Some Remarks on Small Secret LWE

Another Look at HELib's Choices of Parameters

Martin R. Albrecht

07/05/2016

Outline

Introduction

Base Line

Swapping Error and Secret

Modulus Switching

Sparse Secrets

Results

Introduction

Learning with Errors

The Learning with Errors (LWE) problem was defined by Oded Regev.[1]

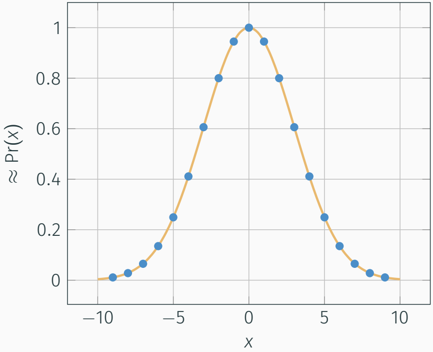
Given $(\mathbf{A}, \mathbf{c})$ with $\mathbf{c} \in \mathbb{Z}_q^m$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ and small $\mathbf{e} \in \mathbb{Z}^m$ is

$$\begin{pmatrix} \\ \\ \mathbf{c} \\ \\ \\ \end{pmatrix} = \begin{pmatrix} \leftarrow & n & \rightarrow \\ & & \\ & \mathbf{A} & \\ & & \\ & & \end{pmatrix} \cdot \begin{pmatrix} \\ \mathbf{s} \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \mathbf{e} \\ \\ \\ \end{pmatrix}$$

or $\mathbf{c} \leftarrow_{\$} \mathcal{U}\left(\mathbb{Z}_q^m\right)$.

[1] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In: *37th ACM STOC*. ed. by Harold N. Gabow and Ronald Fagin. ACM Press, May 2005, pp. 84–93.

Parameters



- Parameters are:
  - dimension n,
  - modulus q (e.g. $q \approx n^2$),
  - noise size α (e.g. $\alpha q \approx \sqrt{n}$),
  - number of samples m.
- Elements of $\mathbf{A}, \mathbf{s}, \mathbf{e}, \mathbf{c}$ are in $\mathbb{Z}_q$.
- $\mathbf{e}$ is sampled from χ_α, a discrete Gaussian with width

$$\sigma = \frac{\alpha q}{\sqrt{2\pi}}.$$

FHE-schemes based on LWE

BGV Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In: *ITCS 2012*. Ed. by Shafi Goldwasser. ACM, Jan. 2012, pp. 309–325

FV Junfeng Fan and Frederik Vercauteren. Somewhat Practical Fully Homomorphic Encryption. Cryptology ePrint Archive, Report 2012/144. `http://eprint.iacr.org/2012/144`. 2012

LTV Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: *44th ACM STOC*. ed. by Howard J. Karloff and Toniann Pitassi. ACM Press, May 2012, pp. 1219–1234 [2]

YASHE Joppe W. Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig. Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme. In: *14th IMA International Conference on Cryptography and Coding*. Ed. by Martijn Stam. Vol. 8308. LNCS. Springer, Heidelberg, Dec. 2013, pp. 45–64. DOI: `10.1007/978-3-642-45239-0_4`

[2] See Léo's talk for attacks on LTV and YASHE exploiting that they are not quite LWE.

FHE-schemes based on LWE (cont.)

GSW Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In: *CRYPTO 2013, Part I*. ed. by Ran Canetti and Juan A. Garay. Vol. 8042. LNCS. Springer, Heidelberg, Aug. 2013, pp. 75–92. DOI: `10.1007/978-3-642-40041-4_5`

AGCD Jung Hee Cheon and Damien Stehlé. Fully Homomorphic Encryption over the Integers Revisited. In: *EUROCRYPT 2015, Part I*. ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. LNCS. Springer, Heidelberg, Apr. 2015, pp. 513–536. DOI: `10.1007/978-3-662-46800-5_20`

Small Secrets

- FHE schemes based on LWE typically choose very small secrets.
- For example, $\mathbf{s}_i \leftarrow \{-1, 0, 1\}$ or $\mathbf{s}_\mathrm{i} \leftarrow \{0, 1\}$.
- `HElib`[3] typically chooses $\mathbf{s}$ such that $w = 64$ entries are ± 1 and all remaining entries are 0, regardless of dimension n.
- The same strategy is used in a recent comparison study.[4]

How many bits of security does this cost?

[3] Shai Halevi and Victor Shoup. Algorithms in HElib. In: *CRYPTO 2014, Part I*. ed. by Juan A. Garay and Rosario Gennaro. Vol. 8616. LNCS. Springer, Heidelberg, Aug. 2014, pp. 554–571. DOI: 10.1007/978-3-662-44371-2_31.

[4] Ana Costache and Nigel P. Smart. Which Ring Based Somewhat Homomorphic Encryption Scheme is Best? In: *CT-RSA 2016*. Ed. by Kazue Sako. Vol. 9610. LNCS. Springer, Heidelberg, 2016, pp. 325–340. DOI: 10.1007/978-3-319-29485-8_19.

Binary LWE Secret Distributions

$\mathcal{B}^{+}$ each component is independently sampled uniformly from $\{0, 1\}$.

$\mathcal{B}^{-}$ each component is independently sampled uniformly from $\{-1, 0, 1\}$.

$\mathcal{B}^{\pm}_{h}$ like above but with guarantee that h components are non-zero.

Hardness: LWE Normal Form

Given samples

$$(\mathbf{a}, c) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

with $\mathbf{a} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, $e \leftarrow D_{\alpha q,0}$ and $\mathbf{s} \in \mathbb{Z}_q^n$, we can construct samples

$$(\mathbf{a}, c) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{e} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

with $\mathbf{a} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, $e \leftarrow D_{\alpha q,0}$ and $\mathbf{e}$ such that all components

$$e_i \leftarrow D_{\alpha q,0}$$

in polynomial time.[5]

[5] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Heidelberg, Aug. 2009, pp. 595–618.

