Exploring Trade-offs in Batch Bounded Distance Decoding

Martin R. Albrecht[1], Benjamin R. Curtis[1], and Thomas Wunderer[2] *

[1] Information Security Group, Royal Holloway, University of London, UK
[2] Bundesamt für Sicherheit in der Informationstechnik (BSI), Germany
martin.albrecht@royalholloway.ac.uk,
benjamin.curtis.2015@rhul.ac.uk,
thomas.wunderer@bsi.bund.de

Abstract. Algorithms for solving the Bounded Distance Decoding problem (BDD) are used for estimating the security of lattice-based cryptographic primitives, since these algorithms can be employed to solve variants of the Learning with Errors problem (LWE). In certain parameter regimes where the target vector is small and/or sparse, batches of BDD instances emerge from a combinatorial approach where several components of the target vector are guessed before decoding. In this work we explore trade-offs in solving "Batch-BDD", and apply our techniques to the small-secret Learning with Errors problem. We compare our techniques to previous works which solve batches of BDD instances, such as the hybrid lattice-reduction and meet-in-the-middle attack. Our results are a mixed bag. We show that, in the "enumeration setting" and with BKZ reduction, our techniques outperform a variant of the hybrid attack which does not consider time-memory trade-offs in the guessing phase for certain Round5 (17-bits out of 466), Round5-IoT (19-bits out of 240), and NTRU LPrime (23-bits out of 385) parameter sets. On the other hand, our techniques do not outperform the Hybrid Attack under standard, albeit unrealistic, assumptions. Finally, as expected, our techniques do not improve on previous works in the "sieving setting" (under standard assumptions) where combinatorial attacks in general do not perform well.

Keywords: Bounded Distance Decoding, cryptanalysis, hybrid attack, lattice-based cryptography, LWE, NTRU.

1 Introduction

The *Bounded Distance Decoding* problem with parameter $0 < \alpha$ asks to find the closest vector in some lattice $\Lambda \subset \mathbb{R}^d$ to some target vector $\mathbf{t} \in \mathbb{R}^d$ under

* The research of Albrecht was supported by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701) and EPSRC grants EP/S02087X/1 and EP/S020330/1. The research of Curtis was supported by the EPSRC and the UK government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/K035584/1).

the guarantee that the distance between the lattice and $\mathbf{t}$ is at most $\alpha \cdot \lambda_1(\Lambda)$, where $\lambda_1(\Lambda)$ is the length of a shortest vector in Λ. Establishing the concrete cost of solving BDD has received renewed attention in recent years because algorithms for solving this problem give rise to cryptanalytic attacks on schemes based on the hardness of the *Learning with Errors* problem (LWE) [Reg05] as well as the NTRU problem [HPS96]. These problems have been established as popular building blocks for realising post-quantum secure primitives such as public key encryption [HPS96, Reg05], key encapsulation [SAB+17, SHRS17], key exchange [LP11, DXL12, ADPS16] and digital signatures [BAA+17, PFH+17] as well as advanced primitives such as fully homomorphic encryption (FHE) [GSW13, BGV14].

Informally, LWE challenges an adversary to determine the secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ given $(\mathbf{A}, \mathbf{b}) \in (\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n)$ from a noisy linear system $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, where $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is uniformly random, and the error vector $\mathbf{e} \in \mathbb{Z}_q^m$ is drawn from some distribution χ producing small entries. In what follows, we will assume that the vector $\mathbf{s}$ also has short entries. If $\mathbf{s}$ follows χ then this is known as *normal form* LWE and is no easier than if $\mathbf{s}$ is uniformly random [ACPS09].

Similarly, NTRU challenges an adversary to recover (small multiples of) f, g in some polynomial quotient ring R, given $h = f^{-1} \cdot g$, where f is sampled to have an inverse and f, g are sampled from some distributions χ_f, χ_g producing small entries.

Both of these problems can be solved using an algorithm solving the *unique Shortest Vector Problem* (uSVP) and this approach is often considered in the literature. For LWE, we consider the lattice

$$\{(\mathbf{x}, \mathbf{y}, c) \in \mathbb{Z}^{n+m+1} \mid \mathbf{A} \cdot \mathbf{x} + \mathbf{y} - c \cdot \mathbf{b} \equiv \mathbf{0} \bmod q\},$$

for NTRU we consider the lattice

$$\{(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}^{2n} \mid \mathbf{H} \cdot \mathbf{x} - \mathbf{y} \equiv \mathbf{0} \bmod q\}$$

where $\mathbf{H}$ is the matrix produced by considering $h \cdot x^i$ for $0 \leq i < n$ with n being the degree of the ring R. These lattices contain $(\mathbf{s}, \mathbf{e}, 1)$ resp. $(\mathbf{f}, \mathbf{g})$ – the coefficient vectors of f and g – which are, for typical choices of parameters, unusually short. To solve uSVP, we may employ a lattice reduction algorithm such as BKZ [SE94, CN11] to find this unusually short vector using the success condition

$$\sqrt{\beta/d} \cdot \lambda_1(\Lambda) \leq \delta^{2\beta - d} \cdot \det(\Lambda)^{1/d}$$

from [ADPS16], which was experimentally verified in [AGVW17].

The lattice considered in the case of LWE is an embedding lattice for solving BDD via uSVP. We may also tackle BDD directly by first running lattice reduction to find a basis of sufficiently good quality for the lattice

$$\{(\mathbf{x},\ \mathbf{A} \cdot \mathbf{x} \bmod q) \mid \mathbf{x} \in \mathbb{Z}^n\}$$

followed by either Babai's Nearest Plane algorithm [Bab86] or pruned enumeration [LP11, LN13]. This is known as the *decoding attack* in the cryptographic

literature. Note that we may think of Babai's Nearest Plane algorithm as a form of pruned enumeration where all pruning coefficients are small enough to enforce the "Babai branch" of the search tree.

In the case of NTRU, BDD instances emerge from the *Hybrid Attack* [How07, GvVW17, Wun19] which combines guessing some coefficients of f or g and Babai's Nearest Plane algorithm. Due to the low cost, this algorithmic choice is natural as the adversary has to perform many calls to the BDD oracle: one for each guess. Thus, the algorithm has two phases: (a) a *lattice reduction phase* producing a sufficiently orthogonal basis used later and (b) a *guess and verify phase* where guesses are verified by running a BDD solver against the previously reduced basis and a target vector derived from the particular guess. The second step can – and is typically considered to – be realised using a meet-in-the-middle (mitm) or time-memory trade-off approach. The Hybrid Attack can be extended to LWE [BGPW16].

Contribution. From the discussion above we may consider the Hybrid Attack as a form of batched (candidate) BDD enumeration, where many points need to be decoded against the same lattice. Furthermore, we may consider the uSVP embedding approach and the Nearest Plane algorithm as endpoints of a continuum of strategies for solving (batch) BDD: the final enumeration is either (essentially) as expensive as the initial lattice reduction or optimised to be as cheap as possible to decode a large number of points. In this work we explore this continuum of strategies for solving LWE instances with small (and sparse) secrets $\mathbf{s}$ such as [GHS12, GMZB$^+$19, BCLvV19]. That is, we trade lattice-reduction preprocessing cost with BDD enumeration cost to reduce the overall cost of BDD. We note that in our parameterisation our algorithm solves many BDD-like instances with $\alpha \approx 1$ where a unique solution is not guaranteed to exist. In other words, we actually solve many instances of CVP.

In more detail, we present a *guess-and-verify* decoding approach which, like the Hybrid Attack, makes use of a guessing approach to reduce the dimension of the BDD problem. However, in our guess-and-verify decoding, we employ a more expensive BDD solver than Babai's Nearest Plane, i.e. we enumerate candidate solutions rather than just following the Babai branch. To establish the dimension in which we perform enumeration, we deploy (a slight variant of) the success condition from [ADPS16], i.e. we pick parameters so that that the distance between our target and the projected sub-lattice is slightly smaller than the expected shortest vector in that sub-lattice. Therefore, as opposed to applying a low probability BDD solver on a large number of (candidate) BDD instances, our technique applies heavier enumeration, with a higher probability of success, to a smaller number of (candidate) BDD instances.

Findings. Our results are presented in Tables 1 and 2 (see also Tables 3 and 4). We apply our techniques on parameter sets for NTRU LPrime [BCLvV19], Round5 [GMZB$^+$19] and HElib [Hal18].

Table 1 considers the enumeration setting, where the SVP oracle is realised using lattice-point enumeration [FP85, Kan83] and highlights that our non-mitm variant outperforms the non-mitm variant of the Hybrid Attack for the Round5, Round5-IoT and NTRU LPrime parameter sets by 17-bits, 19-bits and 23-bits respectively. These results assume the basis shape after lattice reduction exhibits an HKZ-shape in the last block, as observed in practice for BKZ reduction and predicted by the BKZ simulator [CN11], instead of a "line" as predicted by the Geometric Series Assumption (GSA). We also include estimates assuming the GSA holds also in the last block. In this setting, the non-mitm variant of the Hybrid Attack outperforms our techniques. We further note that for HElib our approach closes the gap between the dual and the primal attack observed in [Alb17]. Note that here, since β is relatively small, the output basis shape from the BKZ Simulator is very close to the Geometric Series Assumption (GSA), and thus the results are similar in each case.

Table 2 considers the sieving setting, where the SVP oracle is instantiated using a lattice sieving algorithm [AKS01, BDGL16]. Here, combinatorial approaches (i.e. guessing components of the secret) do not improve the running time of a BDD approach for the Round5, Round5-IoT, and NTRU LPrime parameter sets. Thus, our approach only marginally outperforms the uSVP attack by decoupling β and η, i.e. our approach reduces to the usual "decoding" approach [LP11, LN13] translated to the sieving setting (see also [ADH$^+$19]). Such estimates are marked by † in all tables.

We stress that a "g-v decoding" estimate that is lower than a "hybrid" estimate does not necessarily imply an invalidation of any security claim made by the designers of the schemes considered in our work, and to highlight this point we consider the (pre-quantum) security claim of each scheme, denoted by λ, in all of our tables. In particular, there are several points within our analysis in which we have had to make assumptions, and, whilst the assumptions we have chosen are reasonable based on currently known techniques, designers have made different assumptions to ours, and this can change the ordering of attack complexities. We consider a spectrum of such assumptions and their effect in Appendix B.

