On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL

Martin R. Albrecht

Information Security Group, Royal Holloway, University of London

Learning with Errors

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

Given $(\mathbf{A}, \mathbf{c})$ with uniform $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, uniform $\mathbf{s} \in \mathbb{Z}_q^n$ and small $\mathbf{e} \in \mathbb{Z}^m$ is $\mathbf{c} \leftarrow\!\!\$\, \mathcal{U}\left(\mathbb{Z}_q^m\right)$ or

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

[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.

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, implemented `HELib`

FV Junfeng Fan and Frederik Vercauteren. Somewhat Practical Fully Homomorphic Encryption. Cryptology ePrint Archive, Report 2012/144. `http://eprint.iacr.org/2012/144`. 2012, implemented in SEAL v2

Small Secrets

- `HElib` typically chooses $\mathbf{s}$ such that $w = 64$ entries are ± 1 and all remaining entries are 0, regardless of dimension n.
- `SEAL` chooses $\mathbf{s}_i \xleftarrow{\$} \{-1, 0, 1\}$.

How many bits of security does this cost?

Hardness: Reductions v Constructions

"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 $\{0,1\}$."[2]

"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."[3]

---

[2] 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.

[3] 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.

Lattice Attacks

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

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

using

- uSVP embedding or
- Babai's nearest planes resp. enumeration.

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.

Dual Attack

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

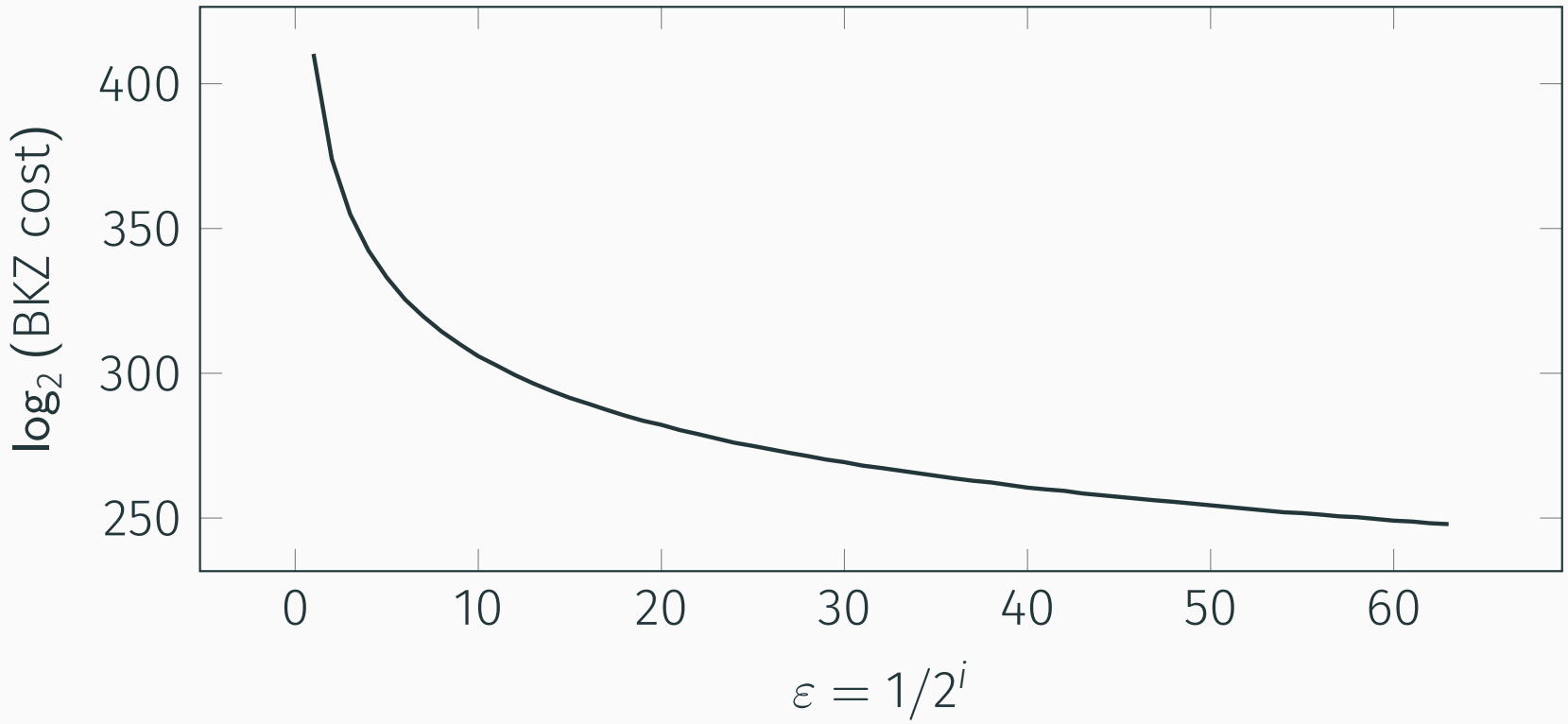
1. Construct a basis of the dual lattice from $\mathbf{A}$.
2. Run lattice reduction algorithm to obtain short vectors $\mathbf{v}_i$.
3. Check if $\langle \mathbf{v}_i, \mathbf{c} \rangle$ are small.[4]