Hardness: Reductions

"A major part of our reduction [...] is therefore dedicated to showing reduction from LWE (in dimension n) with arbitrary secret in $\mathbb{Z}_q^n$ to LWE (in dimension $n \log_2 q$) with a secret chosen uniformly over $\{0, 1\}$."[6]

[6] Zvika Brakerski et al. Classical hardness of learning with errors. In: *45th ACM STOC*. ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM Press, June 2013, pp. 575–584.

Hardness: Algorithms

"[This work] suggests that this is overkill and that even $n \log \log n$ may be more than sufficient."[7]

[7] Shi Bai and Steven D. Galbraith. Lattice Decoding Attacks on Binary LWE. . In: *ACISP 14*. Ed. by Willy Susilo and Yi Mu. Vol. 8544. LNCS. Springer, Heidelberg, July 2014, pp. 322–337. DOI: 10.1007/978-3-319-08344-5_21.

Hardness: Constructions

"This brings up the question of whether one can get better attacks against LWE instances with a very sparse secret (much smaller than even the noise). [...] it seems that the very sparse secret should only add maybe one bit to the modulus/noise ratio."[8]

[8] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic Evaluation of the AES Circuit. Cryptology ePrint Archive, Report 2012/099. `http://eprint.iacr.org/2012/099`. 2012.

Base Line

Dual Attack

Short Integer Solutions (SIS)

Given $q \in \mathbb{Z}$, a matrix $\mathbf{A}$, and $t < q$; find $\mathbf{y}$ with $0 < \|\mathbf{y}\| \leq t$ and

$$\mathbf{y} \cdot \mathbf{A} \equiv \mathbf{0} \pmod{q}.$$

- Find a short $\mathbf{y}$ solving SIS on $\mathbf{A}$.
- Given LWE samples $\mathbf{A}, \mathbf{c}$ where $\mathbf{c} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}$ or $\mathbf{c}$ uniform.
- Compute $\langle \mathbf{y}, \mathbf{c} \rangle$.
  - If $\mathbf{c} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}$, then $\langle \mathbf{y}, \mathbf{c} \rangle = \langle \mathbf{y} \cdot \mathbf{A}, \mathbf{s} \rangle + \langle \mathbf{y}, \mathbf{e} \rangle \equiv \langle \mathbf{y}, \mathbf{e} \rangle \pmod{q}$.
  - If $\mathbf{c}$ is uniformly random, so is $\langle \mathbf{y}, \mathbf{c} \rangle$.
- If $\mathbf{y}$ is short then $\langle \mathbf{y}, \mathbf{e} \rangle$ is also short.

Dual Attack

Lemma

Given an LWE instance characterised by n, α, q and a vector $\mathbf{v}$ *of length* $\|\mathbf{v}\|$ *such that* $\mathbf{v} \cdot \mathbf{A} \equiv 0 \pmod{q}$*, the advantage of distinguishing* $\langle \mathbf{v}, \mathbf{e} \rangle$ *from random is close to*[9]

$$\exp(-\pi(\|\mathbf{v}\| \cdot \alpha)^2).$$

[9] Richard Lindner and Chris Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. In: *CT-RSA 2011.* Ed. by Aggelos Kiayias. Vol. 6558. LNCS. Springer, Heidelberg, Feb. 2011, pp. 319–339.

Dual Attack

A reduced lattice basis contains short vectors. In particular, the first vector is short: $\|\mathbf{v}\| \approx \delta_0^m \, q^{n/m}$.

1. Construct a basis of the dual lattice from $\mathbf{A}$.
2. Run lattice reduction algorithm to obtain short vectors $\mathbf{v}_i$.
3. Check if $\mathbf{v}_i \cdot \mathbf{A}$ are small.[10]

[10] Daniele Micciancio and Oded Regev. Lattice-based Cryptography. In: *Post-Quantum Cryptography*. Ed. by Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Berlin, Heidelberg, New York: Springer, 2009, pp. 147–191.

Dual Attack

A reduced lattice basis contains short vectors. In particular, the first vector is short: $\|\mathbf{v}\| \approx \delta_0^m \, q^{n/m}$.

1. Construct a basis of the dual lattice from $\mathbf{A}$.
2. Run lattice reduction algorithm to obtain short vectors $\mathbf{v}_i$.
3. Check if $\mathbf{v}_i \cdot \mathbf{A}$ are small.[10]

Cost

How expensive is it to achieve the target quality?

[10] Daniele Micciancio and Oded Regev. Lattice-based Cryptography. In: *Post-Quantum Cryptography*. Ed. by Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Berlin, Heidelberg, New York: Springer, 2009, pp. 147–191.

HELib

```
long FindM(long k, long L, long c, long p, long d, long s,
           long chosen_m, bool verbose) {
  // get a lower-bound on the parameter N=phi(m):
  …
  // 6. To get k-bit security we need N>log(Q0/sigma)(k+110)/7.2, i.e.
  //    roughly N > (L+1)*pSize*(1+1/c)(k+110) / 7.2

  // Compute a bound on m, and make sure that it is not too large
  double cc = 1.0+(1.0/(double)c);
  double dN = ceil((L+1)*FHE_pSize*cc*(k+110)/7.2);
  …
  return m;
}
```

Lindner-Peikert Estimates

Lindner and Peikert[11] give an estimate for the runtime (in seconds) of BKZ as

$$\log t_{BKZ}(\delta_0) = \frac{1.8}{\log \delta_0} - 110$$

based on experiments with BKZ in the NTL library.

[11] Richard Lindner and Chris Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. In: *CT-RSA 2011*. Ed. by Aggelos Kiayias. Vol. 6558. LNCS. Springer, Heidelberg, Feb. 2011, pp. 319–339.

Lindner-Peikert Estimates

- The LP model does not fit the implementation of BKZ in NTL.
- NTL does not implement preprocessing of local blocks with BKZ recursively.[12]
- Hence, its enumeration requires $2^{\Omega(k^2)}$ time in block size k.

[12] See Damien's talk on lattice reduction (and fplll's implementation).

Lindner-Peikert Estimates

The LP model assumes a linear relation between $1/k$ and $\log(\delta_0)$, but from the “lattice rule-of-thumb” ($\delta_0 \approx k^{1/(2k)}$) we get[13]

Lemma

The log of the time complexity achieve a root-Hermite factor δ_0 with BKZ is

$$\Omega\left(\frac{\log(1/\log\delta_0)}{\log\delta_0}\right)$$

if calling the SVP oracle costs $2^{\Omega(k)}$.