Limitations & Future Work. The Hybrid Attack is defined as a hybrid of a meet-in-the-middle guess-and-verify step and lattice reduction. For a guess $\mathbf{v}_g$ we can decode using Babai's Nearest Plane algorithm in the hope of finding the remaining components of the secret $\mathbf{v}_l$. In a meet-in-the-middle step, the guessed part of the secret $\mathbf{v}_g$ is split into two sub-guesses $\mathbf{v}_g = \mathbf{v}'_g + \mathbf{v}''_g$, and we in turn have *two* applications of Babai's Nearest Plane: one for each "half" of the original guess. Then, each decoded vector is stored in a hash table using a locality sensitive hash function [Wun19] which permits to find collisions in this table which, with some probability, correspond to $\mathbf{v}_g$. Whilst this approach allows the correct vector $\mathbf{v}_g$ to be found more quickly by reducing the search space for guessing, it also introduces an additional probability of failure. That is, we have to hope that the output of our BDD solver is homomorphic: if the guess $\mathbf{v}'_g$ corresponds to $\mathbf{v}'_l$ and $\mathbf{v}''_g$ corresponds to $\mathbf{v}''_l$, then we hope that $\mathbf{v}'_l + \mathbf{v}''_l = \mathbf{v}_l$. The

attack	τ	β	η	BDD cost	$\lvert S\rvert$	repeats	d	#pp	$\log_2(\mathsf{rop})$
NTRU LPrime: $n = 761, q = 4591, \sigma = \sqrt{2/3}, h = 250$									$\lambda = 222$
uSVP (GSA)	92	458	458	n/a	1	$2^{57.2}$	1220	n/a	384.6
Dual (GSA)	69	495	n/a	n/a	n/a	$2^{320.9}$	1281	11	374.0
g-v decoding (GSA)	285	430	102	$2^{336.1}$	$2^{252.8}$	$2^{34.1}$	1026	55	337.6
non mitm hybrid (GSA)	275	400	n/a	$2^{324.1}$	$2^{255.6}$	$2^{49.6}$	1036	57	**325.7**
g-v decoding	225	435	272	$2^{360.9}$	$2^{146.4}$	$2^{53.9}$	1086	28	**362.1**
non mitm hybrid	305	395	n/a	$2^{384.3}$	$2^{252.3}$	$2^{113.1}$	1006	53	385.3
Round5: $n = 756, q = 2^{12}, \sigma \approx 4.61, h = 242$									$\lambda = 270$
uSVP (GSA)	230	449	449	n/a	1	$2^{160.1}$	936	n/a	478.9
Dual (GSA)	63	626	n/a	n/a	n/a	$2^{413.5}$	1227	19	489.2
g-v decoding (GSA)	365	490	117	$2^{415.2}$	$2^{297.9}$	$2^{60.2}$	814	62	416.9
non mitm hybrid (GSA)	335	445	n/a	$2^{391.0}$	$2^{295.7}$	$2^{76.8}$	844	64	**392.5**
g-v decoding	290	490	320	$2^{448.2}$	$2^{157.2}$	$2^{92.6}$	889	28	**449.6**
non mitm hybrid	365	420	n/a	$2^{465.5}$	$2^{274.2}$	$2^{172.9}$	814	55	466.6
Round5 (IoT): $n = 372, q = 2^{11}, \sigma \approx 4.61, h = 178$									$\lambda = 129$
uSVP (GSA)	0	335	335	n/a	n/a	1	682	n/a	220.0
Dual (GSA)	32	334	n/a	n/a	n/a	$2^{174.7}$	661	14	221.7
g-v decoding (GSA)	65	315	224	$2^{213.1}$	$2^{79.2}$	$2^{9.0}$	616	22	214.3
non mitm hybrid (GSA)	115	270	n/a	$2^{203.6}$	$2^{149.5}$	$2^{36.9}$	566	43	**205.5**
g-v decoding	50	320	266	$2^{220.4}$	$2^{51.6}$	$2^{12.8}$	631	13	**221.4**
non mitm hybrid	120	270	n/a	$2^{239.6}$	$2^{150.8}$	$2^{71.5}$	561	42	240.6
HElib-1024: $n = 1024, q = 2^{47}, \sigma \approx 3.19, h = 64$									
uSVP (GSA)	140	105	105	n/a	1	$2^{14.0}$	1670	n/a	75.5
Dual (GSA)	189	107	n/a	n/a	n/a	$2^{22.3}$	1680	7	68.4
g-v decoding (GSA)	185	100	48	$2^{66.7}$	$2^{29.5}$	$2^{9.9}$	1624	4	69.1
non mitm hybrid (GSA)	210	100	n/a	$2^{67.5}$	$2^{36.6}$	$2^{10.7}$	1599	5	69.9

τ is the (fixed) guessing dimension, β is the blocksize used in lattice reduction, η is the enumeration dimension considered, BDD cost is the cost of solving BDD (via enumeration) in the dimension η projected sublattice, $|S|$ is the size of the search space considered, i.e the number of points on which we decode (chosen as a union $\cup_{i=0}^{\#pp}\mathcal{S}_i$ of sets $\mathcal{S}_i$ containing all length τ vectors with Hamming weight i), d is the dimension of the lattice considered, #pp denotes the maximal hamming weight considered in the search space, and rop is the cost of running the algorithm in CPU cycles; "g-v decoding" is the technique described in this work. Best in class are highlighted in bold where meaningful. "λ" values outline the security claims of each scheme, considering similar (pre-quantum) cost models and (pre-quantum) attacks; we note that such values of λ can be generated using vastly different assumptions.

Table 1: Estimates in enumeration setting.

probability that this occurs has been analysed in the case that the BDD solver is Babai's Nearest Plane in [Wun19]. However, this refined model is not employed e.g. in submissions to the NIST PQC process [BCLvV17, SHRS17, ZCHW17].

Our techniques share the same genealogy as the Hybrid Attack. However, in the main body of this work we consider only a setting where the guess-and-verify

attack	τ	β	η	BDD cost	$\lvert S\rvert$	repeats	d	#pp	$\log_2(\mathsf{rop})$
NTRU LPrime: $n = 761, q = 4591, \sigma = \sqrt{2/3}, h = 250$									$\lambda = 155$
uSVP (GSA)	0	532	532	n/a	n/a	1	1352	n/a	185.1
Dual (GSA)	45	586	n/a	n/a	n/a	$2^{148.0}$	1383	14	203.1
g-v decoding (GSA)	0	515	549	$2^{179.7}$	1	1	1351	n/a	**181.0**†
non mitm hybrid (GSA)	170	580	n/a	$2^{218.9}$	$2^{178.1}$	$2^{21.4}$	1181	43	220.8
g-v decoding	0	530	562	$2^{183.5}$	1	1	1351	n/a	**185.3**†
non mitm hybrid	230	615	n/a	$2^{302.1}$	$2^{189.6}$	$2^{93.3}$	1121	40	303.3
Round5: $n = 756, q = 2^{12}, \sigma \approx 4.61, h = 242$									$\lambda = 193$
uSVP (GSA)	0	664	664	n/a	n/a	1	1266	n/a	223.6
Dual (GSA)	46	748	n/a	n/a	n/a	$2^{198.6}$	1325	13	251.0
g-v decoding (GSA)	0	645	679	$2^{217.7}$	1	1	1265	n/a	**218.9**†
non mitm hybrid (GSA)	225	705	n/a	$2^{273.7}$	$2^{215.4}$	$2^{39.3}$	1040	49	275.2
g-v decoding	0	660	699	$2^{223.5}$	1	1	1265	n/a	**224.1**†
non mitm hybrid	290	700	n/a	$2^{393.9}$	$2^{214.8}$	$2^{160.3}$	975	43	394.9
Round5 (IoT): $n = 372, q = 2^{11}, \sigma \approx 4.61, h = 178$									$\lambda = 96$
uSVP (GSA)	0	335	335	n/a	n/a	1	682	n/a	126.6
Dual (GSA)	22	396	n/a	n/a	n/a	$2^{104.2}$	710	1	145.0
g-v decoding (GSA)	0	315	349	$2^{121.6}$	1	1	681	n/a	**122.4**†
non mitm hybrid (GSA)	85	375	n/a	$2^{155.2}$	$2^{119.5}$	$2^{18.3}$	596	38	156.9
g-v decoding	0	320	358	$2^{123.9}$	1	1	681	n/a	**124.3**†
non mitm hybrid	95	380	n/a	$2^{204.5}$	$2^{122.1}$	$2^{65.1}$	586	35	205.6
HElib-1024: $n = 1024, q = 2^{47}, \sigma \approx 3.19, h = 64$									
uSVP (GSA)	0	137	137	n/a	n/a	1	1939	n/a	70.3
Dual (GSA)	80	115	n/a	n/a	n/a	$2^{19.6}$	1741	7	67.1
g-v decoding (GSA)	85	125	50	$2^{66.6}$	$2^{30.0}$	$2^{2.6}$	1853	5	69.8
non mitm hybrid (GSA)	155	115	n/a	$2^{66.2}$	$2^{40.1}$	$2^{5.6}$	1783	6	69.8

Table 2: Estimates in the sieving setting, where BKZ and the BDD solver are instantiated with sieving algorithms. Notation as in Table 1. Estimates marked with † correspond to standard BDD decoding.

step is executed without the time-memory trade-off. In Appendix A, then, we also consider a meet-in-the-middle approach or time-memory trade-off where we (a) assume that collisions occur with probability one, (b) assume a square-root speed-up in the search phase and (c) ignore the memory cost.