[4] 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, Heidelberg, 2009, pp. 147–191.

1. Amortising Costs

Dual Attack: Trade-off

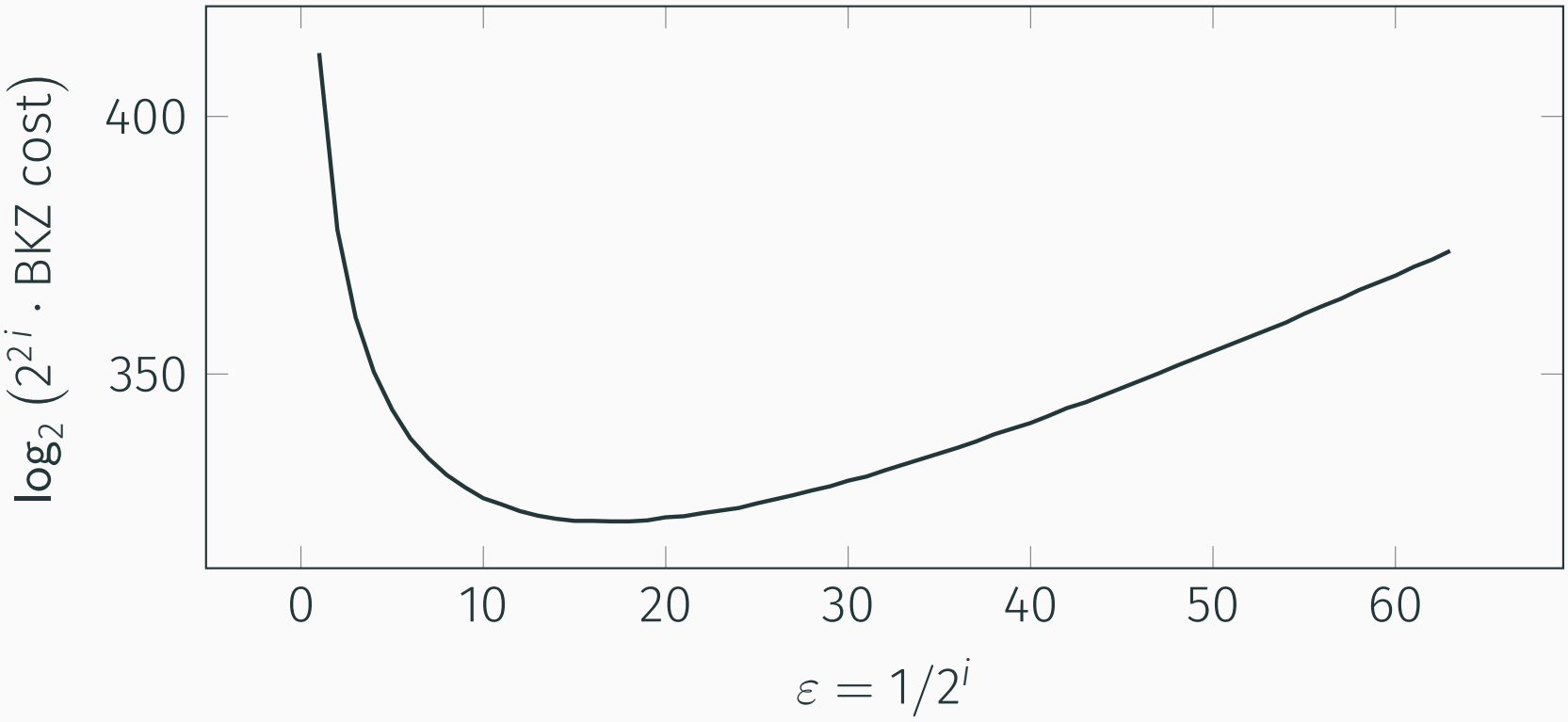
Given an LWE instance characterised by n, α, q and a vector $\mathbf{v}$ of length $\|\mathbf{v}\|$ such that $\mathbf{v} \cdot \mathbf{A} \equiv 0 \pmod{q}$, the advantage ε of distinguishing $\langle \mathbf{v}, \mathbf{c} \rangle$ from random is close to[5] $\exp(-\pi(\|\mathbf{v}\| \cdot \alpha)^2)$.



