameter $r^{(1)}$ and compute the probability

$$p_{ex}^{(1)} := \mathbb{E}_{x \leftarrow \mathcal{G}_\sigma} \left[1 - (1 - p_{rep}(x))^{R^{(i)}/2} \right]$$

from Lemma 7. However, the computation involves extremely complicated integral formula

$$\mathop{\mathbb{E}}_{x \leftarrow \mathcal{G}_\sigma} \left[1 - (1 - p_{rep}(x))^{R^{(1)}} \right] = \int \rho_\sigma(x) \cdot \left(1 - (1 - p_{rep}(x))^{R^{(1)}} \right) dx,$$

whose computation is horribly slow. Instead, we simply fix $r^{(1)}$ by the maximal one such that

$$R^{(i)} \geq \frac{C_{proj}}{\mathbb{E}_{x \leftarrow \mathcal{G}_\sigma}[p_{rep}(x)]} \tag{D14}$$

for some constant $C_{proj} > 0$, whose underlying rationale is a naive approximation $p_{ex}^{(1)} = \mathbb{E}[1 - (1 - p(x))^R] \approx 1 - (1 - \mathbb{E}[p(x)])^R$; Eq. (D14) is the choice that makes the RHS $1 - \exp(-C_{proj}) \approx 1$.

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

Note that the denominator can be computed by admissibility probability, by

$$\begin{aligned}\mathop{\mathbb{E}}_{x\leftarrow\mathcal{G}_\sigma}[p_{rep}(x)] &= \mathop{\mathbb{E}}_{x\leftarrow\mathcal{G}_\sigma}\left(\frac{(2\ell^{(1)})^{r^{(1)}}}{\mathrm{vol}\left(\mathcal{P}(B^*_{r^{(1)}})\right)}\cdot\prod_{i\in[r^{(1)}]}\left(1-\frac{|x_i|}{2\ell^{(1)}}\right)\right)\\ &= \frac{(2\ell^{(1)})^{r^{(1)}}}{\mathrm{vol}\left(\mathcal{P}(B^*_{r^{(1)}})\right)}\cdot p_{adm}\left(\mathcal{C}^{r^{(1)}}_{\ell^{(1)}},\mathcal{G}_\sigma\right).\end{aligned}$$

The numbers in Section 8.3 come from $C_{proj} = 10$, which is an ad-hoc choice from some internal experiments.

Choice of LSH parameters. First, we fix LSH block lengths by $b^{(1)}_{lsh} = b^{(2)}_{lsh} = 2\ell^{(1)} = 12\sigma$. Then, we need to optimize the choice of LSH repetition number $R^{(1)}_{lsh}$ and $R^{(2)}_{lsh}$, while computing the success probability of near-collision finding by Equation (7):

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

Note that we take a similar approach to Equation (D14) to determine the number of LSH repetitions: we simply fix $R^{(1)}_{lsh}$ by the minimal one such that

$$R^{(1)}_{lsh} \ge \frac{C_{lsh}}{\mathbb{E}_{e\leftarrow\mathcal{G}_\sigma}[p_{lsh}(\mathcal{P}(B^*),e,b^{(1)}_{lsh})]} \tag{D15}$$

for some $C_{lsh} > 0$, where the denominator can be computed again as admissibility probability

$$\mathop{\mathbb{E}}_{e\leftarrow\mathcal{G}_\sigma}[p_{lsh}(\mathcal{P}(B^*),e,b^{(1)}_{lsh})] = \mathop{\mathbb{E}}_{e\leftarrow\mathcal{G}_\sigma}\left(\prod_{\substack{1\le i\le m\\ n_i>1}}\left(1-\frac{|e_i|}{b^{(1)}_{lsh}}\right)\right) = p_{adm}(\mathcal{C}^m_{b^{(1)}_{lsh}},\mathcal{G}_\sigma).$$