Although our work uses preprocessing to control the dimension of the CVP problems we then have to solve in batch, we do not make specific use of techniques for solving the *Closest Vector Problem with Preprocessing* (CVPP) [Mic01]. In particular, the works [Laa16, DLdW16] discuss a time-memory trade-off in the sieving setting. After lattice reduction a *preprocessing* step can be deployed, generating a list of short vectors in some projected sublattice which can then be used to carry out many, typically cheaper, *query* steps (one for each guess). In the enumeration setting, pre-computing many reduced lattice bases allows lattice-point enumeration to be run many times, each with a low probability of

success, and without incurring an additional cost for preprocessing. Thus, these techniques offer the potential for improvements to our techniques when used instead of our assumption that the cost of solving CVP is the same as the cost of solving SVP.

We also note that this work naturally gives rise to a quantum variant where quantum algorithms are used for enumeration and for the guess-and-verify phase.

In summary, we note that any analysis of variants of the Hybrid Attack requires usage of several assumptions: (a) usage of a sieving-based, or enumeration-based, SVP oracle in the lattice reduction phase (of which we consider both), (b) an output lattice basis shape, namely GSA, Z-shaped, or usage the BKZ Simulator (of which we consider the BKZ Simulator and the GSA), (c) the guessing strategy, namely brute-force, meet-in-the-middle, or quantum (of which we assume brute-force, and also consider a square-root-style meet-in-the-middle search in Appendix A), (d) the choice of BDD/CVP solver i.e Babai's algorithm, pruned enumeration, or sieving (of which we consider all three possibilities) with or without preprocessing (which we do not consider). Thus, while our work explores trade-offs in batch BDD solving it still leaves many possible variants of such trade-offs unexplored and a full investigation of the concrete cost of solving this problem is still outstanding.

Related Work. The hardness of small-secret LWE was considered in [BLP+13], which gives a reduction from LWE in dimension n with secrets sampled from $\mathbb{Z}_q^n$ to LWE in dimension $n \log q$ with a secret sampled uniformly over $\{0,1\}^n$. This reduction has recently been revisited in [Mic18]. Several of the NTRU and LWE-based schemes submitted to the NIST standardisation process make use of small secrets [CPL+17, GMZB+19, SPL+17, BCLvV19].

As mentioned above, several works have explored the cost of solving CVP with Preprocessing such as [Mic01, Laa16, DLdW16]

Algorithmically, our work clearly builds on the line of works exploring hybrids of combinatorial and lattice reduction algorithms such as [MS01, How07, BGPW16, Alb17, GvVW17, Wun19]. We may also consider this work as a parameter space exploration for a specialisation of [LN13] to the case of batch BDD where many (candidate) BDD instances need to be solved. We note that the "decoding attack" is one of the three attacks considered by default in the LWE Estimator [APS15]. Thus, this work may also be considered as an investigation into the effectiveness of this attack compared to the other two considered there ("uSVP" and "Dual").

2 Preliminaries

Notation. We denote columns vectors by lower case bold letters, e.g. $\mathbf{b}$. Matrices are represented by upper case bold letters, e.g. $\mathbf{B}$. We denote the i^{th} component of the vector $\mathbf{b}$ by b_i, where i begins at one, and similarly the $(i,j)^{th}$ entry of a matrix by $B_{i,j}$. We write $\mathbf{B}_i$ for the i^{th} column of $\mathbf{B}$. Abusing notation, we denote by $(\mathbf{v}_1, \mathbf{v}_2, c)$ the vector formed by concatenating the entries of $\mathbf{v}_1$, $\mathbf{v}_2$

and the scalar c. We denote the i^{th} unit vector as $\mathbf{u}_i$. A *lattice* $\Lambda = \Lambda(\mathbf{B})$ is a discrete subgroup of $\mathbb{R}^n$ which can be characterised by a (column) basis $\mathbf{B}$: $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d\}$ which can itself be represented in matrix form $\mathbf{B} = [\mathbf{b}_1 \mid \mathbf{b}_2 \mid \cdots \mid \mathbf{b}_d]$. When $d = n$, the lattice has full rank. We denote the corresponding Gram-Schmidt orthogonalised (GSO) vectors by $\mathbf{b}_i^*$. We write $\mathbf{B}_{(\tau)}$ to represent the $d \times (d - \tau)$ submatrix of $\mathbf{B}$ constructed via dropping τ random columns of $\mathbf{B}$, i.e. $\mathbf{B}_{(\tau)} = [\mathbf{b}_{i_1} \mid \mathbf{b}_{i_2} \mid \cdots \mid \mathbf{b}_{i_{d-\tau}}]$ for some indices $1 \leq i_1 < i_2 < \cdots < i_{d-\tau} \leq d$. We similarly write $\mathbf{b}_{(\tau)}$ to denote dropping the corresponding τ components of the vector $\mathbf{b}$. We write $\pi_i(\mathbf{x})$ to denote the orthogonal projection of $\mathbf{x}$ onto the space spanned by the set of vectors $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{i-1}\}$. We denote by $\lambda_i(\Lambda)$ the i^{th} successive minima of the lattice Λ, i.e. the radius of the smallest ball, centred at the origin, containing at least i linearly independent lattice vectors. Unless stated otherwise, our logarithms are to base two.

$T_{\mathsf{bdd}}(\eta)$ denotes the cost of solving BDD in a dimension η projected lattice (typically $\pi_{d-\eta+1}(\Lambda)$). This process has probability p_{bdd} of returning the projection $\pi_{d-\eta+1}(\mathbf{v})$ of a closest (in the projected sublattice) lattice vector $\mathbf{v}$ to our target $\mathbf{t}$. Typically, we will have $p_{\mathsf{bdd}} \approx 1$. We denote the probability of correctly guessing τ random zeros of the LWE secret (where τ is the fixed guessing dimension) by p_0. The probability p_i for $1 \leq i \leq \min(\tau, h)$ represents the probability that the guessed components of the LWE secret contain i non-zero components. $\mathcal{S}_j$ is the set of all ternary vectors of length τ with Hamming weight j. Furthermore,

$$p_{\mathsf{babai}} \approx \prod_{1 \leq i \leq d} \left(1 - \frac{2}{B(\frac{d-1}{2}, \frac{1}{2})} \int_{\min(r_i, 1)}^{1} (1 - t^2)^{(d-3)/2} \, dt\right)$$

is the (heuristic) probability of Babai's algorithm lifting the projected short vector (found via solving BDD) to the full lattice [Wun19]. Here d is the number of dimensions we have to lift the projected solution through, and $r_i = \|\mathbf{b}_i^*\|/2\|\mathbf{v}\|$ where $\|\mathbf{v}\|$ is the (expected) norm of the target vector, and $B(\cdot, \cdot)$ denotes the Beta function. We cost this lifting process as in [Wun19] to be $T_{\mathsf{lift}} = d^2/2^{1.06}$.

Our work is concerned with the following computational problem:

Definition 1 (α-Bounded Distance Decoding ($\mathbf{BDD}_\alpha$)). *Given a lattice basis $\mathbf{B}$, a vector $\mathbf{t}$, and a parameter $0 < \alpha$ such that the Euclidean distance $\mathit{dist}(\mathbf{t}, \mathbf{B}) < \alpha\lambda_1(\mathbf{B})$, find the lattice vector $\mathbf{v} \in \Lambda(\mathbf{B})$ which is closest to $\mathbf{t}$.*

As discussed above, we can use a BDD_α solver for solving the NTRU and LWE problems. In this case we have $\alpha < 1/2$ which guarantees unique decoding (up to signs and rotations in the case of NTRU). To solve these instances, we may repeatedly call a BDD solver on smaller lattices where $\alpha \approx 1$. When solving the instances with $\alpha \approx 1$ we do not have a guarantee of unique decoding, only an expectation. In this case, we might more appropriately refer to the instances as CVP instances.

Definition 2 (NTRU [HPS96]). *Let n, q be positive integers, $\phi \in \mathbb{Z}[x]$ be a monic polynomial of degree n, and $\mathbb{R}_q = \mathbb{Z}_q[x]/(\phi)$. Let $f \in \mathbb{R}_q^\times, g \in \mathbb{R}_q$ be small polynomials (i.e. having small coefficients) and $h = g \cdot f^{-1} \bmod q$.*

Search-NTRU *is the problem of recovering* f *or* g *given* h.
Decision-NTRU *is the problem of deciding if* h *is of the form* $h = g \cdot f^{-1}$ *or is chosen uniformly at random.*

Definition 3 (LWE [Reg05]). *Let* n, q *be positive integers,* χ *be a probability distribution on* $\mathbb{Z}$ *and* $\mathbf{s}$ *be a secret vector in* $\mathbb{Z}_q^n$. *We denote the LWE Distribution* $L_{\mathbf{s},\chi,q}$ *as the distribution on* $\mathbb{Z}_q^n \times \mathbb{Z}_q$ *given by choosing* $\mathbf{a} \in \mathbb{Z}_q^n$ *uniformly at random, choosing* $e \in \mathbb{Z}$ *according to* χ *and considering it as an element of* $\mathbb{Z}_q$, *and outputting* $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.
Search-LWE *is the problem of recovering the vector* $\mathbf{s}$ *from a collection* $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ *of samples drawn according to* $L_{\mathbf{s},\chi,q}$.
Decision-LWE *is the problem of distinguishing whether samples* $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ *are drawn from the LWE distribution* $L_{\mathbf{s},\chi,q}$ *or uniformly from* $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

LWE, as defined in [Reg05], makes use of a rounded Gaussian distribution for the error distribution χ. However, LWE is typically considered with a discrete Gaussian distribution [LP11]. In practice, many schemes choose bounded uniform error distributions $\mathcal{U}_{[a,b]}$ [LDK+17] or binomial distributions [PAA+17].

Many constructions in the literature make use of Ring-LWE [SSTX09, LPR10] or Module-LWE [LS15] where the vectors $\mathbf{a}_i$ are not uniformly random, but have structure induced by a ring or module. We can treat these problems as LWE by ignoring this additional structure. Since our work does not exploit this additional structure, this is how we will proceed.

Secret Distributions. LWE, as defined, samples a secret $\mathbf{s}$ uniformly at random from $\mathbb{Z}_q^n$. In practice, many schemes choose to restrict the space of potential secrets by sampling secrets which are *small* and/or *sparse* by sampling from some *secret distribution* χ_s. Below we outline typical choices for these secret distributions, extending the notation from [Alb17].