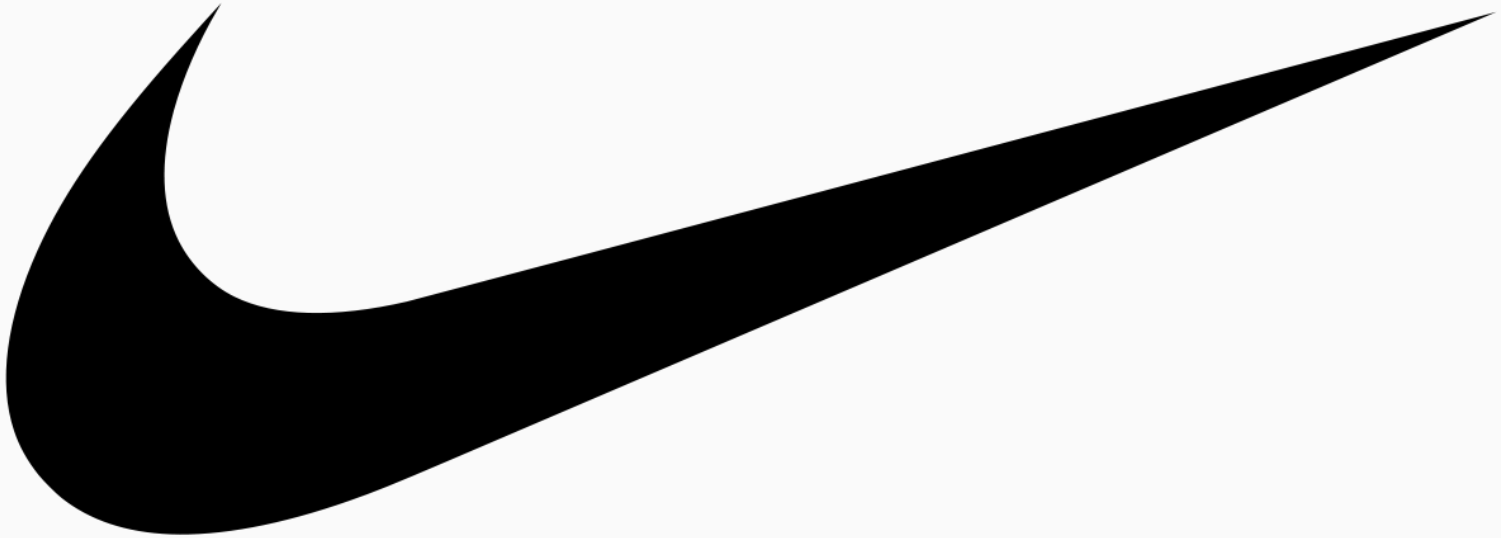
[5] Richard Lindner and Chris Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. In: *CT-RSA 2011*. Ed. by Aggelos Kiayias. Vol. 6558. LNCS. Springer, Heidelberg, Feb. 2011, pp. 319–339.

Amplifying Advantage

To achieve constant advantage, repeat experiment $\approx 1/\varepsilon^2$ times for majority vote.



Just do it™



Amortising Costs

Avoiding $1/\varepsilon^2$ calls to BKZ in block size β.