Similarly, $R^{(2)}_{lsh}$ is chosen by the minimal one such that

$$R^{(2)}_{lsh} \ge \frac{C^{(2)}_{lsh}}{\mathbb{E}_{e\leftarrow\mathcal{C}^{\ell^{(1)}}_{r^{(1)}}}[p_{lsh}(\mathcal{P}_{r^{(1)}}(B^*),e,b^{(2)}_{lsh})]} = \frac{C^{(2)}_{lsh}}{p_{adm}(\mathcal{C}^{r^{(1)}}_{b^{(2)}_{lsh}},\mathcal{U}(\mathcal{C}^{r^{(1)}}_{\ell^{(1)}}))}$$

Again, we use $C_{lsh} = 10$ for the numbers in Section 8.3.

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

Top-level set size N_t. The top-level set size N_t is fixed by $N_t = \frac{\sqrt{3}|\mathcal{T}^d(w^{(t)})|}{\sqrt{R^{(t)}}}$, which makes the probability $p^{(t)}_{sp} = 1-\exp(-R^{(t)}\cdot(N_t/|\mathcal{T}^d(w^{(t)})|)^2) = 1-\exp(-3)\approx 0.95$.

References

[1] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM) **56**(6), 1–40 (2009)

[2] Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: Annual Cryptology Conference, pp. 40–56 (2013). Springer

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 $0 \le i \le t-1$
- $\chi^{(0)} = \mathcal{G}_\sigma$ and $\chi^{(i)} = \mathcal{U}_{\ell^{(i)}}$ for $1 \le i \le t$.
- $r^{(i)}$: 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. (D14))
- $b_{lsh}^{(i)} = 12\sigma$ for $1 \le i \le t$
- $R_{ncf}^{(i)} = \left\lceil C_{lsh}/\mathbb{E}_{\chi^{(i-1)}}[p_{lsh}(e, b_{lsh}^{(i)})] \right\rceil$ with $C_{lsh} = 10$ for $1 \le i \le t$ (See Eq. (D15))

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

Fig. D1: Parameter search range

[3] Güneysu, T., Lyubashevsky, V., Pöppelmann, T.: Practical lattice-based cryptography: A signature scheme for embedded systems. In: International Workshop on Cryptographic Hardware and Embedded Systems, pp. 530–547 (2012). Springer

[4] Chen, C., Danba, O., Hoffstein, J., Hülsing, A., Rijneveld, J., Schanck, J.M., Saito, T., Schwabe, P., Whyte, W., Xagawa, K., Yamakawa, T., Zhang, Z.: PQC round-3 candidate: NTRU. Technical report (2020). Accessed: 2022-05-20

[5] Baan, H., Bhattacharya, S., Fluhrer, S., Garcia-Morchon, O., Laarhoven, T., Rietman, R., Saarinen, M.-J.O., Tolhuizen, L., Zhang, Z.: Round5: Compact and fast post-quantum public-key encryption. In: International Conference on Post-Quantum Cryptography, pp. 83–102 (2019). Springer

[6] Bossuat, J.-P., Mouchet, C., Troncoso-Pastoriza, J., Hubaux, J.-P.: Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 587–617 (2021). Springer

[7] Lee, J.-W., Lee, E., Lee, Y., Kim, Y.-S., No, J.-S.: 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, pp. 618–647 (2021). Springer

[8] Lee, Y., Lee, J.-W., Kim, Y.-S., Kim, Y., No, J.-S., Kang, H.: 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, pp. 551–580 (2022). Springer

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

[10] Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: International Algorithmic Number Theory Symposium, pp. 267–288 (1998). Springer

[11] May, A.: How to meet ternary LWE keys. In: Annual International Cryptology Conference, pp. 701–731 (2021). Springer

[12] Howgrave-Graham, N., Joux, A.: New generic algorithms for hard knapsacks. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 235–256 (2010). Springer

[13] Becker, A., Coron, J.-S., Joux, A.: Improved generic algorithms for hard knapsacks. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 364–385 (2011). Springer

[14] Bonnetain, X., Bricout, R., Schrottenloher, A., Shen, Y.: Improved classical and quantum algorithms for subset-sum. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 633–666 (2020). Springer