Definition 4 (Small Secret Distributions). *Let* n, q *be positive integers.*
$\mathcal{B}^-$ *is the probability distribution on* $\mathbb{Z}_q^n$ *where each component is independently sampled uniformly at random from* $\{-1, 0, 1\}$.
$\mathcal{B}_h^-$ *is the probability distribution on* $\mathbb{Z}_q^n$ *where components are sampled uniformly at random from* $\{-1, 0, 1\}$ *with the additional guarantee that exactly* h *components are non-zero.*
$\mathcal{B}_{(h_1,h_2)}^-$ *is the probability distribution on* $\mathbb{Z}_q^n$ *where components are sampled uniformly at random from* $\{-1, 0, 1\}$ *with the additional guarantee that exactly* h_1 *components are equal to* -1 *and exactly* h_2 *components are equal to* 1.

We refer to a *small-secret LWE instance* as an LWE instance which samples secrets from one of the distributions outlined in Definition 4. Examples of such parameter choices are seen in homomorphic encryption libraries HElib ($\mathcal{B}_{64}^-$) [Hal18] and SEAL ($\mathcal{B}^-$) [SEA18]. In the NTRU domain, f is typically drawn from the distribution $\mathcal{B}_h^-$ for some h. The NTRUPrime submission to

the NIST PQC standardisation process considers the distributions $\mathcal{B}_{250}^{-}$ and $\mathcal{B}_{286}^{-}$ [BCLvV19]. Some other schemes in the NTRU domain make use of the secret distribution $\mathcal{B}_{(\frac{h}{2},\frac{h}{2})}^{-}$, where h is the Hamming weight of the secret.

Lattice Reduction and BDD Solver Costs. We consider lattice reduction when instantiated with either enumeration or sieving for solving the shortest vector problem. In particular, we consider the BKZ algorithm [SE94] parametrised by a block size β which determines the running time (at least exponential in β) and output quality. For enumeration [FP85, Kan83], we consider the cost of lattice reduction using blocksize β on a lattice of dimension d to be

$$T_{\mathsf{BKZ}}(\beta, d) = 8\,d \cdot 2^{0.18728\beta\log(\beta) - 1.019\beta + 16.1} \text{ enum. nodes}$$

which is taken from [APS15] based on experiments from [CN11]. To translate from the number of nodes visited during enumeration to CPU cycles, the literature typically assumes one node $\approx$ 100 CPU cycles [dt16]. For sieving [AKS01, BDGL16], we consider the cost of lattice reduction using blocksize β on a lattice of dimension d to be:

$$T_{\mathsf{BKZ}}(\beta, d) = 8\,d \cdot 2^{0.292\beta + 16.4} \tag{1}$$

where the constant term is somewhat arbitrarily picked as in [APS15].

"Core"-style BKZ cost models used in e.g [ADPS16], where the cost of BKZ is equated to the cost of a single SVP call and lower order terms are simply set to zero, are not considered in our work. There are several reasons for this: first, these estimates do not claim to capture the running time but constitute explicit lower bounds. Second, as a consequence, these costs suggest, in contrast to the state-of-the-art, that combinatorial techniques do not perform as well, especially in the sieving setting [ACD+18]. Third, since we are using the BKZ simulator from [CN11] to emulate the effect of lattice reduction more precisely, the number of tours has an effect on the output basis shape.

We also make use of both enumeration-based, and sieving-based, BDD_1/CVP solvers. When using enumeration to solve BDD_1/CVP in dimension η, we assume a cost of:

$$T_{\mathsf{bdd}}(\eta) = 2^{0.18728\eta\log(\eta) - 1.019\eta + 16.1} \text{ enum. nodes}$$

where again we assume that one node $\approx$ 100 CPU cycles [dt16]. Such an enumeration is assumed to succeed with probability close to one, i.e. $p_{\mathsf{bdd}} \approx 1$. When using sieving to solve BDD_1/CVP in dimension η, we assume a cost of:

$$T_{\mathsf{bdd}}(\eta) = 2^{0.292\eta + 16.4}$$

based on the results of [Laa16], which suggest that without preprocessing (see the discussion in the introduction) sieving for short vectors has the same asymptotic cost as sieving for close vectors. We assume that this sieving process suceeds with probability close to one, i.e $p_{\mathsf{bdd}} \approx 1$. We note that it is always clear from context whether an enumeration-based, or a sieving-based, BDD_1/CVP solver is being deployed.

2.1 The Geometric Series Assumption and 'Z-shaped' bases.

To determine the performance of the algorithms considered in this work, we are interested in the lengths of the GSO vectors after lattice reduction has taken place. We briefly recall the notion of the *root-Hermite factor* δ, which describes the quality of a basis after lattice reduction.

Definition 5 (root-Hermite factor). *For a basis* $\mathbf{B}$ *of a lattice* Λ *of dimension* d*, the root-Hermite factor is defined to be:*

$$\delta = \left(\frac{\|\mathbf{b}_1\|}{\det(\Lambda)^{1/d}} \right)^{1/d}.$$

For BKZ blocksizes considered in this work ($\beta \geq 40$), this value is well approximated [Che13] by $\delta^{2(\beta-1)} = \frac{\beta}{2\pi e}(\beta\pi)^{1/\beta}$. This value decreases towards 1 as the lattice reduction blocksize β increases.

In the context of the decoding attack, the lengths of the GSO vectors will determine the success probability of our BDD solvers. These lengths may be approximated the Geometric Series Assumption (GSA) [Sch03]:

Definition 6 (Geometric Series Assumption). *Let* $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d\}$ *be a basis of a lattice* Λ*, of quality* δ*, that is output by some lattice reduction algorithm. Then the lengths* $\|\mathbf{b}_i^*\|$ for $(1 \leq i \leq d)$ *of the Gram-Schmidt vectors of this basis are approximated by* $\|\mathbf{b}_i^*\| = \alpha^{i-1}\|\mathbf{b}_1\|$ *for some* $0 < \alpha < 1$.[3]

We can combine this assumption with $\|\mathbf{b}_1\| = \delta^d \cdot \det(\Lambda)^{1/d}$ and $\prod \|\mathbf{b}_i^*\| = \det(\Lambda)$ to determine $\alpha \approx \delta^{-2}$.

There are several models in the literature for the behaviour of the BKZ algorithm on q-ary lattices. In most of the literature on solving LWE via BDD, we use the public LWE matrix $\mathbf{A} \in \mathbb{Z}_q^{(m \times n)}$ to construct a lattice basis $\mathbb{Z}^{(n+m)\times(n+m)}$ for which it is commonly assumed that the Geometric Series Assumption is relatively accurate after running BKZ-β with $\beta \ll m + n$. The literature on analysing the Hybrid Attack considers lattice bases of the form

$$\begin{pmatrix} q\mathbf{I}_m & \mathbf{A} \\ 0 & \mathbf{I}_n \end{pmatrix}. \tag{2}$$

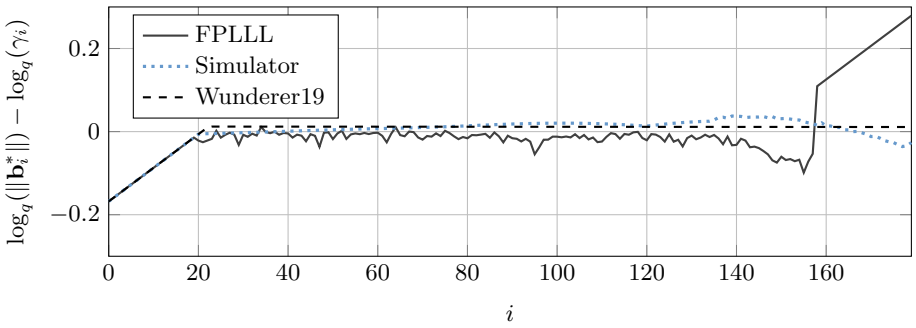
This form of writing the basis immediately suggests that the GSA might not hold. Specifically, when the GSA predicts that $\|\mathbf{b}_1\| > q$, this is longer than the first vector already in the basis before lattice reduction and thus we will obtain $\|\mathbf{b}_1\| = q$. As a consequence, lattice reduction is expected to produce a "Z-shaped" basis [How07], comprised of leading qs, trailing ones and a middle part approximated by the GSA.

[3] Note that, following the literature, we are overloading notation here: this α is unrelated to the BDD approximation factor α. It will always be clear from context which α we are referring to.

In Figure 1 we give an illustrative example, chosen to highlight the effect, of the output of lattice reduction as implemented in FPLLL [dt16] which clearly illustrates the Z-shape. To model this Z-shape, we can insist on no vector after lattice reduction having norm $> q$ [Wun19]. Let k be the number of lattice vectors which follow a GSA-style behaviour, and the other $(d-k)$ vectors remain as q. Explicitly, we have:

$$\|\mathbf{b}_i^*\| = \begin{cases} q & \text{if } i \leq d-k \\ \delta^{-2(i-(d-k)-1)+k} q^{\frac{k-n}{k}} & \text{otherwise.} \end{cases}$$

Making the assumption that $\|\mathbf{b}^*_{(d-k+1)}\| \approx q$ [Wun19], it is possible to compute a value for k, namely $k = \min\left(\left\lfloor \sqrt{\frac{n}{\log_q(\delta)}} \right\rfloor, d\right)$. Indeed, running the BKZ simulator from [Che13] will output a predicted shape closely resembling this prediction for $\beta \ll m+n$. On the other hand, [Wun19] makes no attempt to model the number of trailing ones. Indeed, while some works pick the length of this part by choosing a sublattice to reduce [HPS+15], no work in the literature offers a way of predicting the number of trailing ones.



We define $\gamma_i = \alpha^{i-1}\delta^d \det(\Lambda)^{1/d}$ and thus the GSA corresponds to the line at $y = 0$.

Fig. 1: Example of BKZ-60 reduction on a q-ary lattice of dimension $d = 180$ with $q = 17$ and volume 17^{80} for bases constructed as in (2), along with the output of BKZ simulation and the heuristic from [Wun19].