1. $\mathbf{L} \leftarrow$ basis for $\{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} \cdot \mathbf{A} \equiv 0 \bmod q\}$
2. $\mathbf{R} \leftarrow$ BKZ-β reduced basis for $\mathbf{L}$
3. Repeat:
   - 3.1 $\mathbf{U} \leftarrow\!\!\$$ a sparse unimodular matrix with small entries
   - 3.2 $\mathbf{R}_i \leftarrow$ BKZ-β' reduced basis for $\mathbf{U} \cdot \mathbf{R}$
   - 3.3 $\mathbf{y}_i \leftarrow$ shortest row vector in $\mathbf{R}_i$
   - 3.4 $w_i \leftarrow \langle \mathbf{y}_i, \mathbf{c} \rangle$
4. Decide if w_i is uniform or not.

We give empirical evidence that the quality of $\mathbf{R}_i$ isn't "too bad": for $\beta' = 2$, they are $< 2 \cdot \delta_0^m \cdot q^{n/m}$ with δ_0 for BKZ-β.

2. Scaling

Scaling for Dual Attack

- We do not need to find $\mathbf{v} \cdot \mathbf{A} \equiv 0 \bmod q$, but any short $\mathbf{v}$ such that $\mathbf{v} \cdot \mathbf{A} = \mathbf{w}$ is short suffices.
- Consider the normal form of the dual attack on LWE

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

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

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

Scaling for Dual Attack

- Aim is to balance $\| \langle \mathbf{w}, \mathbf{s} \rangle \| \approx \| \langle \mathbf{v}, \mathbf{e} \rangle \|$ when $\|\mathbf{s}\|$ is small.
- Scale the lattice[6]

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

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

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

- The final error we aim to distinguish from uniform is

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

[6] 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.

Scaling for Dual Attack

From

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

we find c by solving

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

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

3. Sparse Secrets

Ignoring Components

- When the secret is sparse, most columns of $\mathbf{A}$ are irrelevant.
- The probability of getting lucky ($\mathbf{s}_i = 0$) when ignoring k random components in dimension n with in total h entries $\mathbf{s}_i \neq 0$ follows a hypergeometric distribution

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

- Solving (with high enough probability) $\approx 1/P_k$ instances in dimension $n - k$ solves our instance at dimension n.

Ignoring Components in Dual Attack

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

Ignoring Components in Dual Attack

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

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

$$
\langle c \cdot \mathbf{w}_{k:}, \mathbf{s}_{k:} \rangle + \langle \mathbf{v}, \mathbf{e} \rangle
$$

Postprocessing

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

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

$$a'_{0,0} + \langle c \cdot \mathbf{w}_{k:}, \mathbf{s}_{k:} \rangle + \langle \mathbf{v}, \mathbf{e} \rangle$$

Postprocessing

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

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

1. Repeat overall experiment $\left(\sum_{j=0}^{\ell} P_{k,j}\right)^{-1}$ times
2. For each:
   2.1 Perform $\sum_{i=0}^{\ell} \binom{k}{i} \cdot 2^i$ checks for shifted "small distributions", reusing short vector output by lattice reduction.

Overall

Silke

Variant of dual attack for small (and sparse) secrets:

1. Perform BKZ-β once and then run BKZ-β' with $\beta' < \beta$ to make many short vectors
2. Scale the normal form of the dual lattice.
3. If sparse, ignore presumed-zero columns, correcting for mistakes by checking for shifted distributions.

Results

n	1024	2048	4096	8192	16384
SEAL (pre 2.1) 80-bit					
$\log_2 q$	47.5	95.4	192.0	392.1	799.6
dual	83.1	78.2	73.7	71.1	70.6
SILKE$_{\text{small}}$	68.1	69.0	68.2	68.4	68.8
HElib 80-bit					
$\log_2 q$	47.0	87.0	167.0	326.0	638.0
dual	85.2	85.2	85.3	84.6	85.5
SILKE$_{\text{sparse}}$	61.3	65.0	67.9	70.2	73.1
HElib 128-bit					
$\log_2 q$	38.0	70.0	134.0	261.0	511.0
dual	110.7	110.1	109.3	108.8	108.9
SILKE$_{\text{sparse}}$	73.2	77.4	81.2	84.0	86.4

Thank you

Questions?

https://ia.cr/2017/047