[13] Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of Learning with Errors. In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203.

LP = A Subexponential Attack on Regev's LWE

Lemma

Given an LWE instance parametrised by n, $q = n^c$, $\alpha q = \sqrt{n}$. A lattice reduction algorithm achieving log root-Hermite factor

$$\log \delta_0 = \frac{\left(\left(c - \frac{1}{2}\right) \log n + \log \sqrt{\ln(1/\varepsilon)/\pi}\right)^2}{4 c n \log n}$$

can be used to distinguish the LWE distribution with advantage ε.[14]

Picking $\log \sqrt{\ln(1/\varepsilon)/\pi} \approx 1$ and $c = 2$ we gets

$$\log \delta_0 = \frac{9\ \log n}{32\, n} \text{ and } \log\left(t_{BKZ}(\delta_0)\right) = \frac{32\, n}{5\ \log n} - 110.$$

[14] Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of Learning with Errors. In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203.

BKZ 2.0 Based Guestimates

We'll assume[15]

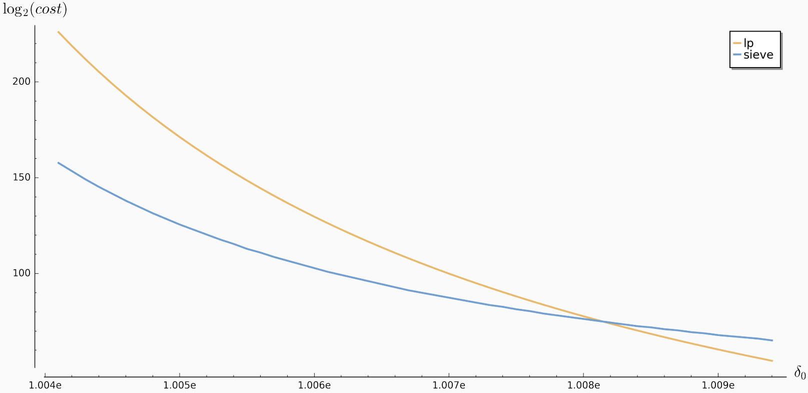
- $\delta_0 \approx \left(\frac{k}{2\pi e} (\pi k)^{\frac{1}{k}} \right)^{\frac{1}{2(k-1)}}$
- sieving is used as the SVP oracle in dimension k
- sieving in blocksize k costs $t_k = 2^{0.3366\, k + 12.31}$ clock cycles
- BKZ-k costs $\frac{n^3}{k^2} \log(n) \cdot t_k$ cycles

Samples

We will also assume access to as many samples as needed.

[15] Yuanmi Chen. Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe. PhD thesis. Paris 7, 2013; Thijs Laarhoven. Sieving for Shortest Vectors in Lattices Using Angular Locality-Sensitive Hashing. In: *CRYPTO 2015, Part I*. ed. by Rosario Gennaro and Matthew J. B. Robshaw. Vol. 9215. LNCS. Springer, Heidelberg, Aug. 2015, pp. 3–22. DOI: `10.1007/978-3-662-47989-6_1`; Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Analyzing Blockwise Lattice Algorithms Using Dynamical Systems. In: *CRYPTO 2011*. Ed. by Phillip Rogaway. Vol. 6841. LNCS. Springer, Heidelberg, Aug. 2011, pp. 447–464.

Comparison


$\log_2(cost)$
200
150
100
lp
sieve
δ_0
1.004e
1.005e
1.006e
1.007e
1.008e
1.009e

Rolling Example

We use the following LWE parameters as a rolling example throughout this talk.

- dimension $n = 2048$,
- modulus $q \approx 2^{63.4}$,
- noise parameter $\alpha \approx 2^{-60.4}$, i.e. standard deviation $\sigma \approx 3.2$,
- $h = 64$ components of the secret are ± 1, all other components are zero, $\sigma_s \approx 0.44$: $\mathcal{B}^{-}_{64}$

This is inspired by parameters choices in `HElib`.

Lattice Attacks

Dual Attack solve Short Integer Solutions problem (SIS) in the left kernel of $\mathbf{A}$, i.e.

$$\text{find a short } \mathbf{w} \text{ such that } \mathbf{w} \cdot \mathbf{A} = 0$$

and check if $\langle \mathbf{w}, \mathbf{c} \rangle = \mathbf{w} \cdot (\mathbf{A} \cdot \mathbf{s} + \mathbf{e}) = \langle \mathbf{w}, \mathbf{e} \rangle$ is short.

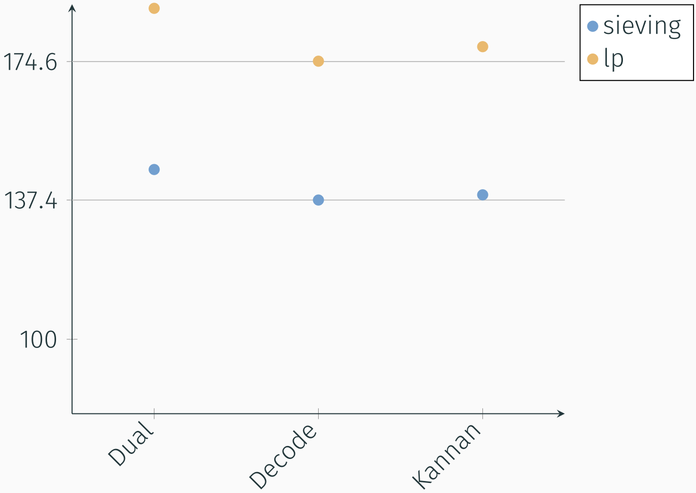
Primal Attack solve Bounded Distance Decoding problem (BDD), i.e.

$$\text{find } \mathbf{s}' \text{ s.t. } \|\mathbf{w} - \mathbf{c}\| \text{ with } \mathbf{w} = \mathbf{A} \cdot \mathbf{s}' \text{ is minimised}$$

using

- Kannan’s embedding or
- Babai’s nearest planes (Decoding).

