Cool + Cruel = Dual

Elena Kirshanova

based on joint work with A. Karenin, J. Nowakowski, E. W. Postlethwaite, and F. Virdia

Charm Workshop

Let me explain the title

- In 2024 Nolte et al. propose an attack on sparse LWE called Cool + Cruel
- In 2025 Wenger et al. claimed that the ‘Cool and Cruel’ (C+C) approach outperformed in practice established attacks on LWE such as primal attacks

Let me explain the title

- In 2024 Nolte et al. propose an attack on sparse LWE called Cool + Cruel
- In 2025 Wenger et al. claimed that the ‘Cool and Cruel’ (C+C) approach outperformed in practice established attacks on LWE such as primal attacks

We show that Cool + Cruel is a version of dual attack on LWE via generalizing this attack to the Bounded Distance Decoding problem.

We show that in practice a version of primal attack is on par in terms of time and better in terms of # LWE samples than Cool+Cruel.

`https://eprint.iacr.org/2025/1002`

Agenda

Part I. Preliminaries

Part II. Dual algorithm for BDD

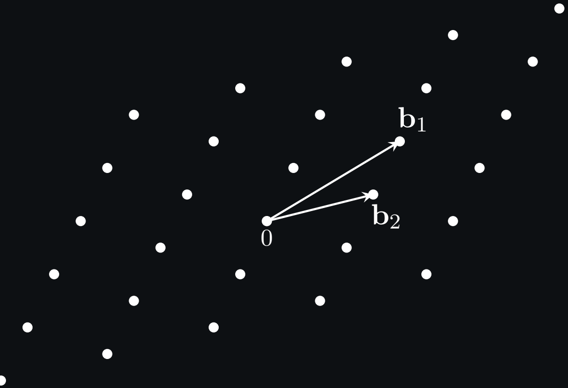
Part III. Cool+Cruel is dual

Part IV. Experiments and conclusions

Part I

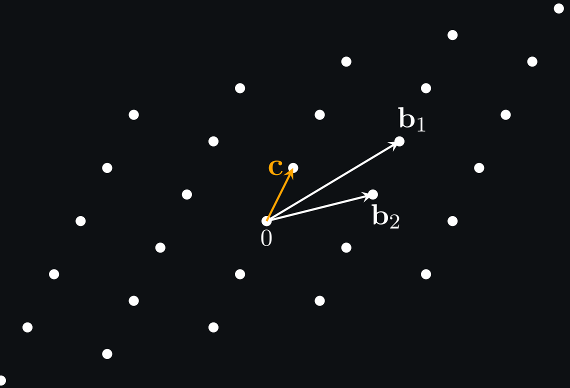
Preliminaries

Lattices: definitions



A lattice is a set $\Lambda = \{\sum_{i \leq n} x_i \mathbf{b}_i \ : \ x_i \in \mathbb{Z}\}$ for linearly independent $\mathbf{b}_i \in \mathbb{R}^n$.
$\{\mathbf{b}_i\}_i$ is a basis of Λ

Lattices: definitions

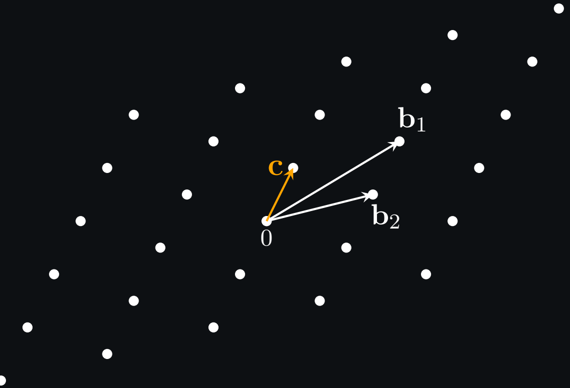


Minimum

$$\lambda_1(\Lambda) = \min_{\mathbf{v}\in\Lambda\backslash\mathbf{0}} \|\mathbf{v}\|_2$$

A lattice is a set $\Lambda = \{\sum_{i\leq n} x_i\mathbf{b}_i \ : \ x_i \in \mathbb{Z}\}$ for linearly independent $\mathbf{b}_i \in \mathbb{R}^n$.
$\{\mathbf{b}_i\}_i$ is a basis of Λ

Lattices: definitions



Minimum

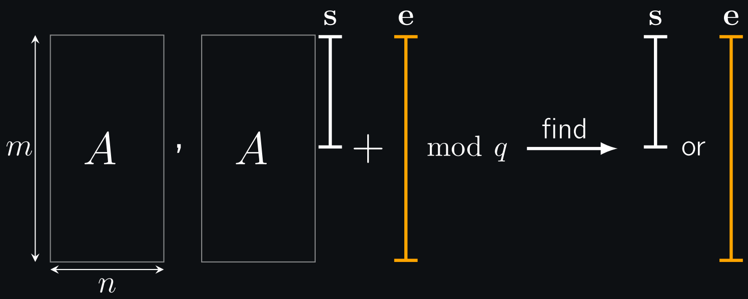
$$\lambda_1(\Lambda) = \min_{\mathbf{v}\in\Lambda\setminus\mathbf{0}} \|\mathbf{v}\|_2$$

Dual lattice

$$\Lambda^\star = \{\mathbf{x} \in \mathrm{Span}(\Lambda) : \langle \mathbf{x}, \mathbf{v}\rangle \in \mathbb{Z}\, \forall \mathbf{v} \in \Lambda\}$$

A lattice is a set $\Lambda = \{\sum_{i\le n} x_i \mathbf{b}_i \ : \ x_i \in \mathbb{Z}\}$ for linearly independent $\mathbf{b}_i \in \mathbb{R}^n$.
$\{\mathbf{b}_i\}_i$ is a basis of Λ

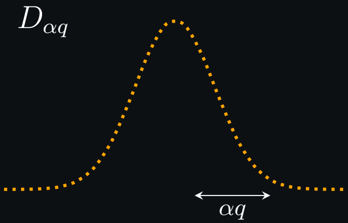
LWE (Regev'05)