For this reason, in this work we assume that the q-ary structure of the lattice does not impact the shape of the basis after lattice reduction, i.e. we do not assume leading qs or trailing 1s. This assumption can be made to hold by rerandomising the input basis for lattice reduction. Considering the techniques in this work in a setting exploiting the q-ary structure is an interesting area for future work.

2.2 Decoding Small-secret LWE

The decoding approach for solving small-secret LWE instances [BG14] $\mathbf{b} = \mathbf{A}\cdot\mathbf{s}+\mathbf{e}$ constructs a lattice for which the vector $(\mathbf{b}, \mathbf{0})$ is separated by the short vector $(-\mathbf{e}, \mathbf{s})$ to the lattice point $(\mathbf{A}\cdot\mathbf{s} \bmod q, \mathbf{s})$. The basis of the lattice is given by the columns of the matrix $\mathbf{B}$, where

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_m & \mathbf{A} \\ 0 & \mathbf{I}_n \end{pmatrix}.$$

In more detail, we have that

$$\begin{pmatrix} q\mathbf{I}_m & \mathbf{A} \\ \mathbf{0} & \mathbf{I}_n \end{pmatrix} \begin{pmatrix} * \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} q* + \mathbf{A}\cdot\mathbf{s} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} -\mathbf{e} \\ \mathbf{s} \end{pmatrix} \bmod q.$$

After lattice reduction on the lattice spanned by $\mathbf{B}$, we perform enumeration around the target point $(\mathbf{b}, \mathbf{0})$, which is close to a unique lattice point. With some probability, this enumeration will return this unique closest lattice point, enabling recovery of the LWE secret. As noted in [MS01] we can combine this attack with *dimension reduction techniques*, where τ components of the secret are guessed (as zero) and the associated decoding problem is solved in dimension $(d-\tau)$. We refer to this variant as *drop-and-solve decoding*.

2.3 The Hybrid Meet-in-the-middle and Lattice Reduction Attack

The Hybrid meet-in-the-middle and lattice reduction strategy was introduced by Howgrave-Graham in [How07]. It leverages small and sparse target vectors by combining a lattice reduction phase with a guess and verify phase. As opposed to solving BDD in a dimension d lattice, the Hybrid Attack sets a guessing dimension τ, carries out lattice reduction in dimension $(d-\tau)$ and solves BDD on a dimension $(d-\tau)$ lattice by decoding on various points corresponding to guesses in the τ-dimensional guessing space. When $s \leftarrow\!\!\$\ \mathcal{B}^-$, the coefficients s_i of the secret vector are contained within the set $\{-1, 0, 1\}$. We then observe that

$$\mathbf{A}\cdot\mathbf{s} = \sum_{\{i|s_i=1\}} \mathbf{A}_i - \sum_{\{j|s_j=-1\}} \mathbf{A}_j. \tag{3}$$

We begin by choosing a guessing dimension τ and generate the lattice basis determined by the columns of the matrix

$$\mathbf{X} = \begin{pmatrix} q\mathbf{I}_m & \mathbf{A}_{(\tau)} \\ 0 & \mathbf{I}_{n-\tau} \end{pmatrix}$$

where $\mathbf{A}_{(\tau)}$ denotes the matrix $\mathbf{A}$ with τ random columns dropped. We then reduce this basis to $\tilde{\mathbf{X}} = \mathsf{BKZ}_\beta(\mathbf{X})$, and use it to carry out decoding using Babai's Nearest Plane [Bab86] on vectors in the τ-dimensional guessing space. We have:

$$\begin{pmatrix} q\mathbf{I}_m & \mathbf{A}_{(\tau)} \\ 0 & \mathbf{I}_{n-\tau} \end{pmatrix} \begin{pmatrix} * \\ \mathbf{s}_{(\tau)} \end{pmatrix} = \begin{pmatrix} q* + \mathbf{A}_{(\tau)}\cdot\mathbf{s}_{(\tau)} \\ \mathbf{s}_{(\tau)} \end{pmatrix}.$$

If we correctly guess zero components then we have $\mathbf{A}_{(\tau)} \cdot \mathbf{s}_{(\tau)} = \mathbf{A} \cdot \mathbf{s}$, allowing us to decode on the point $(\mathbf{b}, \mathbf{0})$. Otherwise, assuming without loss of generality that the last τ components of $\mathbf{s}$ were the guessed components, we can make a new guess $\mathbf{v}_g = \sum_{k=d-\tau+1}^{d} c_k \cdot \mathbf{u}_k$ for some values $c_k \in \{-1, 0, 1\}$. For this new guess, we can decode on the point $(\mathbf{b} - \sum_{k=n-\tau+1}^{n} c_k \cdot \mathbf{A}_k, 0)$. For the correct guess $\mathbf{v} = \sum_{k=n-\tau+1}^{n} s_k \cdot \mathbf{u}_k$ we have

$$\mathbf{b} - \sum_{k=n-\tau+1}^{n} s_k \cdot \mathbf{A}_k = \mathbf{A}_{(\tau)} \cdot \mathbf{s}_{(\tau)} + \mathbf{e} \bmod q.$$

Therefore, we have that $(\mathbf{b} - \sum_{k=d-\tau+1}^{d} s_k \cdot \mathbf{A}_k, 0)$ is separated from the lattice point $(\mathbf{A}_{(\tau)} \cdot \mathbf{s}_{(\tau)} \bmod q, \mathbf{s}_{(\tau)})$ by the vector $(-\mathbf{e}, \mathbf{s}_{(\tau)})$, as is required. Typically, this guessing is realised via the usage of a time-memory trade-off approach. As previously mentioned, throughout this work we do not consider any meet-in-the-middle techniques, except for compatibility with previous works in Appendix A.

3 A Spectrum of Decoding Approaches

In this section we outline the expected costs of: (i) the classical "decoding" approach in the LWE literature, (i) the "drop-and-solve decoding" approach, which is a combination of zero-guessing and solving a single BDD instance in a reduced dimension, and (ii) our *guess-and-verify* decoding approach, where multiple BDD instances are solved per lattice reduction step. Our attack parameters are chosen such that $p_{\mathsf{bdd}} \approx 1$ and $p_{\mathsf{babai}} \approx 1$, although for clarity we include these probabilities in the running times presented in this section. Recall that p_{bdd} corresponds to the probability of solving BDD in the dimension η projected sublattice, where η is chosen in our work such that the probability of lifting the solution to the full lattice is $p_{\mathsf{babai}} \approx 1$. Exploring trade-offs which arise by varying these probabilities is interesting future work.

Running example. Throughout this section we make use of a running example to illustrate the behaviour of the approaches under consideration. We consider the small-secret LWE parameter set $n = 653, q = 4621, \sigma \approx \sqrt{2/3}, \chi_s = \mathcal{B}_{100}^{-}$ and use this parameter set as a reference throughout. We note that for this parameter set, a combinatorial dual attack costs $2^{214.9}$ CPU cycles ($\beta = 210$), and a combinatorial uSVP attack, assuming use of the GSA, costs $2^{209.6}$ CPU cycles ($\beta = 223$), according to the LWE Estimator [APS15][4], under the enumeration-based BKZ cost model mentioned in Section 2.

Decoding. We start by outlining the expected running time of the decoding approach described in Section 2.2. Typically, the cost of lattice reduction and

[4] All estimates use the LWE Estimator as of commit `3019847`.

decoding are balanced, and the output BDD probability determines the number of times the algorithm is repeated. The total expected running time is

$$T_{\mathsf{Dec}} = \frac{T_{\mathsf{BKZ}}(\beta_{\mathsf{Dec}}, d) + T_{\mathsf{bdd}}(\eta_{\mathsf{Dec}})}{p_{\mathsf{babai}} \cdot p_{\mathsf{bdd}}}.$$

Lattice reduction is carried out on the full lattice with block size β_{Dec}, and a BDD_1 solver is used on a projected sub-lattice of dimension η_{Dec}, which is determined by (a variant of) the success condition in [ADPS16]. Here p_{babai} is the probability of lifting the candidate solution from $\pi_{d-\eta_{\mathsf{Dec}}+1}(\Lambda)$ to the full lattice. Since η_{Dec} is determined using the condition from [ADPS16] we have $p_{\mathsf{babai}} \approx 1$ [AGVW17]. For our running example parameter set, assuming the GSA, this approach has a cost of $2^{293.0}$ CPU cycles with an optimal blocksize $\beta_{\mathsf{Dec}} = 419$, where $\eta_{\mathsf{Dec}} = 429$. Note that when $\eta_{\mathsf{Dec}} = \beta_{\mathsf{Dec}}$ then this is equivalent to the uSVP approach in [ADPS16].

Drop-and-solve decoding. In this approach, we guess τ zero components of $\mathbf{s}$ and then run the decoding attack in dimension $(d - \tau)$ [MS01, Alb17, ACD+18]. If we are unsuccessful, we restart with a fresh guess for the positions of zeroes. The core idea is that the lower running time of the dimension-reduced problem will trade-off positively against the probability of guessing zero components. If we correctly guess, for example, the first τ zeros, then $(s_{\tau+1}, \ldots, s_n, \mathbf{e})$ can be found via solving BDD_1 in the dimension-reduced problem. The total expected running time of this strategy is

$$T_{\mathsf{dsDec}} = \frac{T_{\mathsf{BKZ}}(\beta_{\mathsf{dsDec}}, d - \tau) + T_{\mathsf{bdd}}(\eta_{\mathsf{dsDec}})}{p_{\mathsf{babai}} \cdot p_{\mathsf{bdd}} \cdot p_0}.$$

Here p_0 denotes the probability of correctly guessing τ random zeros of the LWE secret. Lattice reduction is carried out on a lattice of dimension $(d - \tau)$ with block size β_{dsDec}, and enumeration is carried out in the projected sub-lattice whose dimension corresponds again to (a variant of) the [ADPS16] success condition. The meaning of p_{babai} is as above. For our running example parameter set, assuming the GSA, this attack returns a complexity of $2^{208.2}$ CPU cycles with optimal values of $\beta_{\mathsf{dsDec}} = 170$ and $\tau = 315$.