Base Line



Swapping Error and Secret

Swapping Error and Secret

> *"applying the reduction technique of Applebaum et al.*[16] *to switch the key with part of the error vector, thus getting a smaller LWE error."*[17]

[16] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Heidelberg, Aug. 2009, pp. 595–618.

[17] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic Evaluation of the AES Circuit. Cryptology ePrint Archive, Report 2012/099. `http://eprint.iacr.org/2012/099`. 2012.

Swapping Error and Secret

- Let A_0 denotes the first n rows of A, A_1 the next n rows, etc.
- $\mathbf{e}_0, \mathbf{e}_1, \ldots$ are the corresponding parts of the error vector and
- $\mathbf{c}_0, \mathbf{c}_1, \ldots$ the corresponding parts of $\mathbf{c}$.
- For $i = 0$ we have $\mathbf{c}_0 = \mathsf{A}_0 \cdot \mathbf{s} + \mathbf{e}_0$ or

$$\mathsf{A}_0^{-1} \cdot \mathbf{c}_0 = \mathbf{s} + \mathsf{A}_0^{-1}\mathbf{e}_0.$$

- For $i > 0$ we have $\mathbf{c}_i = \mathsf{A}_i \cdot \mathbf{s} + \mathbf{e}_i$, which together with the above gives

$$\mathsf{A}_i \cdot \mathsf{A}_0^{-1} \cdot \mathbf{c}_0 - \mathbf{c}_i = \mathsf{A}_i \cdot (\mathbf{s} + \mathsf{A}_0^{-1}\mathbf{e}_0) - \mathbf{c}_i = \mathsf{A}_i \cdot \mathsf{A}_0^{-1}\mathbf{e}_0 - \mathbf{e}_i.$$

Bai-Gal Algorithm

- Consider the lattice

$$\Lambda = \{\mathbf{v} \in \mathbb{Z}^{n+m} | (\mathbf{A}|\mathbf{I}_m) \cdot \mathbf{v} \equiv 0 \pmod q\}$$

- It has an unusually short vector $(\mathbf{s}||\mathbf{e})$.
- When $\|\mathbf{s}\| \ll \|\mathbf{e}\|$, the vector $(\mathbf{s}||\mathbf{e})$ is uneven in length.
- Rescale the first part to have the same norm as the second.[18]

[18] Shi Bai and Steven D. Galbraith. Lattice Decoding Attacks on Binary LWE. . In: *ACISP 14*. Ed. by Willy Susilo and Yi Mu. Vol. 8544. LNCS. Springer, Heidelberg, July 2014, pp. 322–337. DOI: 10.1007/978-3-319-08344-5_21.

Bai-Gal Algorithm

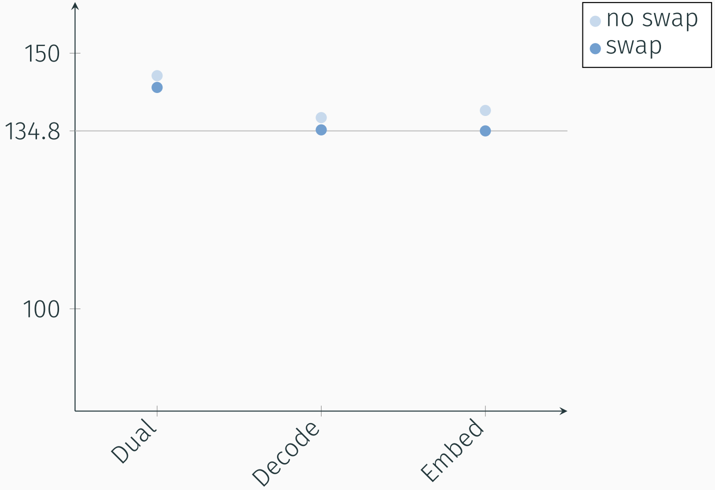
- Consider the lattice

$$\Lambda = \{\mathbf{v} \in \mathbb{Z}^{n+m} | (\mathbf{A}|\mathbf{I}_m) \cdot \mathbf{v} \equiv 0 \pmod q\}$$

- It has an unusually short vector $(\mathbf{s}||\mathbf{e})$.
- When $\|\mathbf{s}\| \ll \|\mathbf{e}\|$, the vector $(\mathbf{s}||\mathbf{e})$ is uneven in length.
- Rescale the first part to have the same norm as the second.[18]
  - When $\mathbf{s} \leftarrow_{\$} \mathcal{B}^{-}$, the volume of the lattice is scaled by σ^n.
  - When $\mathbf{s} \leftarrow_{\$} \mathcal{B}^{+}$ the volume of the lattice is scaled by $(2\sigma)^n$ because we can scale by 2σ and then rebalance.
  - When $\mathbf{s} \leftarrow_{\$} \mathcal{B}^{\pm}_{hw}$ the volume is scaled depending on the *hw*.

[18] Shi Bai and Steven D. Galbraith. Lattice Decoding Attacks on Binary LWE. . In: *ACISP 14*. Ed. by Willy Susilo and Yi Mu. Vol. 8544. LNCS. Springer, Heidelberg, July 2014, pp. 322–337. DOI: 10.1007/978-3-319-08344-5_21.