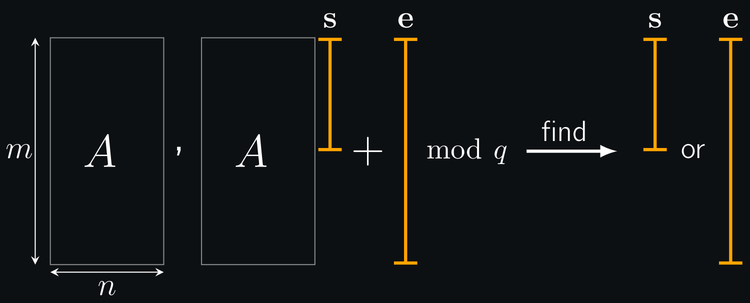
$$A \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$$

$$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^{n}$$

$$\mathbf{e} \leftarrow D_{\alpha q}^{m}$$



LWE in practice



$$A \overset{\$}{\leftarrow} \mathbb{Z}_q^{m\times n}$$

$$\mathbf{s}, \mathbf{e} \overset{\$}{\leftarrow} \mathcal{D}$$

$\mathcal{D}$ — Low entropy distr.

Examples of $\mathcal{D}$:

Central Binomial on $[-a, a]$ (Kyber, Dilithium)

Binary: $\Pr[1] = \Pr[0] = 1/2$ (FHE)

Ternary: $\Pr[1] = \Pr[-1] = \Pr[0] = 1/3$ (FHE)

Ternary with small Hamming weight (NTRU)

Bounded Distance Decoding (BDD)

Primal

$$\Lambda = \mathcal{L}(\mathbf{B})$$

Given $\mathbf{t} = \mathbf{v} + \mathbf{x}$,

where $\mathbf{v} \in \Lambda, \|\mathbf{x}\| < \frac{1}{2}\lambda_1(\Lambda)$,

find $\mathbf{v}$.

Bounded Distance Decoding (BDD)

Primal

$$\Lambda = \mathcal{L}(\mathbf{B})$$

Given $\mathbf{t} = \mathbf{v} + \mathbf{x}$,

where $\mathbf{v} \in \Lambda$, $\|\mathbf{x}\| < \frac{1}{2}\lambda_1(\Lambda)$,

find $\mathbf{v}$.

Dual

$$\Lambda^\star = \mathcal{L}(\mathbf{D}),\ \mathbf{D} = \mathbf{B}(\mathbf{B}^T \cdot B)^{-1}$$

Given $\mathbf{t}$ s.t. $\mathbf{D}^T\mathbf{t} = \mathbf{D}^T\mathbf{x} \mod 1$,

for $\|\mathbf{x}\| < \frac{1}{2}\lambda_1(\Lambda)$,

find $\mathbf{x}$.

LWE is BDD

Primal

$$\Lambda_{\mathsf{LWE}} = \{(\mathbf{y}, \mathbf{z}) \in \mathbb{Z}^m \times \mathbb{Z}^n : \mathbf{y} = -\mathbf{A}\mathbf{z} \bmod q\}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_n & \\ -\mathbf{A} & q\mathbf{I}_m \end{bmatrix}, \ \mathbf{t} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix}$$

– a BDD instance;

LWE is BDD

Primal

$$\Lambda_{\mathsf{LWE}} = \{(\mathbf{y}, \mathbf{z}) \in \mathbb{Z}^m \times \mathbb{Z}^n \ : \\ \mathbf{y} \ = \ -\mathbf{A}\mathbf{z} \bmod q\}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_n & \\ -\mathbf{A} & q\mathbf{I}_m \end{bmatrix}, \ \mathbf{t} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix}$$

– a BDD instance; Indeed,

$$\mathbf{B} \cdot \begin{bmatrix} -\mathbf{s} \\ \frac{1}{q}(\mathbf{b} - \mathbf{A}\mathbf{s} - \mathbf{e}) \end{bmatrix} = \begin{bmatrix} -\mathbf{s} \\ \mathbf{b} - \mathbf{e} \end{bmatrix} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} -\mathbf{s} \\ -\mathbf{e} \end{bmatrix}$$

LWE is BDD

Primal

$$\Lambda_{\mathsf{LWE}} = \{(\mathbf{y},\mathbf{z}) \in \mathbb{Z}^m \times \mathbb{Z}^n \ : \ \mathbf{y} \ = \ -\mathbf{A}\mathbf{z} \bmod q\}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_n & \\ -\mathbf{A} & q\mathbf{I}_m \end{bmatrix}, \ \mathbf{t} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix}$$

– a BDD instance; Indeed,

$$\mathbf{B} \cdot \begin{bmatrix} -\mathbf{s} \\ \frac{1}{q}(\mathbf{b} - \mathbf{A}\mathbf{s} - \mathbf{e}) \end{bmatrix} = \begin{bmatrix} -\mathbf{s} \\ \mathbf{b} - \mathbf{e} \end{bmatrix} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} -\mathbf{s} \\ -\mathbf{e} \end{bmatrix}$$

Dual

$$\Lambda^{\star}_{\mathsf{LWE}} = \{(\mathbf{y},\mathbf{z}) \in \tfrac{1}{q}\mathbb{Z}^m \times \tfrac{1}{q}\mathbb{Z}^n \ : \ \mathbf{y} \ = \ \mathbf{A}^T\mathbf{z} \bmod q\}$$

$$\mathbf{D} = \frac{1}{q}\begin{bmatrix} q\mathbf{I}_n & \mathbf{A}^T \\ & I_m \end{bmatrix}, \ \mathbf{t} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix}$$

– a BDD instance;

LWE is BDD

Primal

$$\Lambda_{\text{LWE}} = \{(\mathbf{y},\mathbf{z}) \in \mathbb{Z}^m \times \mathbb{Z}^n \ : \ \mathbf{y} \ = \ -\mathbf{A}\mathbf{z} \bmod \ q\}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_n & \\ -\mathbf{A} & q\mathbf{I}_m \end{bmatrix}, \ \mathbf{t} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix}$$