Guess-and-verify decoding. There is no a priori reason to restrict the decoding algorithm in any guess-and-verify decoding attack to Babai's Nearest Plane algorithm (as in the hybrid attack). Instead, we may employ stronger BDD_1 solvers which in turn permits a reduction in the cost of preprocessing, or the usage of a lower guessing dimension. In this *g-v decoding* attack, we consider a BDD_1 dimension as defined by (a variant of) the success condition from [ADPS16]. The overall expected cost of this approach then becomes

$$T_{\mathsf{gvDec}} = \frac{T_{\mathsf{BKZ}}(\beta_{\mathsf{gvDec}}, d - \tau) + \|\mathcal{S}_{t_{\mathsf{gvDec}}}\| \cdot T_{\mathsf{bdd}}(\eta_{\mathsf{gvDec}})}{p_{\mathsf{babai}} \cdot p_{\mathsf{bdd}} \cdot (\sum_{i=0}^{t_{\mathsf{gvDec}}} p_i)}.$$

Where the probabilities p_i for $1 \leq i \leq t_{\mathsf{gvDec}}$ represents the probability that the guessed components of the LWE secret contain i non-zero components. For our running example parameter set, assuming the GSA, this attack returns a complexity of $2^{186.1}$ CPU cycles with $\beta_{\mathsf{gvDec}} = 225$ and $\tau = 335$, with optimal choices of $\eta_{\mathsf{gvDec}} = 49$ $t_{\mathsf{gvDec}} = 16$ so that $\|\mathcal{S}_{t_{\mathsf{gvDec}}}\| = \sum_{i=0}^{16} \binom{335}{i} \cdot 2^i \approx 2^{105.5}$.

We note that "guess-and-verify" decoding encompasses the usual decoding strategy ($\tau = 0, \|\mathcal{S}_{t_{\mathsf{gvDec}}}\| = 1, \sum p_i = 1$) and the "drop-and-solve" strategy ($\tau > 0, \|\mathcal{S}_{t_{\mathsf{gvDec}}}\| = 1, t_{\mathsf{gvDec}} = 0$). On the other hand, as specified here, it does not encompass the hybrid attack (even without time-memory trade-offs) since we insist on picking β, η such that $p_{\mathsf{babai}} \approx 1$ which is not the case for the Hybrid Attack in general.

4 Estimates

In this section we apply our techniques to parameter sets from the NTRU-LPrime [BCLvV19] and Round5 [BGL+18] submissions to the NIST PQC standardisation process, as well as a parameter set used in the homomorphic encryption library HElib [Hal18]. We compare our results against the LWE estimator under the same assumptions, i.e. considering the cost models in Section 2 and the Geometric Series Assumption. We also present our results considering usage of the BKZ simulator, which estimates the shape of the basis after BKZ reduction. Our results are given in Tables 1 and 2.

NTRU LPrime. We consider one of the three NTRU LPrime parameter sets from [BCLvV19]. The construction is based on LWE with a ternary, fixed Hamming weight, secret and a random ternary error. Specifically, the parameter set considered is:

$$n = 761, q = 4591, \sigma \approx \sqrt{2/3}, \chi_s = \mathcal{B}_{250}^{-}.$$

Round5. For Round5, we consider the NIST level 3 parameter set from [GMZB+19]. Round 5 is based on the Learning with Rounding problem (LWR) [BPR12] with a ternary, fixed hamming weight, secret. In the case of LWR, we have an additional parameter p which is an additional modulus considered in the deterministic rounding process. In this case, we have $\sigma \approx \sqrt{\frac{(q/p)^2-1}{12}}$ as in [ACD+18]. We can therefore model this parameter set as LWE with

$$n = 756, \sigma \approx 4.61, q = 2^{12}, p = 2^8, \chi_s = \mathcal{B}_{242}^{-}.$$

We also consider the IoT specific use-case parameter set from [GMZB+19]. We can model this parameter set as LWE with

$$n = 372, \sigma \approx 4.61, q = 2^{11}, p = 2^7, \chi_s = \mathcal{B}_{178}^{-}.$$

HElib. We also consider our approach in the context of the homomorphic encryption library HElib. To compare with previous works, we consider the sparse-secret parameter set outlined in [Alb17]. Specifically, the parameter set we consider is:

$$n = 1024, q = 2^{47}, \sigma \approx 3.19, \chi_s = \mathcal{B}_{64}^{-}.$$

Acknowledgements

The authors thank the anonymous SAC reviewers for their feedback, which has been used to improve this work.

References

ACD+18. Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the LWE, NTRU schemes! In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 351–367. Springer, Heidelberg, September 2018.

ACPS09. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Heidelberg, August 2009.

ADH+19. Martin R. Albrecht, Lo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. Cryptology ePrint Archive, Report 2019/089, 2019. `https://eprint.iacr.org/2019/089`.

ADPS16. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16*, pages 327–343. USENIX Association, 2016.

AGVW17. Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 297–322. Springer, Heidelberg, December 2017.

AKS01. Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *33rd ACM STOC*, pages 601–610. ACM Press, July 2001.

Alb17. Martin R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 103–129. Springer, Heidelberg, April / May 2017.

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

BAA+17. Nina Bindel, Sedat Akleylek, Erdem Alkim, Paulo S. L. M. Barreto, Johannes Buchmann, Edward Eaton, Gus Gutoski, Juliane Kramer,

Patrick Longa, Harun Polat, Jefferson E. Ricardini, and Gustavo Zanon. qtesla. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

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

BCLvV17. Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. Ntru prime. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

BCLvV19. Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. Ntru prime. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions`.

BDGL16. Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert Krauthgamer, editor, *27th SODA*, pages 10–24. ACM-SIAM, January 2016.

BG14. Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary LWE. In Willy Susilo and Yi Mu, editors, *ACISP 14*, volume 8544 of *LNCS*, pages 322–337. Springer, Heidelberg, July 2014.

BGL+18. Sauvik Bhattacharya, Óscar García-Morchón, Thijs Laarhoven, Ronald Rietman, Markku-Juhani O. Saarinen, Ludo Tolhuizen, and Zhenfei Zhang. Round5: Compact and fast post-quantum public-key encryption. Cryptology ePrint Archive, Report 2018/725, 2018. `https://eprint.iacr.org/2018/725`.

BGPW16. Johannes A. Buchmann, Florian Göpfert, Rachel Player, and Thomas Wunderer. On the hardness of LWE with binary error: Revisiting the hybrid lattice-reduction and meet-in-the-middle attack. In David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 16*, volume 9646 of *LNCS*, pages 24–43. Springer, Heidelberg, April 2016.

BGV14. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):13, 2014.

BLP+13. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 575–584. ACM Press, June 2013.

BPR12. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Heidelberg, April 2012.

Che13. Yuanmi Chen. *Rduction de rseau et scurit concrte du chiffrement compltement homomorphe*. PhD thesis, Paris 7, 2013.

CN11. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2011.

CPL+17. Jung Hee Cheon, Sangjoon Park, Joohee Lee, Duhyeong Kim, Yongsoo Song, Seungwan Hong, Dongwoo Kim, Jinsu Kim, Seong-Min Hong, Aaram Yun, Jeongsu Kim, Haeryong Park, Eunyoung Choi, Kimoon kim, Jun-Sub Kim, and Jieun Lee. Lizard. Technical report, National Institute of

Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

DLdW16. Emmanouil Doulgerakis, Thijs Laarhoven, and Benne de Weger. Finding closest lattice vectors using approximate voronoi cells. Cryptology ePrint Archive, Report 2016/888, 2016. https://eprint.iacr.org/2016/888.

dt16. The FPLLL development team. fplll, a lattice reduction library. Available at https://github.com/fplll/fplll, 2016.

Duc19. L. Ducas, 2019. Thread on pqc-forum. Available at https://groups.google.com/a/list.nist.gov/forum/#!topic/pqc-forum/JwR0_fpNujc.

DXL12. Jintai Ding, Xiang Xie, and Xiaodong Lin. A simple provably secure key exchange scheme based on the learning with errors problem. Cryptology ePrint Archive, Report 2012/688, 2012. http://eprint.iacr.org/2012/688.

FP85. U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of Computation*, 44(170):463463, May 1985.

GHS12. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 850–867. Springer, Heidelberg, August 2012.

GMZB+19. Oscar Garcia-Morchon, Zhenfei Zhang, Sauvik Bhattacharya, Ronald Rietman, Ludo Tolhuizen, Jose-Luis Torre-Arce, Hayo Baan, Markku-Juhani O. Saarinen, Scott Fluhrer, Thijs Laarhoven, and Rachel Player. Round5. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

GSW13. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, August 2013.

GvVW17. Florian Göpfert, Christine van Vredendaal, and Thomas Wunderer. A hybrid lattice basis reduction and quantum search attack on LWE. In *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, pages 184–202, 2017.

Hal18. Shai Halevi. Helib. https://github.com/shaih/HElib, 2018.

How07. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 150–169. Springer, Heidelberg, August 2007.

HPS96. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A new high speed public key cryptosystem, 1996. Draft Distributed at Crypto'96, available at http://web.securityinnovation.com/hubfs/files/ntru-orig.pdf.

HPS+15. Jeff Hoffstein, Jill Pipher, John M. Schanck, Joseph H. Silverman, William Whyte, and Zhenfei Zhang. Choosing parameters for NTRUEncrypt. Cryptology ePrint Archive, Report 2015/708, 2015. http://eprint.iacr.org/2015/708.

Kan83. Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *15th ACM STOC*, pages 193–206. ACM Press, April 1983.

Laa16. Thijs Laarhoven. Sieving for closest lattice vectors (with preprocessing). In Roberto Avanzi and Howard M. Heys, editors, *SAC 2016*, volume 10532 of *LNCS*, pages 523–542. Springer, Heidelberg, August 2016.

LDK+17. Vadim Lyubashevsky, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Peter Schwabe, Gregor Seiler, and Damien Stehle. Crystals-dilithium. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

LN13. Mingjie Liu and Phong Q. Nguyen. Solving BDD by enumeration: An update. In Ed Dawson, editor, *CT-RSA 2013*, volume 7779 of *LNCS*, pages 293–309. Springer, Heidelberg, February / March 2013.