Swapping Error and Secret: Sieving



For our rolling example this reduces α from $2^{-60.4}$ to $\approx 2^{-60.8}$

Modulus Switching

Modulus Switching

Lemma

Let $(\mathbf{a}, c) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ *be an LWE sample and*

$$p \approx \sqrt{\frac{2\pi\, n}{12}} \cdot \frac{\sigma_s}{\alpha},$$

where σ_s *is the standard deviation of components of* $\mathbf{s}$*. If* $p < q$ *then*

$$\left(\left\lfloor \frac{p}{q} \cdot \mathbf{a} \right\rceil , \left\lfloor \frac{p}{q} \cdot c \right\rceil \right) \text{ in } \mathbb{Z}_p^n \times \mathbb{Z}_p$$

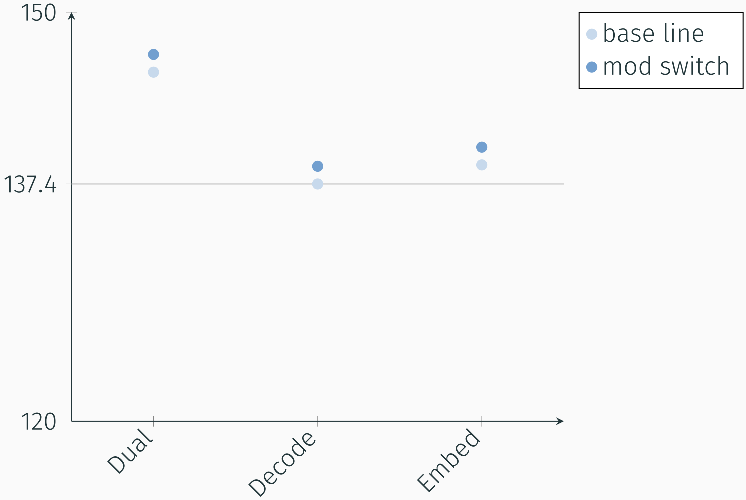
follows a distribution close to an LWE distribution with $n, \sqrt{2}\,\alpha, p$*.*[19]

[19] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. . In: *52nd FOCS*. ed. by Rafail Ostrovsky. IEEE Computer Society Press, Oct. 2011, pp. 97–106.

Modulus Switching in Cryptanalysis

When the secret is much smaller than the noise, applying modulus switching produces an easier LWE problem.

Modulus Switching: Sieving



Modulus Switching in Combinatorial Dual Attack

- BKW can seen as a combinatorial version of the Dual Attack.
- It was originally proposed for Learning Parity with Noise (LPN) which can be viewed as a special case of LWE over $\mathbb{Z}_2$.
- For BKW, variants of modulus switching lead to big performance gains.

BKW Algorithm

Assume $(\mathbf{a}_{21}, \mathbf{a}_{22}) = (0, 1)$, then:

$$
\left(\begin{array}{cc|ccc|c}
\mathbf{a}_{11} & \mathbf{a}_{12} & \mathbf{a}_{13} & \cdots & \mathbf{a}_{1n} & c_1 \\
\mathbf{a}_{21} & \mathbf{a}_{22} & \mathbf{a}_{23} & \cdots & \mathbf{a}_{2n} & c_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \\
\mathbf{a}_{m1} & \mathbf{a}_{m2} & \mathbf{a}_{m3} & \cdots & \mathbf{a}_{mn} & c_m
\end{array}\right)
$$

$$
-\left[\begin{array}{cc|ccc|c}
0 & 0 & \mathbf{t}_{13} & \cdots & \mathbf{t}_{1n} & c_{t,1} \\
0 & 1 & \mathbf{t}_{23} & \cdots & \mathbf{t}_{2n} & c_{t,2} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \\
q-1 & q-1 & \mathbf{t}_{q^2 3} & \cdots & \mathbf{t}_{q^2 n} & c_{t,q^2}
\end{array}\right]
$$

$$
\Rightarrow \left(\begin{array}{cc|ccc|c}
\mathbf{a}_{11} & \mathbf{a}_{12} & \mathbf{a}_{13} & \cdots & \mathbf{a}_{1n} & \tilde{c}_1 \\
0 & 0 & \tilde{\mathbf{a}}_{23} & \cdots & \tilde{\mathbf{a}}_{2n} & \tilde{c}_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \\
\mathbf{a}_{m1} & \mathbf{a}_{m2} & \mathbf{a}_{m3} & \cdots & \mathbf{a}_{mn} & c_m
\end{array}\right)
$$

Lazy Modulus Switching

- Create elimination tables which only eliminate the most significant bits
- As a consequence columns are not reduced to zero but to small entries.
- This can be seen as a lazy variant of modulus switching.[20]

[20] Martin R. Albrecht, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Lazy Modulus Switching for the BKW Algorithm on LWE. . In: *PKC 2014*. Ed. by Hugo Krawczyk. Vol. 8383. LNCS. Springer, Heidelberg, Mar. 2014, pp. 429–445. DOI: `10.1007/978-3-642-54631-0_25`.

Lazy Modulus Switching

- Create elimination tables which only eliminate the most significant bits
- As a consequence columns are not reduced to zero but to small entries.
- This can be seen as a lazy variant of modulus switching.[20]
- When eliminating higher order bits in columns with bigger indices, the noise of already reduced columns grows back.

[20] Martin R. Albrecht, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Lazy Modulus Switching for the BKW Algorithm on LWE. . In: *PKC 2014*. Ed. by Hugo Krawczyk. Vol. 8383. LNCS. Springer, Heidelberg, Mar. 2014, pp. 429–445. DOI: `10.1007/978-3-642-54631-0_25`.

Uneven Noise Contribution

$$(-1,-9|\ \ 7,-9|-1,\ \ 6)\ -\ (-2,-9|-5,\ \ 9|-5,-4)\quad ,\quad (\ \ 3,-1|\ \ 0,\ \ 0|\ \ 2,\ \ 6)\ -\ (\ \ 4,6|-2,\ \ 7|-4,-9)$$

$$=\qquad\qquad\qquad\qquad\qquad\qquad =$$

$$(\ \ 1,\ \ 0|-7,\ \ 1|\ \ 4,-9)\qquad -\qquad (-1,\ \ 1|-6,\ \ 2|\ \ 6,-\ \ 4)$$

$$=$$

$$(\ \ 2,-1|-1,-1|\ \ 2,\ \ 5)$$

Balancing Noise

- Pick decreasing moduli (increasing noise levels) for consecutive blocks to address this problem.
- Complexity now dominated by the size of the first table for eliminating first components.
- To compensate for this, choose increasing blocksizes b_i for each block.[21]

[21] Paul Kirchner and Pierre-Alain Fouque. An Improved BKW Algorithm for LWE with Applications to Cryptography and Lattices. In: *CRYPTO 2015, Part I*. ed. by Rosario Gennaro and Matthew J. B. Robshaw. Vol. 9215. LNCS. Springer, Heidelberg, Aug. 2015, pp. 43–62. DOI: 10.1007/978-3-662-47989-6_3.