– a BDD instance; Indeed,

$$\mathbf{B} \cdot \begin{bmatrix} -\mathbf{s} \\ \frac{1}{q}(\mathbf{b} - \mathbf{A}\mathbf{s} - \mathbf{e}) \end{bmatrix} = \begin{bmatrix} -\mathbf{s} \\ \mathbf{b} - \mathbf{e} \end{bmatrix} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} -\mathbf{s} \\ -\mathbf{e} \end{bmatrix}$$

Dual

$$\Lambda^{\star}_{\text{LWE}} = \{(\mathbf{y},\mathbf{z}) \in \tfrac{1}{q}\mathbb{Z}^m \times \tfrac{1}{q}\mathbb{Z}^n \ : \ \mathbf{y} \ = \ \mathbf{A}^T\mathbf{z} \bmod \ q\}$$

$$\mathbf{D} = \frac{1}{q}\begin{bmatrix} q\mathbf{I}_n & \mathbf{A}^T \\ & I_m \end{bmatrix}, \ \mathbf{t} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix}$$

– a BDD instance; Indeed,

$$\mathbf{D}^T \cdot \mathbf{t} = \frac{1}{q}\begin{bmatrix} q\mathbf{I}_n & \mathbf{A}^T \\ & I_m \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix} = \frac{1}{q}\begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix}$$

$$\mathbf{D}^T \cdot \mathbf{x} = \frac{1}{q}\begin{bmatrix} q\mathbf{I}_n & \mathbf{A}^T \\ & I_m \end{bmatrix} \cdot \begin{bmatrix} -\mathbf{s} \\ -\mathbf{e} \end{bmatrix} = \frac{1}{q}\begin{bmatrix} -q\mathbf{s} \\ -\mathbf{A}\mathbf{s} - \mathbf{e} \end{bmatrix}$$

LWE is BDD

Primal

$$\Lambda_{\mathsf{LWE}} = \{(\mathbf{y}, \mathbf{z}) \in \mathbb{Z}^m \times \mathbb{Z}^n \, : \, \mathbf{y} = -\mathbf{A}\mathbf{z} \bmod q\}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_n & \\ -\mathbf{A} & q\mathbf{I}_m \end{bmatrix}, \, \mathbf{t} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix}$$

Primal attacks reduce Λ_{LWE},

or a lattice related to it.

Ex.: Kannan's Embedding

Hybrid attacks.

Dual

$$\Lambda^{\star}_{\mathsf{LWE}} = \{(\mathbf{y}, \mathbf{z}) \in \tfrac{1}{q}\mathbb{Z}^m \times \tfrac{1}{q}\mathbb{Z}^n \, : \, \mathbf{y} = \mathbf{A}^T\mathbf{z} \bmod q\}$$

$$\mathbf{D} = \frac{1}{q}\begin{bmatrix} q\mathbf{I}_n & \mathbf{A}^T \\ & \mathbf{I}_m \end{bmatrix}, \, \mathbf{t} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix}$$

Dual attacks find short vectors in $\Lambda^{\star}_{\mathsf{LWE}}$

Idea behind the dual attacks

LWE sample: $\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \mod q$

$$\Lambda^{\star}_{\mathsf{LWE}} = \{(\mathbf{y}, \mathbf{z}) \in \frac{1}{q}\mathbb{Z}^m \times \frac{1}{q}\mathbb{Z}^n \ : \ \mathbf{y} = \mathbf{A}^T\mathbf{z} \bmod \ q\}$$

Assume we have a short vector

$$\mathbf{w} \in \Lambda^{\star}_{\mathsf{LWE}} : \mathbf{w} = (\mathbf{w_1}, \mathbf{w_2}) \ : \ \mathbf{w_1} = \mathbf{A}^T\mathbf{w_2} \mod q.$$

Idea behind the dual attacks

LWE sample: $\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \mod q$

$$\Lambda^{\star}_{\text{LWE}} = \{(\mathbf{y}, \mathbf{z}) \in \frac{1}{q}\mathbb{Z}^m \times \frac{1}{q}\mathbb{Z}^n \ : \ \mathbf{y} = \mathbf{A}^T\mathbf{z} \bmod \ q\}$$

Assume we have a short vector

$$\mathbf{w} \in \Lambda^{\star}_{\text{LWE}} : \mathbf{w} = (\mathbf{w_1}, \mathbf{w_2}) \ : \ \mathbf{w_1} = \mathbf{A}^T\mathbf{w_2} \mod q.$$

Then,

$$\langle \mathbf{w_2}, \mathbf{b} \rangle = \langle \mathbf{w_2}, \mathbf{A}\mathbf{s} \rangle + \langle \mathbf{w_2}, \mathbf{e} \rangle = \langle \mathbf{A}^T\mathbf{w_2}, \mathbf{s} \rangle + \langle \mathbf{w_2}, \mathbf{e} \rangle = \langle \mathbf{w_1}, \mathbf{s} \rangle + \langle \mathbf{w_2}, \mathbf{e} \rangle - \text{short!}$$

Having many short $\mathbf{w}$'s allows to build a distinguisher for LWE!

Idea behind the dual attacks

Dual attack proceeds in two steps:

1. Reduce LWE to its decision variant
2. Solve the decision problem using many short vectors from the dual lattice

Part II

Generalizing dual attack to BDD

Decision BDD

Primal

$\Lambda = \mathcal{L}(\mathbf{B})$

Given $\mathbf{t} \in \mathrm{Span}(\Lambda)$,

decide if there exist $\mathbf{v} \in \Lambda$,

and $\mathbf{x}$ s.t. $\|\mathbf{x}\| < \frac{1}{2}\lambda_1(\Lambda)$,

and $\mathbf{t} = \mathbf{v} + \mathbf{x}$.

Dual

$\Lambda^\star = \mathcal{L}(\mathbf{D})$

Given $\mathbf{t} \in \mathrm{Span}(\Lambda)$,

decide if there exist $\mathbf{x} \in \Lambda$, s.t.

