SOME REMARKS ON SMALL SECRET LWE

Martin R. Albrecht @martinralbrecht

Auckland, December 4, 2015

Information Security Group, Royal Holloway, University of London

OUTLINE

Introduction

Warm Up

Modulus Switching

- Coded-BKW

Swapping Error and Secret

A Different Embedding Approach

Exploiting Sparse Secrets

Introduction

Warm Up

Modulus Switching

Coded-BKW

Swapping Error and Secret

A Different Embedding Approach

Exploiting Sparse Secrets

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

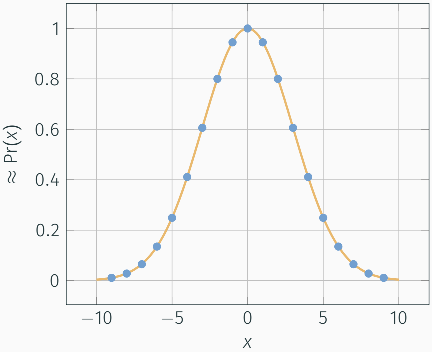
Given $(\mathbf{A}, \mathbf{c})$ with $\mathbf{c} \in \mathbb{Z}_q^m$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \in \mathbb{Z}_q^m$ do we have

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

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

[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,
  - noise size α,
  - number of samples m.
- Elements of $\mathbf{A}, \mathbf{s}, \mathbf{e}, \mathbf{c}$ are in $\mathbb{Z}_q$.
- $\mathbf{e}$ is sampled from a discrete Gaussian with width

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

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.[2]

[2] Benny Applebaum et al. 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.

- Some applications use much smaller secrets.
- For example, $\mathbf{s}_i \leftarrow \{-1, 0, 1\}$ or $\mathbf{s}_\mathrm{i} \leftarrow \{0, 1\}$.
- `HElib`[3] chooses $\mathbf{s}$ such that $h = 64$ entries are ± 1 and all remaining entries are 0, regardless of dimension n.

Question

How much 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.

"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 $\{0, 1\}$."[4]

[4] 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.

"[This work] suggests that this is overkill and that even $n \log \log n$ may be more than sufficient."[5]

[5] 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.

"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."[6]

[6] 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.

- $\mathcal{B}^+$ each component is independently sampled uniformly from $\{0, 1\}$.
- $\mathcal{B}^-$ each component is independently sampled uniformly from $\{-1, 0, 1\}$.
- $\mathcal{B}^+_{hw}$ like $\mathcal{B}^+$ but with guarantee that *hw* components are non-zero.
- $\mathcal{B}^-_{hw}$ like $\mathcal{B}^-$ but with guarantee that *hw* components are non-zero.

In the guestimates below, we assumed

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

`https://github.com/dstehle/fplll`
`https://github.com/malb/fpylll`

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 chosen in `HElib`.

Introduction

Warm Up

Modulus Switching

Coded-BKW

Swapping Error and Secret

A Different Embedding Approach

Exploiting Sparse Secrets

- Clearly, exhaustive search is an option for solving.
- This gives a complexity of about 2^n for $\mathcal{B}^+$ and 3^n for $\mathcal{B}^-$.
- For $\mathcal{B}^-_{64}$ we get a complexity of about $2^{64} \cdot \binom{n}{64}$.

Meet in the Middle

We can about square-root these complexities using standard time-memory trade-offs.

HElib

Plugging our example in gives expected costs of $\approx 2^{470}$ and $\approx 2^{235}$ operations, respectively.

Given $\mathbf{A}, \mathbf{c}$ with $\mathbf{c} = \mathbf{A} \times \mathbf{s} + \mathbf{e}$ or $\mathbf{c} \leftarrow_{\$} \mathbb{Z}_q^m$

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

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

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

Given $\mathbf{A}, \mathbf{c}$ with $\mathbf{c} = \mathbf{A} \times \mathbf{s} + \mathbf{e}$ or $\mathbf{c} \leftarrow_{\$} \mathbb{Z}_q^m$