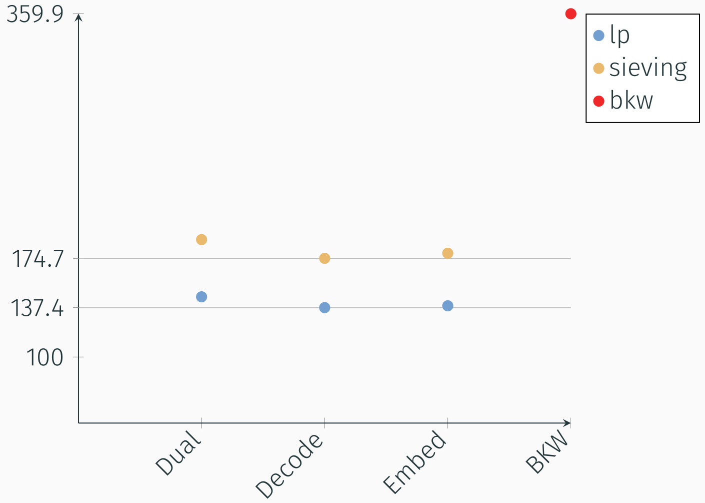
Coded-BKW

This approach can be generalised

- Consider modulus switching as a special form of quantisation (also done in [KF15])
- Choose appropriate lattice code to find good quantisation
- Consider blocks of size b_i as messages which are thrown into buckets based on the codeword they correspond to.[22]

[22] Qian Guo, Thomas Johansson, and Paul Stankovski. Coded-BKW: Solving LWE Using Lattice Codes. In: *CRYPTO 2015, Part I*. ed. by Rosario Gennaro and Matthew J. B. Robshaw. Vol. 9215. LNCS. Springer, Heidelberg, Aug. 2015, pp. 23–42. DOI: 10.1007/978-3-662-47989-6_2.

Coded-BKW



Plain BKW costs $2^{1310.4}$ bit operations.

Modulus Switching for Dual Attack

- Lazy modulus switching proceeds from the observation that we do not need to find $\mathsf{v} \cdot \mathsf{A} \equiv 0 \bmod q$, but any short enough $\mathsf{v} \cdot \mathsf{A}$ suffices.
- Consider the dual attack lattice for the LWE normal form

$$\Lambda(\mathsf{A}) = \{(\mathsf{x}, \mathsf{y}) \in \mathbb{Z}^m \times \mathbb{Z}^n : \mathsf{x} \cdot \mathsf{A} \equiv \mathsf{y} \bmod q\}$$

- Given a short vector $\mathsf{v} = (\mathsf{v}', \mathsf{w}') \in \Lambda(\mathsf{A})$ compute

$$\mathsf{v}' \cdot \mathsf{c} = \mathsf{v}' \cdot (\mathsf{A} \cdot \mathsf{s} + \mathsf{e}) = \langle \mathsf{w}', \mathsf{s} \rangle + \langle \mathsf{v}', \mathsf{e} \rangle$$

Modulus Switching for Dual Attack

- Aim is to balance $\| \langle \mathsf{w}', \mathsf{s} \rangle \| \approx \| \langle \mathsf{v}', \mathsf{e} \rangle \|$ when $\|\mathsf{s}\|$ is small.
- Similar to the Bai-Gail algorithm, consider the scaled dual attack lattice

$$\Lambda(\mathsf{A}) = \{(\mathsf{x}, \mathsf{y}/c) \in \mathbb{Z}^m \times (1/c \cdot \mathbb{Z})^n : \mathsf{x} \cdot \mathsf{A} \equiv \mathsf{y} \bmod q\}$$

  for some constant c.
- Lattice reduction produces a vector $(\mathsf{v}', \mathsf{w}')$ with

$$\|(\mathsf{v}', \mathsf{w}')\| \approx \delta_0^{(m+n)} \cdot (q/c)^{n/(m+n)}.$$

- The final error we aim to distinguish from uniform is

$$e = \mathsf{v}' \cdot \mathsf{A} \cdot \mathsf{s} + \langle \mathsf{v}', \mathsf{e} \rangle = \langle c \cdot \mathsf{w}', \mathsf{s} \rangle + \langle \mathsf{v}', \mathsf{e} \rangle .$$

Modulus Switching for Dual Attack

From

$$e = \mathsf{v}' \cdot \mathsf{A} \cdot \mathsf{s} + \langle \mathsf{v}', \mathsf{e} \rangle = \langle c \cdot \mathsf{w}', \mathsf{s} \rangle + \langle \mathsf{v}', \mathsf{e} \rangle\,.$$

we find c by solving

$$\sqrt{h}\, c = \frac{\alpha\, q}{\sqrt{2\,\pi}} \cdot \sqrt{m - n}$$

which equalises the noise contributions of both parts of the sum.

Modulus Switching for Dual Attack

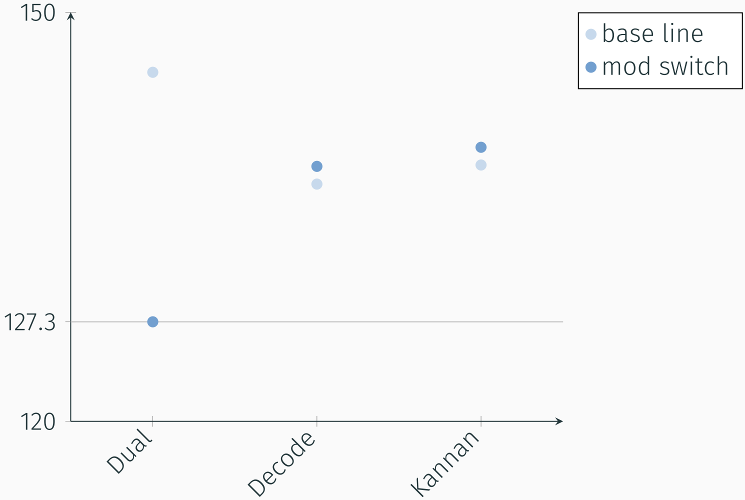
Lemma

Let $m = 2\,n$ and $c = \frac{\alpha\,q}{\sqrt{2\,\pi\,h}} \cdot \sqrt{m-n}$. A lattice reduction algorithm achieving δ_0 such that

$$\log\delta_0 = \frac{\log\left(\frac{\sqrt{8\,\pi}\,(\log(\varepsilon)/\pi)\,\sqrt{n}}{(2\,\pi+1)\,\alpha\sqrt{h}}\right)}{4\,n}$$

leads to an algorithm solving decisional LWE with $\mathbf{s} \leftarrow_{\$} \mathcal{B}^{-}_{64}$ instance with advantage ε and the same cost.