$\|\mathbf{x}\| < \frac{1}{2}\lambda_1(\Lambda)$ and

$\mathbf{D}^T\mathbf{t} = \mathbf{D}^T\mathbf{x} \bmod 1$

Dual attack on BDD

Step I. Reduce Search BDD to an easier Decision BDD

Step II. Solve Decision BDD

Dual attack on BDD

Step I. Reduce Search BDD to an easier Decision BDD

1. Sparsification technique (aka FFT)
   - Used in decision-to-search CVP reduction (see Regev's lecture notes)
   - Proposed by Guo-Johansson for dual attacks on LWE [GJ21], see also [MATZOV]
   - Generalized to BDD by Ducas-Pulles [DP23]

   Main idea: find a sparse sublattice of Λ (=dense sublattice of $\Lambda^\star$) such that $\mathbf{t}$ still gives a BDD instance.

Step II. Solve Decision BDD

Dual attack on BDD

Step I. Reduce Search BDD to an easier Decision BDD

1. Sparsification technique (aka FFT)
   - Used in decision-to-search CVP reduction (see Regev's lecture notes)
   - Proposed by Guo-Johansson for dual attacks on LWE [GJ21], see also [MATZOV]
   - Generalized to BDD by Ducas-Pulles [DP23]

   Main idea: find a sparse sublattice of Λ (=dense sublattice of $\Lambda^\star$) such that $\mathbf{t}$ still gives a BDD instance.

2. Dimension reduction (aka enumeration)
   - Used by Albrecht in his dual attack on LWE [Alb17]
   - Generalized to BDD (see next)

   Main idea: guess a part of $\mathbf{v}$ (for $\mathbf{t} = \mathbf{v} + \mathbf{x}$) using a basis of primal Λ.

Step II. Solve Decision BDD

Dual attack on BDD

Step I. Reduce Search BDD to an easier Decision BDD

1. Sparsification technique (aka FFT)
   - Used in decision-to-search CVP reduction (see Regev's lecture notes)
   - Proposed by Guo-Johansson for dual attacks on LWE [GJ21], see also [MATZOV]
   - Generalized to BDD by Ducas-Pulles [DP23]

   Main idea: find a sparse sublattice of Λ (=dense sublattice of $\Lambda^\star$) such that $\mathbf{t}$ still gives a BDD instance.

2. Dimension reduction (aka enumeration)
   - Used by Albrecht in his dual attack on LWE [Alb17]
   - Generalized to BDD (see next)

   Main idea: guess a part of $\mathbf{v}$ (for $\mathbf{t} = \mathbf{v} + \mathbf{x}$) using a basis of primal Λ.

Step II. Solve Decision BDD

Realized via computing a score function using short vectors from $\Lambda^\star$.

Solving Decision BDD (Step II)

Compute a large (exponential) set of short dual vectors $\{\mathbf{w}_1, \ldots \mathbf{w}_N\} \subset \Lambda^\star$.

YES instance

$$\mathbf{t}_Y = \mathbf{v}_Y + \mathbf{x_Y},\ \|\mathbf{x_Y}\| < \tfrac{1}{2}\lambda_1(\Lambda)$$

$\langle \mathbf{w}_i, \mathbf{t}_Y \rangle \mod 1 \sim$ Gaussian with st.dev

$$\frac{1}{\sqrt{d}}\|\mathbf{w}_i\| \cdot \|\mathbf{x_Y}\|$$

Solving Decision BDD (Step II)

Compute a large (exponential) set of short dual vectors $\{\mathbf{w}_1, \ldots \mathbf{w}_N\} \subset \Lambda^\star$.

YES instance

$$\mathbf{t}_Y = \mathbf{v}_Y + \mathbf{x_Y},\ \|\mathbf{x_Y}\| < \tfrac{1}{2}\lambda_1(\Lambda)$$

$\langle \mathbf{w}_i, \mathbf{t}_Y \rangle \mod 1 \sim$ Gaussian with st.dev

$$\frac{1}{\sqrt{d}}\|\mathbf{w}_i\| \cdot \|\mathbf{x_Y}\|$$

NO instance

$$\mathbf{t}_N = \mathbf{v}_N + \mathbf{x}_N,\ \|\mathbf{x}_N\| \geq \tfrac{1}{2}\lambda_1(\Lambda)$$

$\langle \mathbf{w}_i, \mathbf{t}_N \rangle \mod 1 \sim$ Gaussian with st.dev

$$\frac{1}{\sqrt{d}}\|\mathbf{w}_N\| \cdot \|\mathbf{x}_N\|$$

Solving Decision BDD (Step II)

Compute a large (exponential) set of short dual vectors $\{\mathbf{w}_1, \dots \mathbf{w}_N\} \subset \Lambda^\star$.

YES instance

$$\mathbf{t}_Y = \mathbf{v}_Y + \mathbf{x_Y},\ \|\mathbf{x_Y}\| < \tfrac{1}{2}\lambda_1(\Lambda)$$

$\langle \mathbf{w}_i, \mathbf{t}_Y \rangle \mod 1 \sim$ Gaussian with st.dev

$$\frac{1}{\sqrt{d}}\|\mathbf{w}_i\| \cdot \|\mathbf{x_Y}\|$$

NO instance

$$\mathbf{t}_N = \mathbf{v}_N + \mathbf{x}_N,\ \|\mathbf{x}_N\| \geq \tfrac{1}{2}\lambda_1(\Lambda)$$

$\langle \mathbf{w}_i, \mathbf{t}_N \rangle \mod 1 \sim$ Gaussian with st.dev

$$\frac{1}{\sqrt{d}}\|\mathbf{w}_N\| \cdot \|\mathbf{x}_N\|$$

For small enough $\|\mathbf{x}_Y\|$ and large enough N, the two distributions $\{\langle \mathbf{w}_i, \mathbf{t}_Y \rangle\}$ and $\{\langle \mathbf{w}_i, \mathbf{t}_N \rangle\}$ can be distinguished: $\langle \mathbf{w}_i, \mathbf{t}_Y \rangle \mod 1$ is more concentrated around 0.