LP11. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 319–339. Springer, Heidelberg, February 2011.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May / June 2010.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography*, 75(3):565–599, 2015.

Mic01. Daniele Micciancio. The hardness of the closest vector problem with preprocessing. *IEEE Trans. Information Theory*, 47(3):1212–1215, 2001.

Mic18. Daniele Micciancio. On the hardness of lwe with binary error. Technical report, February 2018. `http://cseweb.ucsd.edu/~daniele/papers/BinLWE.pdf`.

MS01. Alexander May and Joseph H. Silverman. Dimension reduction methods for convolution modular lattices. In *Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001, Revised Papers*, pages 110–125, 2001.

PAA+17. Thomas Poppelmann, Erdem Alkim, Roberto Avanzi, Joppe Bos, Leo Ducas, Antonio de la Piedra, Peter Schwabe, and Douglas Stebila. Newhope. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

PFH+17. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

SAB+17. Peter Schwabe, Roberto Avanzi, Joppe Bos, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehle. Crystals-kyber. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

Sch03. Claus Peter Schnorr. Lattice reduction by random sampling and birthday methods. In Helmut Alt and Michel Habib, editors, *STACS 2003*, pages 145–156, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

SE94. Claus-Peter Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66:181–199, 1994.

SEA18. Simple Encrypted Arithmetic Library (release 3.1.0). `https://github.com/Microsoft/SEAL`, December 2018. Microsoft Research, Redmond, WA.

SHRS17. John M. Schanck, Andreas Hulsing, Joost Rijneveld, and Peter Schwabe. Ntru-hrss-kem. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

SPL+17. Minhye Seo, Jong Hwan Park, Dong Hoon Lee, Suhri Kim, and Seung-Joon Lee. Emblem and r.emblem. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

SSTX09. Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, Heidelberg, December 2009.

Wun19. Thomas Wunderer. A detailed analysis of the hybrid lattice-reduction and meet-in-the-middle attack. *J. Mathematical Cryptology*, 13(1):1–26, 2019.

ZCHW17. Zhenfei Zhang, Cong Chen, Jeffrey Hoffstein, and William Whyte. Ntruencrypt. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

A Meet-in-the-middle

We consider a meet-in-the-middle approach where we assume (a) that collisions occur with probability 1, and (b) a square-root speed-up in the search phase. As discussed above, this modelling has been shown to be not correct [Wun19]. We include it here for compatibility with previous works which make similar assumptions, e.g [BCLvV19]. On the other hand, we assume "free memory" here, i.e. we do not take the cost of memory into account. As illustrated in Table 3 under these assumptions and in the enumeration setting the Hybrid Attack and "g-v decoding" are essentially on par; in the sieving setting, BDD decoding is the most efficient.

attack	τ	β	η	BDD cost	$\lvert S\rvert$	repeats	d	#pp	$\log_2(\mathsf{rop})$
NTRU LPrime: $n = 761, q = 4591, \sigma = \sqrt{2/3}, h = 250$									$\lambda = 222$
sqrt g-v decoding (GSA)	370	350	43	$2^{243.5}$	$2^{412.2}$	$2^{11.5}$	941	102	245.0
sqrt hybrid (GSA)	360	335	n/a	$2^{240.1}$	$2^{402.8}$	$2^{20.0}$	951	100	**241.3**
sqrt g-v decoding	370	380	119	$2^{273.6}$	$2^{400.0}$	$2^{15.5}$	941	97	**274.7**
sqrt hybrid	395	350	n/a	$2^{274.9}$	$2^{428.3}$	$2^{42.1}$	916	104	275.9
Round5: $n = 756, q = 2^{12}, \sigma \approx 4.61, h = 242$									$\lambda = 270$
sqrt g-v decoding (GSA)	445	395	37	$2^{283.9}$	$2^{490.0}$	$2^{14.1}$	734	120	285.5
sqrt hybrid (GSA)	425	365	n/a	$2^{277.0}$	$2^{453.9}$	$2^{32.0}$	754	109	**278.0**
sqrt g-v decoding	450	430	131	$2^{324.8}$	$2^{474.6}$	$2^{22.7}$	729	113	325.9
sqrt hybrid	460	390	n/a	$2^{320.5}$	$2^{496.6}$	$2^{54.3}$	719	120	**321.6**
Round5 (IoT): $n = 372, q = 2^{11}, \sigma \approx 4.61, h = 178$									$\lambda = 129$
sqrt g-v decoding (GSA)	175	250	48	$2^{156.7}$	$2^{250.8}$	$2^{4.0}$	506	80	157.8
sqrt hybrid (GSA)	165	225	n/a	$2^{151.6}$	$2^{234.5}$	$2^{17.4}$	516	74	**152.9**
g-v decoding	170	270	101	$2^{174.3}$	$2^{237.2}$	$2^{6.5}$	511	73	175.4
non mitm hybrid	180	240	n/a	$2^{172.2}$	$2^{256.4}$	$2^{27.1}$	501	81	**173.4**
HElib-1024: $n = 1024, q = 2^{47}, \sigma \approx 3.19, h = 64$									
sqrt g-v decoding (GSA)	210	95	36	$2^{59.9}$	$2^{59.7}$	$2^{5.2}$	1599	9	62.0
sqrt hybrid (GSA)	270	85	n/a	$2^{60.8}$	$2^{63.1}$	$2^{9.1}$	1539	9	61.8

Table 3: Estimates in enumeration setting considering a "meet-in-the-middle" approach which does not consider probabilities of failure in the meet-in-the-middle phase. Such an approach considers a square-root speed-up in the guessing phase. Notation as in Table 1.

attack	τ	β	η	BDD cost	$\lvert S\rvert$	repeats	d	#pp	$\log_2(\mathsf{rop})$
NTRU LPrime: $n = 761, q = 4591, \sigma = \sqrt{2/3}, h = 250$									$\lambda = 155$
sqrt g-v decoding (GSA)	0	515	549	$2^{179.7}$	1	1	1351	0	**181.0**†
sqrt hybrid (GSA)	260	475	n/a	$2^{181.3}$	$2^{296.5}$	$2^{13.9}$	1091	75	182.7
sqrt g-v decoding	0	530	562	$2^{183.5}$	1	1	1351	0	**185.3**†
sqrt hybrid	315	550	n/a	$2^{230.4}$	$2^{341.1}$	$2^{40.9}$	1036	83	231.7
Round5: $n = 756, q = 2^{12}, \sigma \approx 4.61, h = 242$									$\lambda = 193$
sqrt g-v decoding (GSA)	0	645	679	$2^{217.7}$	1	1	1265	0	218.9†
sqrt hybrid (GSA)	320	565	n/a	$2^{216.4}$	$2^{350.7}$	$2^{22.3}$	945	86	**217.5**
sqrt g-v decoding	0	660	699	$2^{223.5}$	1	1	1265	0	**224.1**†
sqrt hybrid	390	660	n/a	$2^{288.4}$	$2^{405.8}$	$2^{67.0}$	875	96	289.6
Round5 (IoT): $n = 372, q = 2^{11}, \sigma \approx 4.61, h = 178$									$\lambda = 96$
sqrt g-v decoding (GSA)	0	315	349	$2^{121.6}$	1	1	681	0	**122.4**†
sqrt hybrid (GSA)	135	300	n/a	$2^{126.9}$	$2^{196.9}$	$2^{11.4}$	546	65	128.2
sqrt g-v decoding	0	320	358	$2^{123.9}$	1	1	681	0	**124.3**†
sqrt hybrid	155	335	n/a	$2^{155.2}$	$2^{218.1}$	$2^{29.2}$	526	68	156.3
HElib-1024: $n = 1024, q = 2^{47}, \sigma \approx 3.19, h = 64$									
sqrt g-v decoding (GSA)	195	100	31	$2^{63.1}$	$2^{53.4}$	$2^{5.3}$	1743	8	65.1
sqrt hybrid (GSA)	235	95	n/a	$2^{61.7}$	$2^{72.1}$	$2^{5.2}$	1703	11	63.6

Table 4: Estimates in sieving setting for a "meet-in-the-middle" approach as in Table 3. Notation as in Table 1. Estimates marked with † correspond to standard BDD decoding.

B Case Study: NTRU LPrime

We consider NTRU LPrime [BCLvV19] as a case study of assumptions within the Hybrid Attack. This issue has recently been discussed on the "pqc-forum" associated to the NIST standardisation process [Duc19]. In the NTRUPrime Round 2 submission document [BCLvV19] an updated security evaluation is provided based on the analysis of [Wun19]. The security evaluation considers uSVP and Hybrid approaches, and does not consider dual attacks. The Hybrid Attack analysis is made using the following assumptions:

- The modified GSA for q-ary lattices [Wun19] is considered as the output basis shape of BKZ, based on the technique of reducing a sublattice of dimension $d-k$, where k is the number of "untouched" q-vectors.
- The use of the formula from [Wun19] for the success of Babai's Nearest Plane algorithm, i.e

$$p_{\mathsf{np}} \approx \prod_{1 \leq i \leq d} \left(1 - \frac{2}{B(\frac{d-1}{2}, \frac{1}{2})} \int_{\min(r_i,1)}^{1} (1-t^2)^{(d-3)/2}\, dt\right)$$

  where $r_i = \frac{\|\mathbf{b}_i^*\|}{2\|\mathbf{v}\|}$, $\|\mathbf{v}\|$ is the expected length of the target vector, i.e $\|\mathbf{v}\| = \sqrt{\sigma^2 \cdot m + \frac{n-\tau}{n} \cdot h}$, and $B(\cdot,\cdot)$ is the Beta function.
- The cost of Babai's Nearest Plane algorithm is considered to be one operation.
- In the meet-in-the-middle variant of the Hybrid Attack, the probability of collisions is one.
- In the quantum variant of the Hybrid Attack, the techniques from [GvVW17] are considered, which improves the search compared to Grover's algorithm.
- Lattice scaling is considered for the uSVP attack, but not for the hybrid attack.
- Drop-and-solve style techniques are considered in the uSVP attack.
- Memory consumption for the meet-in-the-middle step is considered.
- Core-style BKZ cost models are considered, i.e $2^{0.292\beta}$ (no lower order terms) in the sieving setting, and $2^{0.18728\beta\log(\beta)-1.019\beta+16.1}$ in the enumeration setting.