[15] Albrecht, M.R., Göpfert, F., Virdia, F., Wunderer, T.: Revisiting the expected cost of solving uSVP and applications to LWE. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 297–322 (2017). Springer

[16] Liu, M., Nguyen, P.Q.: Solving bdd by enumeration: An update. In: Cryptographers' Track at the RSA Conference, pp. 293–309 (2013). Springer

[17] Howgrave-Graham, N.: A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In: Annual International Cryptology Conference, pp. 150–169 (2007). Springer

[18] Son, Y., Cheon, J.H.: 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, pp. 11–20 (2019)

[19] Wunderer, T.: A detailed analysis of the hybrid lattice-reduction and meet-in-the-middle attack. Journal of Mathematical Cryptology **13**(1), 1–26 (2019)

[20] Nguyen, P.Q.: Boosting the hybrid attack on ntru: Torus LSH, Permuted HNF

and Boxed Sphere. NIST Third PQC Standardization Conference (2021)

[21] Wang, Y., Ikematsu, Y., Akiyama, K., Takagi, T.: Cryptanalysis of giophantustm schemes against hybrid attack. In: Proceedings of the 7th ACM Workshop on ASIA Public-Key Cryptography, pp. 28–35 (2020)

[22] HEAAN. https://github.com/snucrypto/HEAAN. Accessed: 2023-10-06

[23] Lattigo. https://github.com/tuneinsight/lattigo. Accessed: 2023-10-06

[24] Kirshanova, E., May, A.: How to find ternary LWE keys using locality sensitive hashing. In: IMA International Conference on Cryptography and Coding, pp. 247–264 (2021). Springer

[25] Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 1–20 (2011). Springer

[26] Bai, S., Galbraith, S.D.: Lattice decoding attacks on binary LWE. In: Australasian Conference on Information Security and Privacy, pp. 322–337 (2014). Springer

[27] Albrecht, M.R.: 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, pp. 103–129 (2017). Springer

[28] Lattice-Estimator. https://github.com/malb/lattice-estimator. Accessed: 2023-10-06

[29] Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. In: Cryptographers' Track at the RSA Conference, pp. 319–339 (2011). Springer

[30] Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, pp. 604–613 (1998)

[31] Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: Proceedings of the Twentieth Annual Symposium on Computational Geometry, pp. 253–262 (2004)

[32] Har-Peled, S., Indyk, P., Motwani, R.: Approximate nearest neighbor: Towards removing the curse of dimensionality. Theory of computing **8**(1), 321–350 (2012)

[33] Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: Bootstrapping for approximate homomorphic encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 360–384 (2018). Springer

[34] Han, K., Ki, D.: Better bootstrapping for approximate homomorphic encryption. In: Cryptographers' Track at the RSA Conference, pp. 364–390 (2020). Springer

[35] Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-kyber: a cca-secure module-lattice-based kem. In: EuroS&P, pp. 353–367 (2018). IEEE

[36] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems **2018**(1), 238–268 (2018) https://doi.org/10.13154/tches.v2018.i1.238-268

[37] IEEE Standard Specification for Public Key Cryptographic Techniques Based on Hard Problems over Lattices. IEEE Std 1363.1-2008. Accessed: 2022-05-27 (2008)

[38] MATZOV: Report on the Security of LWE: Improved Dual Lattice Attack. https://zenodo.org/record/6493704. Accessed: 2023-10-06 (2022)

[39] Cheon, J.H., Hhan, M., Hong, S., Son, Y.: A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret LWE. IEEE Access **7**, 89497–89506 (2019)

[40] Bae, Y., Cheon, J.H., Kim, J., Stehlé, D.: Bootstrapping bits with ckks. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 94–123 (2024). Springer

[41] Ducas, L., Pulles, L.N.: Does the dual-sieve attack on learning with errors even work? In: Annual International Cryptology Conference, pp. 37–69 (2023). Springer

[42] Pouly, A., Shen, Y.: Provable dual attacks on learning with errors. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 256–285 (2024). Springer