Dimension reduction for BDD (Step I)

$$\mathbf{t} = \mathbf{B}\mathbf{u} + \mathbf{x} \quad \text{for some } \mathbf{u} \in \mathbb{Z}^d$$

$$\mathbf{t} = \mathbf{B}_0\mathbf{u}_0 + \mathbf{B}_1\mathbf{u}_1 + \mathbf{x} \text{ for } \mathbf{B} = [\mathbf{B}_0, \mathbf{B}_1], \mathbf{u} = [\mathbf{u}_0, \mathbf{u}_1]$$

Dimension reduction for BDD (Step I)

$$\mathbf{t} = \mathbf{B}\mathbf{u} + \mathbf{x} \quad \text{for some } \mathbf{u} \in \mathbb{Z}^d$$
$$\mathbf{t} = \mathbf{B}_0\mathbf{u}_0 + \mathbf{B}_1\mathbf{u}_1 + \mathbf{x} \text{ for } \mathbf{B} = [\mathbf{B}_0, \mathbf{B}_1], \mathbf{u} = [\mathbf{u}_0, \mathbf{u}_1]$$

Consider two projections:

$$\pi_{\mathbf{B}_0} := \pi_{\mathrm{Span}(\mathbf{B}_0)} - \text{ onto } \mathrm{Span}(\mathbf{B}_0)$$
$$\pi^{\perp}_{\mathbf{B}_0} := \pi^{\perp}_{\mathrm{Span}(\mathbf{B}_0)} - \text{ project orthogonal to } \mathrm{Span}(\mathbf{B}_0)$$

Dimension reduction for BDD (Step I)

$$\mathbf{t} = \mathbf{B}\mathbf{u} + \mathbf{x} \quad \text{for some } \mathbf{u} \in \mathbb{Z}^d$$

$$\mathbf{t} = \mathbf{B}_0\mathbf{u}_0 + \mathbf{B}_1\mathbf{u}_1 + \mathbf{x} \text{ for } \mathbf{B} = [\mathbf{B}_0, \mathbf{B}_1], \mathbf{u} = [\mathbf{u}_0, \mathbf{u}_1]$$

Consider two projections:

$$\pi_{\mathbf{B}_0} := \pi_{\mathrm{Span}(\mathbf{B}_0)} - \text{ onto } \mathrm{Span}(\mathbf{B}_0)$$

$$\pi^{\perp}_{\mathbf{B}_0} := \pi^{\perp}_{\mathrm{Span}(\mathbf{B}_0)} - \text{ project orthogonal to } \mathrm{Span}(\mathbf{B}_0)$$

Apply $\pi_{\mathrm{Span}(\mathbf{B}_0)}, \pi^{\perp}_{\mathrm{Span}(\mathbf{B}_0)}$ to $\mathbf{t}$:

$$\begin{cases} \pi_{\mathbf{B}_0}(\mathbf{t}) = \mathbf{B}_0\mathbf{u}_0 + \pi_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1 + \pi_{\mathbf{B}_0}(\mathbf{x}) \\ \pi^{\perp}_{\mathbf{B}_0}(\mathbf{t}) = \pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1 + \pi^{\perp}_{\mathbf{B}_0}(\mathbf{x}) \end{cases}$$

Dimension reduction for BDD (Step I)

$$\mathbf{t} = \mathbf{B}\mathbf{u} + \mathbf{x} \quad \text{for some } \mathbf{u} \in \mathbb{Z}^d$$

$$\mathbf{t} = \mathbf{B}_0\mathbf{u}_0 + \mathbf{B}_1\mathbf{u}_1 + \mathbf{x} \text{ for } \mathbf{B} = [\mathbf{B}_0, \mathbf{B}_1], \mathbf{u} = [\mathbf{u}_0, \mathbf{u}_1]$$

Consider two projections:

$$\pi_{\mathbf{B}_0} := \pi_{\mathrm{Span}(\mathbf{B}_0)} - \text{ onto } \mathrm{Span}(\mathbf{B}_0)$$

$$\pi_{\mathbf{B}_0}^{\perp} := \pi_{\mathrm{Span}(\mathbf{B}_0)}^{\perp} - \text{ project orthogonal to } \mathrm{Span}(\mathbf{B}_0)$$

Apply $\pi_{\mathrm{Span}(\mathbf{B}_0)}, \pi_{\mathrm{Span}(\mathbf{B}_0)}^{\perp}$ to $\mathbf{t}$:

$$\begin{cases} \pi_{\mathbf{B}_0}(\mathbf{t}) = \mathbf{B}_0\mathbf{u}_0 + \pi_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1 + \pi_{\mathbf{B}_0}(\mathbf{x}) \\ \pi_{\mathbf{B}_0}^{\perp}(\mathbf{t}) = \pi_{\mathbf{B}_0}^{\perp}(\mathbf{B}_1)\mathbf{u}_1 + \pi_{\mathbf{B}_0}^{\perp}(\mathbf{x}) \end{cases} \iff \begin{cases} \pi_{\mathbf{B}_0}(\mathbf{t} - \mathbf{B}_1\mathbf{u}_1) = \mathbf{B}_0\mathbf{u}_0 + \pi_{\mathbf{B}_0}(\mathbf{x}) \\ \pi_{\mathbf{B}_0}^{\perp}(\mathbf{t}) = \pi_{\mathbf{B}_0}^{\perp}(\mathbf{B}_1)\mathbf{u}_1 + \pi_{\mathbf{B}_0}^{\perp}(\mathbf{x}) \end{cases}$$

Dimension reduction for BDD (Step I)

$$\mathbf{t} = \mathbf{B}\mathbf{u} + \mathbf{x} \quad \text{for some } \mathbf{u} \in \mathbb{Z}^d$$

$$\mathbf{t} = \mathbf{B}_0\mathbf{u}_0 + \mathbf{B}_1\mathbf{u}_1 + \mathbf{x} \text{ for } \mathbf{B} = [\mathbf{B}_0, \mathbf{B}_1], \mathbf{u} = [\mathbf{u}_0, \mathbf{u}_1]$$