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

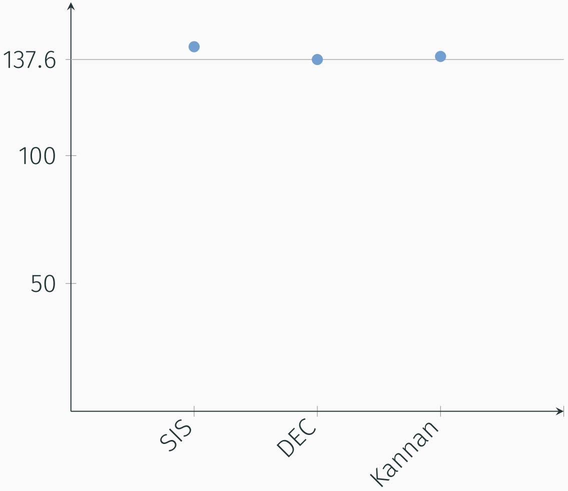
  $$\text{find a short } \mathbf{w} \text{ such that } \mathbf{w} \times \mathbf{A} = 0$$

  and check if $\langle \mathbf{w}, \mathbf{c} \rangle = \mathbf{w} \times (\mathbf{A} \times \mathbf{s} + \mathbf{e}) = \langle \mathbf{w}, \mathbf{e} \rangle$ is short.
- Solve the Bounded Distance Decoding problem (BDD), i.e.

  $$\text{find } \mathbf{s}' \text{ such that } |\mathbf{w} - \mathbf{c}| \text{ with } \mathbf{w} = \mathbf{A} \times \mathbf{s}' \text{ is minimised.}$$

  via Kannan's embedding or Babai's nearest planes.

STANDARD APPROACHES VS. ROLLING EXAMPLE



Introduction

Warm Up

Modulus Switching

- Coded-BKW

Swapping Error and Secret

A Different Embedding Approach

Exploiting Sparse Secrets

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 the secret $\mathbf{s}$. 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 $n, \sqrt{2}\alpha, p$.

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

- We usually simply assume that the rounding noise is also some Gaussian distribution.
- However, the rounding noise is not completely out of our control.
- We know one component that goes into making it:

$$\frac{p}{q} \cdot \mathsf{a} - \left\lfloor \frac{p}{q} \cdot \mathsf{a} \right\rceil$$

SLIGHTLY BETTER MODULUS SWITCHING FOR CRYPTANALYSIS II

Given known vectors $\mathsf{r}_i \leftarrow_{\$} \left(-\frac{1}{2}, \frac{1}{2}\right]^n$ and an unknown fixed vector $\mathsf{s} \leftarrow_{\$} \mathcal{B}$, we call $\mathcal{Q}_{\mathsf{s}}\left(\mathsf{r}_i\right)$ the distribution obtained by outputting $\lfloor\langle \mathsf{r}_i, \mathsf{s}\rangle\rceil$.

$$\mathcal{Q}_{\mathsf{s}}\left(\frac{p}{q}\cdot\mathsf{a} - \left\lfloor\frac{p}{q}\cdot\mathsf{a}\right\rceil\right) = \left\langle\frac{p}{q}\cdot\mathsf{a} - \left\lfloor\frac{p}{q}\cdot\mathsf{a}\right\rceil, \mathsf{s}\right\rangle_p + e'$$

Let $\mathbf{s} \leftarrow_{\$} \mathcal{B}^{+}$. Let $\mathbf{r}_i = \frac{p}{q} \cdot \mathbf{a}_i - \left\lfloor \frac{p}{q} \cdot \mathbf{a}_i \right\rceil$. Let $L_{\mathbf{s},\chi}^{(n)\prime\prime}$ be a distribution which outputs those $(\mathbf{a}_i', c_i')$ where $\sum \mathbf{r}_i' \leq c \cdot \sigma$ with σ the standard deviation of $\mathcal{Q}_{\mathbf{s}}(\mathbf{r}_i)$.