Modulus Switching: Sieving



Sparse Secrets

Exploiting Sparse Secrets

Approaches so far exploit small secrets, but in HELib the secret is sparse, i.e. most components are zero.

$\mathbb{Z}_q^n \approx \mathbb{Z}_{q^2}^{n/2} \approx \mathbb{Z}_{q^n}$

LWE in dimension n and with modulus q is equivalent to LWE in dimension n/k and modulus q^k.[23]

Let $n = 2$, $A = \mathsf{a}_0 \cdot q + \mathsf{a}_1 \pmod{q^2}$ and $S = \mathsf{s}_0 + \mathsf{s}_1 \cdot q \pmod{q^2}$.

$$\begin{aligned}
A \cdot S &= (\mathsf{a}_0 \cdot q + \mathsf{a}_1) \cdot (\mathsf{s}_0 + \mathsf{s}_1 \cdot q) && \pmod{q^2}\\
&= \mathsf{a}_0 \cdot q \cdot \mathsf{s}_0 + \mathsf{a}_1 \cdot \mathsf{s}_0 + \mathsf{a}_0 \cdot q \cdot \mathsf{s}_1 \cdot q + \mathsf{a}_1 \cdot \mathsf{s}_1 \cdot q && \pmod{q^2}\\
&= (\mathsf{a}_0 \cdot \mathsf{s}_0 + \mathsf{a}_1 \cdot \mathsf{s}_1) \cdot q + \mathsf{a}_0 \cdot \mathsf{s}_1 \cdot q^2 + \mathsf{a}_1 \cdot \mathsf{s}_0 && \pmod{q^2}\\
&\approx (\langle \mathsf{a}, \mathsf{s} \rangle \pmod{q}) \cdot q && \pmod{q^2}
\end{aligned}$$

[23] Zvika Brakerski et al. Classical hardness of learning with errors. In: *45th ACM STOC*. ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM Press, June 2013, pp. 575–584.

$\mathbb{Z}_q^n \approx \mathbb{Z}_{q^2}^{n/2} \approx \mathbb{Z}_{q^n}$ for Sparse Secrets

- Transform instance in dimension n to instance in dimension $n/2$ and with modulus q^2.
- The new secret is $\mathsf{S}_i = \mathsf{s}_{2i+0} + \mathsf{s}_{2i+1} \cdot q \pmod{q^2}$ for $0 \leq i < n/2$ where $\mathsf{s}_{2i+1} = 0$ with good probability.
- When this condition holds for all S_i, the secret is shorter than the noise by a factor of $\approx q$.
- Apply your favourite small secret solving strategy.

Ignoring Components

- When the secret is sparse, most columns of $\mathbf{A}$ are irrelevant.
- In our example, the probability that a random coordinate is non-zero is

$$64/2048 = 1/32.$$

- Ignoring k random components in dimension n for an instance with h nonzero components will ignore only zero components with probability

$$P_k = \prod_{i=0}^{k-1} \left(1 - \frac{h}{n-i}\right) = \frac{\binom{n-h}{k}}{\binom{n}{k}}$$

- Solving $\approx 1/P_k$ instances in dimension $n-k$ with sufficiently high advantage solves our instance at dimension n.

Ignoring Components in Dual Attack

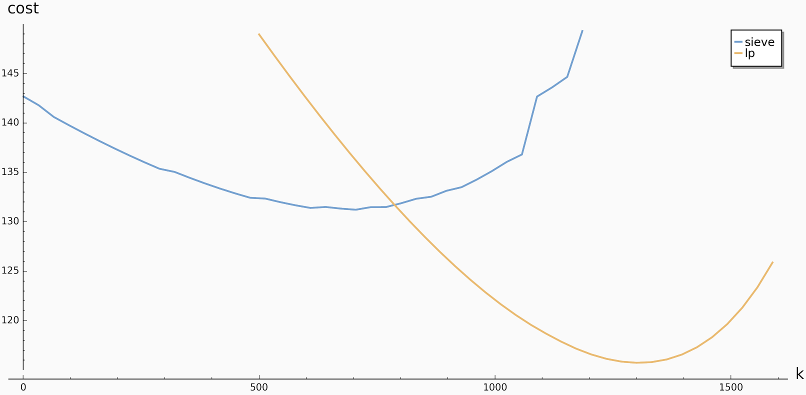
$$0 \overset{?}{=} \begin{array}{c} \mathbf{v} \\ \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{m-3} \\ v_{m-2} \\ v_{m-1} \end{pmatrix} \end{array} \cdot \begin{array}{c} \mathbf{A} \\ \left(\begin{array}{ccc|ccc} a_{0,0} & \cdots & a_{0,k-1} & a_{0,k} & \cdots & a_{0,n-1} \\ a_{1,0} & \cdots & a_{1,k-1} & a_{1,k} & \cdots & a_{1,n-1} \\ a_{2,0} & \cdots & a_{2,k-1} & a_{2,k} & \cdots & a_{2,n-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m-3,0} & \cdots & a_{m-3,k-1} & a_{m-3,k} & \cdots & a_{m-3,n-1} \\ a_{m-2,0} & \cdots & a_{m-2,k-1} & a_{m-2,k} & \cdots & a_{m-2,n-1} \\ a_{m-1,0} & \cdots & a_{m-1,k-1} & a_{m-1,k} & \cdots & a_{m-1,n-1} \end{array}\right) \end{array} \cdot \begin{array}{c} \mathbf{s} \\ \left(\begin{array}{c} s_0 \\ \vdots \\ s_{k-1} \\ \hline s_k \\ \vdots \\ s_{n-1} \end{array}\right) \end{array}$$

$$\overset{?}{=} \left(\begin{array}{ccc|ccc} a'_{0,0} & \cdots & a'_{0,k-1} & 0 & \cdots & 0 \end{array}\right) \cdot \left(\begin{array}{c} s_0 \\ \vdots \\ s_{k-1} \\ \hline s_k \\ \vdots \\ s_{n-1} \end{array}\right)$$