Consider two projections:

$$\pi_{\mathbf{B}_0} := \pi_{\mathrm{Span}(\mathbf{B}_0)} - \text{ onto } \mathrm{Span}(\mathbf{B}_0)$$

$$\pi^{\perp}_{\mathbf{B}_0} := \pi^{\perp}_{\mathrm{Span}(\mathbf{B}_0)} - \text{ project orthogonal to } \mathrm{Span}(\mathbf{B}_0)$$

Apply $\pi_{\mathrm{Span}(\mathbf{B}_0)}, \pi^{\perp}_{\mathrm{Span}(\mathbf{B}_0)}$ to $\mathbf{t}$:

BDD on $\pi_{\mathbf{B}_0}(\mathbf{B})$!

$$\begin{cases} \pi_{\mathbf{B}_0}(\mathbf{t}) = \mathbf{B}_0\mathbf{u}_0 + \pi_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1 + \pi_{\mathbf{B}_0}(\mathbf{x}) \\ \pi^{\perp}_{\mathbf{B}_0}(\mathbf{t}) = \pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1 + \pi^{\perp}_{\mathbf{B}_0}(\mathbf{x}) \end{cases} \iff \begin{cases} \pi_{\mathbf{B}_0}(\mathbf{t} - \mathbf{B}_1\mathbf{u}_1) = \mathbf{B}_0\mathbf{u}_0 + \pi_{\mathbf{B}_0}(\mathbf{x}) \\ \pi^{\perp}_{\mathbf{B}_0}(\mathbf{t}) = \pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1 + \pi^{\perp}_{\mathbf{B}_0}(\mathbf{x}) \end{cases}$$

Dimension reduction for BDD (Step I)

$$\pi_{\mathbf{B}_0}(\mathbf{t} - \mathbf{B}_1\mathbf{u}_1) = \mathbf{B}_0\mathbf{u}_0 + \pi_{\mathbf{B}_0}(\mathbf{x}) \text{ – BDD on } \pi_{\mathbf{B}_0}(\mathbf{B}) \tag{1}$$

$$\pi^{\perp}_{\mathbf{B}_0}(\mathbf{t}) = \pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1 + \pi^{\perp}_{\mathbf{B}_0}(\mathbf{x}), \text{ where } \|\pi^{\perp}_{\mathbf{B}_0}(\mathbf{x})\| \approx \sqrt{k/d}\|\mathbf{x}\| \tag{2}$$

BDD Solver:

1. Enumerate all $\pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1$ that lie within $\sqrt{\frac{k}{d}}\|\mathbf{x}\|$ from $\pi^{\perp}_{\mathbf{B}_0}(\mathbf{t})$ (use e.g. [DucasLectureNotes]) using Eq(2)

Dimension reduction for BDD (Step I)

$$\pi_{\mathbf{B}_0}(\mathbf{t} - \mathbf{B}_1\mathbf{u}_1) = \mathbf{B}_0\mathbf{u}_0 + \pi_{\mathbf{B}_0}(\mathbf{x}) \text{ – BDD on } \pi_{\mathbf{B}_0}(\mathbf{B}) \tag{1}$$

$$\pi^{\perp}_{\mathbf{B}_0}(\mathbf{t}) = \pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1 + \pi^{\perp}_{\mathbf{B}_0}(\mathbf{x}), \text{ where } \|\pi^{\perp}_{\mathbf{B}_0}(\mathbf{x})\| \approx \sqrt{k/d}\|\mathbf{x}\| \tag{2}$$

BDD Solver:

1. Enumerate all $\pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1$ that lie within $\sqrt{\frac{k}{d}}\|\mathbf{x}\|$ from $\pi^{\perp}_{\mathbf{B}_0}(\mathbf{t})$ (use e.g. [DucasLectureNotes]) using Eq(2)
2. Identify the correct $\mathbf{u}_1$ by solving decision BDD

Dimension reduction for BDD (Step I)

$$\pi_{\mathbf{B}_0}(\mathbf{t} - \mathbf{B}_1\mathbf{u}_1) = \mathbf{B}_0\mathbf{u}_0 + \pi_{\mathbf{B}_0}(\mathbf{x}) \text{ – BDD on } \pi_{\mathbf{B}_0}(\mathbf{B}) \tag{1}$$

$$\pi^{\perp}_{\mathbf{B}_0}(\mathbf{t}) = \pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1 + \pi^{\perp}_{\mathbf{B}_0}(\mathbf{x}), \text{ where } \|\pi^{\perp}_{\mathbf{B}_0}(\mathbf{x})\| \approx \sqrt{k/d}\|\mathbf{x}\| \tag{2}$$

BDD Solver:

1. Enumerate all $\pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)\mathbf{u}_1$ that lie within $\sqrt{\frac{k}{d}}\|\mathbf{x}\|$ from $\pi^{\perp}_{\mathbf{B}_0}(\mathbf{t})$ (use e.g. [DucasLectureNotes]) using Eq(2)
2. Identify the correct $\mathbf{u}_1$ by solving decision BDD
3. For the correct $\mathbf{u}_1$ solve search BDD on $\mathcal{L}(\mathbf{B}_0)$ with $\mathbf{t} = \pi_{\mathbf{B}_0}(\mathbf{t} - \mathbf{B}_1\mathbf{u}_1)$ (use e.g. a CVP solver or run primal attack)

Dimension reduction for LWE

The previous algorithm can be easily specialized to LWE. Recall,

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_n & \\ -\mathbf{A} & q\mathbf{I}_m \end{bmatrix} \quad \mathbf{D} = \frac{1}{q} \begin{bmatrix} q\mathbf{I}_n & \mathbf{A}^T \\ & I_m \end{bmatrix}$$

Fact. For all k, $(\mathbf{d}_0, \ldots, \mathbf{d}_{k-1})$ generate a lattice dual to $\pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)$.