We modified the script for estimating security accompanying [BCLvV19] to provide *individual* estimates for a Hybrid Attack with "classical" guessing, and a Hybrid Attack with a meet-in-the-middle approach (as opposed to only outputting the estimate for the fastest attack) to retrieve the estimates in Table 5.

BKZ model	pre-quantum				post-quantum			
	enum		sieving		enum		sieving	
Memory cost	free	"real"	free	"real"	free	"real"	free	"real"
uSVP	364	364	**155**	**210**	187	**187**	**140**	**210**
Hybrid (classical)	307	307	194	235	219	219	183	235
Hybrid (mitm)	**222**	**275**	159	216	**170**	213	149	216

Pre-quantum enumeration corresponds to $T_{\mathsf{BKZ}}(\beta, d) = 2^{0.18728\beta\log(\beta)-1.019\beta+16.1}$, post-quantum enumeration corresponds to $T_{\mathsf{BKZ}}(\beta, d) = 2^{\frac{1}{2}(0.18728\beta\log(\beta)-1.019\beta+16.1)}$. Pre-quantum sieving corresponds to $T_{\mathsf{BKZ}}(\beta, d) = 2^{0.292\beta}$, post-quantum sieving corresponds to $T_{\mathsf{BKZ}}(\beta, d) = 2^{0.265\beta}$. In both sieving cases, *real memory* reverts the cost of lattice reduction to $T_{\mathsf{BKZ}}(\beta, d) = 2^{0.396\beta}$. In the *real memory* cases, memory requirements of the meet-in-the-middle phase (if applicable) are considered.

Table 5: Estimates from the NTRU LPrime Round 2 security script, best in class is highlighted in bold. Based on [BCLvV19, Table 2].

Bridging Assumptions

As discussed above, there are several points during a Hybrid Attack-based security analysis where assumptions are required. In order to cross-check our hybrid attack estimates, we align our code with the assumptions made in the NTRU LPrime security script. That is, we consider the set of assumptions $\mathcal{A}_0$ outlined in Table 6. Explicitly, we assume core-style BKZ models ("pre-quantum sieving" (i.e $2^{0.292\beta}$) and "pre-quantum enumeration" (i.e $2^{0.18728\beta\log(\beta)-1.019\beta+16.1}$), both with "free memory", in the language of [BCLvV19]), we assume the formula for the success probability of Babai's Nearest Plane algorithm from [Wun19] with a cost of one operation, we assume the q-ary GSA, a meet-in-the-middle guessing phase, with associated collision probability of one, we assume the target norm of the vector recovered via the BDD algorithm has Euclidean length $\sqrt{\sigma^2 \cdot m + h \cdot \frac{n-\tau}{n}}$ and we do not consider memory requirements[5], or lattice scaling.

After considering the assumption set $\mathcal{A}_0$, we move through assumptions until we reach those used in our work. In particular, assumption set $\mathcal{A}_1$ corresponds to $\mathcal{A}_0$ with the q-ary GSA swapped for the BKZ simulator, since this is a more accurate measure of the output of BKZ, assumption set $\mathcal{A}_2$ corresponds to $\mathcal{A}_1$ with the cost of Babai's Nearest Plane algorithm altered from one operation to be polynomial in the dimension of the lattice, i.e $\frac{d^2}{2^{1.06}}$ operations as in [Wun19], assumption set $\mathcal{A}_3$ corresponds to $\mathcal{A}_2$ with the core- style cost models changed to cost models which consider eight tours, and assumption set $\mathcal{A}_4$ corresponds to $\mathcal{A}_3$ with the guessing strategy changed from a meet-in-the-middle to a classical guessing strategy, thus dropping the innacurate assumption that collisions occur

[5] Note that [BCLvV19] does contain estimates which consider memory, however we do not compare against them in our work, since we do not consider memory costs.

with probability one. Finally, the only difference between assumptions set $\mathcal{A}_4$ and the assumptions considered in our work is that we consider lattice scaling.

Technique	**Assumption**	$\mathcal{A}_0$	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_4$	Our work
BKZ SVP calls	1	✓	✓	✓			
	$8d$				✓	✓	✓
p_{babai}	$\prod_{1\leq i\leq d}\left(1-\frac{2}{B(\frac{d-1}{2},\frac{1}{2})}\int_{\min(r_i,1)}^{1}(1-t^2)^{(d-3)/2}\right)$	✓	✓	✓	✓	✓	✓
T_{babai}	1	✓	✓				
	$\frac{d^2}{2^{1.06}}$			✓	✓	✓	✓
BKZ output shape	q-ary GSA	✓					
	BKZ Simulator		✓	✓	✓	✓	✓
Guessing strategy	MiTM	✓	✓	✓	✓		
	Classic					✓	✓
Target norm	$\sqrt{\sigma^2\cdot m+h\cdot\frac{n-\tau}{n}}$	✓	✓	✓	✓	✓	✓
Lattice scaling	$\mathbf{s}\mapsto\eta\mathbf{s}:\|\mathbf{s}\|\approx\|\mathbf{e}\|$						✓
MiTM probability	1	✓	✓	✓	✓	✓	✓
Memory considered?	yes						

Table 6: Sets of Assumptions Considered in this Appendix. There are many other alternative assumptions considered throughout the literature which we do not consider in this work.

We present results for each assumption set in Tables 7 and 8. To continue matching the assumptions in the NTRUPrime script, we searched for optimal values of β and τ over the sets $\tau\in\{0,40,80,\dots\}$, $\beta\in\{40,80,120,\dots\}$, we note that, in both our script and the NTRUPrime script, lower estimates can be found by performing a more granular search.

ass	alg	τ	β	η	BDD cost	$\lvert S\rvert$	repeats	d	#pp	$\log_2(\mathsf{rop})$
NTRU LPrime: $n = 761, q = 4591, \sigma = \sqrt{2/3}, h = 250$										
$\mathcal{A}_0$	NTRUPrime script	360	320	n/a	$2^{220.9}$	–	$2^{32.5}$	881	–	222.1
	our script (hybrid)	360	320	n/a	$2^{221.3}$	$2^{375.4}$	$2^{33.6}$	951	89	**222.9**
	our script (g-v decoding)	360	360	83	$2^{239.0}$	$2^{380.5}$	$2^{18.1}$	951	91	240.5
$\mathcal{A}_1$	our script (hybrid)	400	360	n/a	$2^{257.6}$	$2^{442.8}$	$2^{36.2}$	911	109	**258.7**
	our script (g-v decoding)	360	400	139	$2^{270.1}$	$2^{390.6}$	$2^{14.4}$	951	95	271.2
$\mathcal{A}_2$	our script (hybrid)	400	360	n/a	$2^{273.0}$	$2^{404.5}$	$2^{52.2}$	911	94	274.5
	our script (g-v decoding)	360	400	139	$2^{270.1}$	$2^{390.6}$	$2^{14.4}$	951	95	**271.2**
$\mathcal{A}_3$	our script (hybrid)	400	360	n/a	$2^{276.2}$	$2^{442.8}$	$2^{36.2}$	911	109	**277.9**
	our script (g-v decoding)	360	400	139	$2^{283.6}$	$2^{409.8}$	$2^{8.7}$	951	103	284.9
$\mathcal{A}_4$	our script (hybrid)	320	400	n/a	$2^{386.9}$	$2^{256.3}$	$2^{111.8}$	991	53	388.1
	our script (g-v decoding)	240	440	280	$2^{376.0}$	$2^{141.3}$	$2^{68.1}$	1071	26	**380.0**

Table 7: Enumeration-based estimates, each section corresponds to a set of assumptions outlined in Table 6, "– " denotes a value which is not compatible with our notation (for example, our script considers a simple sqrt speed-up in the search space, the NTRUPrime script considers splitting the search space as in a meet-in-the-middle approach).

ass	alg	τ	β	η	BDD cost	$\lvert S\rvert$	repeats	d	#pp	$\log_2(\mathsf{rop})$
NTRU LPrime: $n = 761, q = 4591, \sigma = \sqrt{2/3}, h = 250$										
$\mathcal{A}_0$	NTRUPrime script	240	480	n/a	$2^{156.0}$	–	$2^{18.8}$	1081	–	159.4
	our script (hybrid)	240	480	n/a	$2^{158.1}$	$2^{279.7}$	$2^{18.3}$	1111	72	**159.4**
	our script (g-v decoding)	0	560	562	$2^{164.1}$	1	1	1351	0	$165.0^{\dagger}$
$\mathcal{A}_1$	our script (hybrid)	320	600	n/a	$2^{209.9}$	$2^{348.2}$	$2^{35.8}$	1031	85	211.5
	our script (g-v decoding)	0	560	593	$2^{173.2}$	1	1	1351	0	$\mathbf{173.2}^{\dagger}$
$\mathcal{A}_2$	our script (hybrid)	360	320	n/a	$2^{221.3}$	$2^{375.4}$	$2^{33.6}$	951	89	222.9
	our script (g-v decoding)	0	560	593	$2^{173.2}$	1	1	1351	0	$\mathbf{173.2}^{\dagger}$
$\mathcal{A}_3$	our script (hybrid)	320	600	0	$2^{227.2}$	$2^{338.2}$	$2^{39.1}$	1031	81	228.3
	our script (g-v decoding)	0	560	593	$2^{176.2}$	1	1	1351	0	$\mathbf{177.8}^{\dagger}$
$\mathcal{A}_4$	our script (hybrid)	200	640	0	$2^{297.6}$	$2^{180.7}$	$2^{97.6}$	1151	40	298.6
	our script (g-v decoding)	0	560	593	$2^{176.2}$	1	1	1351	0	$\mathbf{177.8}^{\dagger}$

Table 8: Sieving Estimates. Estimates marked with † correspond to standard BDD decoding.