Ignoring Components in Dual Attack

$$
\begin{aligned}
0 &= \overset{\mathbf{v}}{\begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{m-3} \\ v_{m-2} \\ v_{m-1} \end{pmatrix}} \cdot \overset{\mathbf{A}}{\left(\begin{array}{ccc|ccc} a_{0,0} & \cdots & a_{0,k-1} & a_{0,k} & \cdots & a_{0,n-1} \\ a_{1,0} & \cdots & a_{1,k-1} & a_{1,k} & \cdots & a_{1,n-1} \\ a_{2,0} & \cdots & a_{2,k-1} & a_{2,k} & \cdots & a_{2,n-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m-3,0} & \cdots & a_{m-3,k-1} & a_{m-3,k} & \cdots & a_{m-3,n-1} \\ a_{m-2,0} & \cdots & a_{m-2,k-1} & a_{m-2,k} & \cdots & a_{m-2,n-1} \\ a_{m-1,0} & \cdots & a_{m-1,k-1} & a_{m-1,k} & \cdots & a_{m-1,n-1} \end{array}\right)} \cdot \overset{\mathbf{s}}{\left(\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline s_k \\ \vdots \\ s_{n-1} \end{array}\right)} \\
&= \left(\begin{array}{ccc|ccc} a'_{0,0} & \cdots & a'_{0,k-1} & 0 & \cdots & 0 \end{array}\right) \cdot \left(\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline s_k \\ \vdots \\ s_{n-1} \end{array}\right)
\end{aligned}
$$

Dual Attack



Solving $1/P_k$ instances with $n = 2048 - k$, $\alpha \approx 2^{-60.4}$ and $q \approx 2^{63.4}$.

Postprocessing

$$
a'_{0,0} = \overset{\mathtt{v}}{\begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{m-3} \\ v_{m-2} \\ v_{m-1} \end{pmatrix}} \cdot \overset{\mathtt{A}}{\left(\begin{array}{ccc|ccc} a_{0,0} & \cdots & a_{0,k-1} & a_{0,k} & \cdots & a_{0,n-1} \\ a_{1,0} & \cdots & a_{1,k-1} & a_{1,k} & \cdots & a_{1,n-1} \\ a_{2,0} & \cdots & a_{2,k-1} & a_{2,k} & \cdots & a_{2,n-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m-3,0} & \cdots & a_{m-3,k-1} & a_{m-3,k} & \cdots & a_{m-3,n-1} \\ a_{m-2,0} & \cdots & a_{m-2,k-1} & a_{m-2,k} & \cdots & a_{m-2,n-1} \\ a_{m-1,0} & \cdots & a_{m-1,k-1} & a_{m-1,k} & \cdots & a_{m-1,n-1} \end{array}\right)} \cdot \overset{\mathtt{s}}{\left(\begin{array}{c} 1 \\ \vdots \\ 0 \\ \hline s_k \\ \vdots \\ s_{n-1} \end{array}\right)}
$$

$$
= \left(\begin{array}{ccc|ccc} a'_{0,0} & \cdots & a'_{0,k-1} & 0 & \cdots & 0 \end{array}\right) \cdot \left(\begin{array}{c} 1 \\ \vdots \\ 0 \\ \hline s_k \\ \vdots \\ s_{n-1} \end{array}\right)
$$

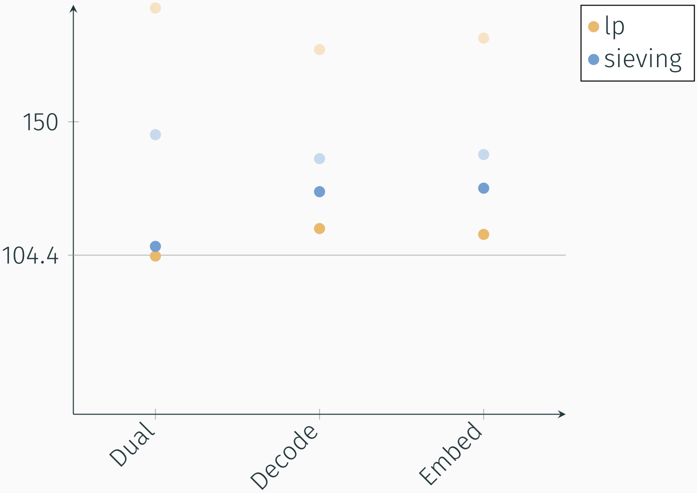
Postprocessing

The probability to drop $k - j$ columns with $s_i = 0$ and exactly j components with $s_i \neq 0$ is

$$P_{k,j} = \frac{\binom{n-h}{k-j}\binom{h}{j}}{\binom{n}{k}}$$

- Repeat experiment $\left(\sum_{j=0}^{\ell} P_{k,j}\right)^{-1}$ times
- Perform $\sum_{i=0}^{\ell} \binom{k}{i} \cdot 2^i$ checks against uniform distribution, reusing short vector output by lattice reduction.

Ignoring Components



Results

Results

	Strategy	Dual		Dec		Embed	
		sieve	lp	sieve	lp	sieve	lp
0	base line	145.6	188.9	137.4	174.7	138.8	178.6
1	secret ↔	143.3	185.2	135.0	171.6	134.8	171.9
2	modulus ↔	127.4	159.5	138.7	177.4	140.1	180.9
3	drop	107.3	104.1	126.1	113.5	127.3	111.5
4	++	96.8	92.9	125.4	113.2	127.3	111.5

After dropping some components the resulting instance still has a sparse and small secret → combine strategies: “++“.

Thank you

Questions?