From the shapes of $\mathbf{B}, \mathbf{D}$ and the above fact:

$$\pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1) = \mathcal{L}(\mathbf{D}_{[0,k)})^{\star} = \mathcal{L}([\mathbf{I}_k, 0^{d-k}])^{\star} = \mathbb{Z}^k \times \{0\}^{d-k}.$$

Dimension reduction for LWE

The previous algorithm can be easily specialized to LWE. Recall,

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_n & \\ -\mathbf{A} & q\mathbf{I}_m \end{bmatrix} \quad \mathbf{D} = \frac{1}{q} \begin{bmatrix} q\mathbf{I}_n & \mathbf{A}^T \\ & I_m \end{bmatrix}$$

Fact. For all k, $(\mathbf{d}_0, \ldots, \mathbf{d}_{k-1})$ generate a lattice dual to $\pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)$.

From the shapes of $\mathbf{B}, \mathbf{D}$ and the above fact:

$$\pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1) = \mathcal{L}(\mathbf{D}_{[0,k)})^{\star} = \mathcal{L}([\mathbf{I}_k, 0^{d-k}])^{\star} = \mathbb{Z}^k \times \{0\}^{d-k}.$$

Therefore,

$$\mathbf{t} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} -\mathbf{s} \\ \mathbf{b} - \mathbf{e} \end{bmatrix} + \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix} \xrightarrow{\pi_{\mathbb{Z}^k \times \{\mathbf{0}\}}} \begin{bmatrix} -\mathbf{s}_{[\mathbf{0},\mathbf{k}]} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{s}_{[\mathbf{0},\mathbf{k}]} \\ \mathbf{0} \end{bmatrix}$$

Enumeration for LWE = Guessing the partial secret!

Dimension reduction for LWE

The previous algorithm can be easily specialized to LWE. Recall,

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_n & \\ -\mathbf{A} & q\mathbf{I}_m \end{bmatrix} \quad \mathbf{D} = \frac{1}{q} \begin{bmatrix} q\mathbf{I}_n & \mathbf{A}^T \\ & I_m \end{bmatrix}$$

Fact. For all k, $(\mathbf{d}_0, \ldots, \mathbf{d}_{k-1})$ generate a lattice dual to $\pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1)$.

From the shapes of $\mathbf{B}, \mathbf{D}$ and the above fact:

$$\pi^{\perp}_{\mathbf{B}_0}(\mathbf{B}_1) = \mathcal{L}(\mathbf{D}_{[0,k)})^\star = \mathcal{L}([\mathbf{I}_k, 0^{d-k}])^\star = \mathbb{Z}^k \times \{0\}^{d-k}.$$

Therefore,

$$\mathbf{t} = \begin{bmatrix} 0^n \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} -\mathbf{s} \\ \mathbf{b} - \mathbf{e} \end{bmatrix} + \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix} \xrightarrow{\pi_{\mathbb{Z}^k \times \{\mathbf{0}\}}} \begin{bmatrix} -\mathbf{s}_{[\mathbf{0},\mathbf{k}]} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{s}_{[\mathbf{0},\mathbf{k}]} \\ \mathbf{0} \end{bmatrix}$$

Enumeration for LWE = Guessing the partial secret!

Thus we recover the dual attack by Albrecht [Alb17] (up to coordinate permutation and scaling).

How to choose k?

1. Choose k such enumeration + Decision BBD balance with the time to find many small dual vectors (as done in [Alb17])

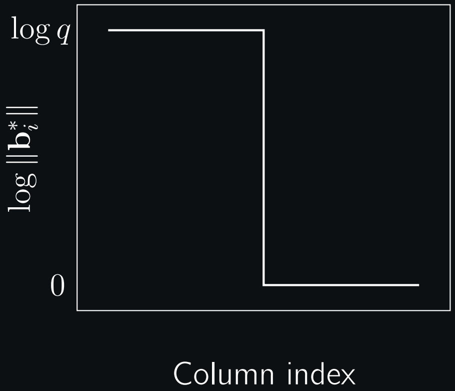
2. Use the Z-shape of reduced dual basis (as done in Cool + Cruel)

Part III

Cool+Cruel as a special case of the dual attack on LWE/BDD

Z-shape of LWE dual ([How07])

$$\mathbf{D}^{\mathsf{CC}} = \begin{bmatrix} & \mathbf{I}_m \\ q\mathbf{I}_n & \mathbf{A}^T \end{bmatrix}$$

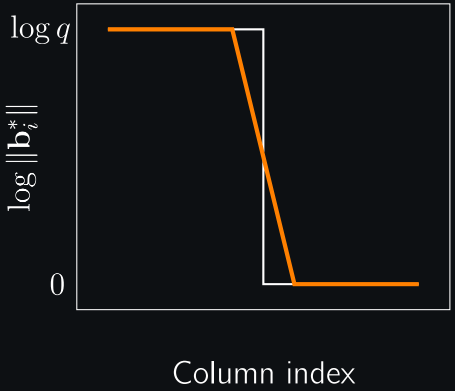


Z-shape of LWE dual ([How07])

$$\mathbf{D}^{\mathsf{CC}} = \begin{bmatrix} & \mathbf{I}_m \\ q\mathbf{I}_n & \mathbf{A}^T \end{bmatrix}$$

$$\Big\downarrow \ \mathsf{BKZ}$$

$$\mathbf{D}^{\mathsf{bkz}} = \begin{bmatrix} \mathbf{0} & \mathbf{D}_0 \\ q\mathbf{I}_k & \mathbf{D}_1 \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix}$$



$$\mathbf{D}^{\mathsf{bkz}} = \mathbf{D}^{\mathsf{CC}} \cdot \mathbf{U} \quad \mathbf{U} - \text{unimodular}$$

Z-shape of LWE dual ([How07])

Effectively BKZ algorithm considers only the last $d - k$ columns of $\mathbf{D}^{\mathsf{CC}}$

$$\mathbf{D}^{\mathsf{CC}} = \begin{bmatrix} & \mathbf{I}_m \\ q\mathbf{I}_n & \mathbf{A}^T \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I}_m \\ q\mathbf{I}_k & \mathbf{0} & \mathbf{A}_0^T \\ \mathbf{0} & q\mathbf{I}_{n-k} & \mathbf{A}_1^T \end{bmatrix}$$

Since BKZ works on projected sublattices, means that BKZ reduces

$$\pi^{\perp}_{\mathbf{0}\times q\mathbf{I}_k\times\mathbf{0}}(\mathbf{D}^{\mathsf{CC}}) = \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \\ \mathbf{0} & \mathbf{0} \\ q\mathbf{I}_{n-k} & \mathbf{A}_1^T \end{bmatrix} \xrightarrow{\mathsf{BKZ}} \begin{bmatrix} \mathbf{D}_0 \\ \mathbf{0} \\ \mathbf{D}_2 \end{bmatrix} = \pi^{\perp}_{\mathbf{0}\times q\mathbf{I}_k\times\mathbf{0}} \begin{bmatrix} \mathbf{0} & \mathbf{D}_0 \\ q\mathbf{I}_k & \mathbf{D}_1 \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix}$$

Conclusion: $\mathbf{D}_0, \mathbf{D}_1$ are small, $\mathbf{D}_1$ is not.

Cool + Cruel

$$\mathbf{D}^{\mathsf{bkz}} = \begin{bmatrix} \begin{matrix} \mathbf{0} & \mathbf{D}_0 \end{matrix} \\ \boxed{\begin{matrix} q\mathbf{I}_k & \mathbf{D}_1 \\ \mathbf{0} & \mathbf{D}_2 \end{matrix}} \end{bmatrix} = \mathbf{D}^{\mathsf{CC}} \cdot \mathbf{U} = \begin{bmatrix} & \mathbf{I}_m \\ q\mathbf{I}_n & \mathbf{A}^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \end{bmatrix} \equiv \begin{bmatrix} \mathbf{U}_1 \\ \boxed{\mathbf{A}^T \mathbf{U}_1} \end{bmatrix} \bmod q$$

Cool + Cruel

$$\mathbf{D}^{\mathsf{bkz}} = \begin{bmatrix} \begin{matrix} \mathbf{0} & \mathbf{D}_0 \end{matrix} \\ \boxed{\begin{matrix} q\mathbf{I}_k & \mathbf{D}_1 \\ \mathbf{0} & \mathbf{D}_2 \end{matrix}} \end{bmatrix} = \mathbf{D}^{\mathsf{CC}} \cdot \mathbf{U} = \begin{bmatrix} & \mathbf{I}_m \\ q\mathbf{I}_n & \mathbf{A}^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \end{bmatrix} \equiv \begin{bmatrix} \mathbf{U}_1 \\ \boxed{\mathbf{A}^T\mathbf{U}_1} \end{bmatrix} \bmod q$$

It means that $\mathbf{A}^{\mathsf{red}} := \mathbf{U}_1 \cdot \mathbf{A} \bmod q$ follows Z-shape form!

$$\mathbf{A}^{\mathsf{bkz}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{D}_1^T & \mathbf{D}_2^T \end{bmatrix}$$

Large “Cruel”

Small “Cool”

Cool + Cruel

$$\mathbf{A}^{\mathsf{bkz}} = \begin{bmatrix} \overbrace{\mathbf{0}}^{k} & \mathbf{0} \\ \mathbf{D}_1^T & \mathbf{D}_2^T \end{bmatrix}$$

Large “Cruel”

Small “Cool”

Algorithm:

1. Guess $\mathbf{s}_0 \leftarrow \mathcal{D}^k$ (LWE secret $\mathbf{s} = [\mathbf{s}_0, \mathbf{s}_1]$)
2. Compute

$$\mathbf{U}_1^T \cdot \mathbf{b} - \begin{bmatrix} \mathbf{0} \\ \mathbf{D}_1^T \end{bmatrix} \cdot \mathbf{s}_0 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{D}_1^T & \mathbf{D}_2^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \end{bmatrix} + \mathbf{U}_1^T \mathbf{e} - \begin{bmatrix} \mathbf{0} \\ \mathbf{D}_1^T \end{bmatrix} \cdot \mathbf{s}_0 = \overbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{D}_2^T \mathbf{s}_1 \end{bmatrix} + \mathbf{U}_1^T \mathbf{e}}^{\text{small}}$$

3. Recover $\mathbf{s}_1$ using some statistical test

Part IV

In practice

Solving Sparse LWE in practice

- Cool+Crue reports on efficient recovery of LWE in relatively high dimensions for extremely sparse LWE (e.g. Hamming weight 11 for ternary secret)
- We show that folklore drop-and-solve strategy is not worse

  $$\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \overset{!}{=} \mathbf{A}'\mathbf{s}' + \mathbf{e},$$

  where $\mathbf{A}'$ consists of columns of A on which $\mathbf{s}$ (and $\mathbf{s}'$) are non-zero.

Conclusions

- Nolte Cool+Crue attack is a re-phrasing of dual attack
- In practice, embarrassingly simple drop-and-solve works no worse
- Open question: concrete complexity of dual/primal for sparse LWE.

References

- [Alb17] M. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL.
- [DP23] L. Ducas, L. N. Pulles. oes the dual-sieve attack on learning with errors even work?
- [GJ21] Q. Guo and T. Johansson. Faster dual lattice attacks for solving LWE with applications to CRYSTALS.
- [MATZOV] DF MATZOV. Report on the security of lwe: improved dual lattice attack, 2022
- [NMW] N. Nolte, M. Malhou, E. Wenger, S. Stevens, C. Yuanchen Li, F. Charton, and K. E. Lauter. The cool and the cruel: Separating hard parts of LWE secrets.
- [WSM] E. Wenger, E. Saxena, M. Malhou, E. Thieu, and K. Lauter. Benchmarking Attacks on Learning with Errors