Then, $\mathcal{Q}_{\mathbf{s}}\left(\mathbf{r}_i'\right)$ for $\mathbf{r}_i' = \frac{p}{q} \cdot \mathbf{a}_i' - \left\lfloor \frac{p}{q} \cdot \mathbf{a}_i' \right\rceil$ satisfies:

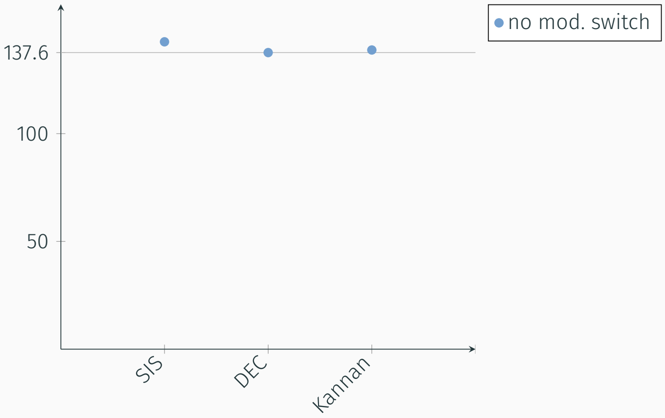
$$\Pr[\mathcal{Q}_{\mathbf{s}}(\mathbf{r}_i') > C \cdot \sigma] \leq \frac{\exp\left(-C^2 + c\,C - c^2/2\right)}{2\,\pi \cdot (C^2 - c\,C)}.$$

Compare:

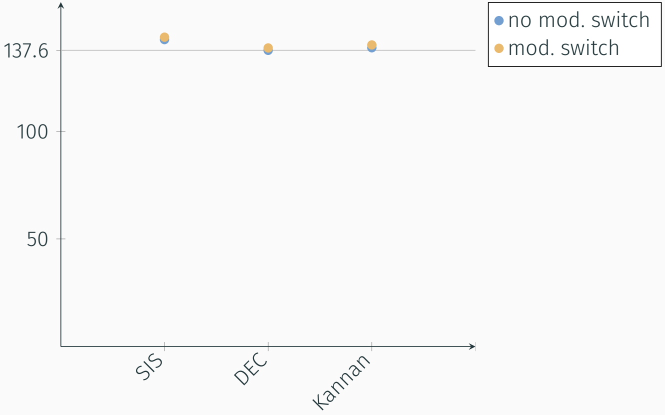
$$\Pr[\mathcal{Q}_{\mathbf{s}}(\mathbf{r}_i) > C \cdot \sigma] \leq \frac{\exp\left(-C^2/2\right)}{C\sqrt{2\pi}}.$$

MODULUS SWITCHING + LATTICE REDUCTION I

Applied to our rolling example:



Applied to our rolling example:



The BKW algorithm was first proposed for the Learning Parity with Noise (LPN) problem which can be viewed as a special case of LWE over $\mathbb{Z}_2$.

We considering $a \approx \log n$ 'blocks' of b elements each.

$$\left(\begin{array}{cc|ccc|c}
a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & c_0 \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & c_1 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \\
a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} & c_m
\end{array}\right)$$

For each block we build a table of all q^b possible values indexed by $\mathbb{Z}_q^b$.

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

For each $\mathbf{z} \in \mathbb{Z}_q^b$ we try to find a row in $\mathbf{A}$ such that it contains $\mathbf{z}$ as a subvector at the target indices.

We use these tables to eliminate b entries in other rows. Assume $(\mathbf{a}_{21}, \mathbf{a}_{22}) = (\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor + 1)$, then:

$$
\begin{pmatrix}
\mathbf{a}_{11} & \mathbf{a}_{12} & \mathbf{a}_{13} & \cdots & \mathbf{a}_{1n} & c_0 \\
\mathbf{a}_{21} & \mathbf{a}_{22} & \mathbf{a}_{23} & \cdots & \mathbf{a}_{2n} & c_1 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \\
\mathbf{a}_{m1} & \mathbf{a}_{m2} & \mathbf{a}_{m3} & \cdots & \mathbf{a}_{mn} & c_m
\end{pmatrix}
$$

$$
+ \quad \begin{bmatrix}
-\lfloor \frac{q}{2} \rfloor & -\lfloor \frac{q}{2} \rfloor & \mathbf{t}_{13} & \cdots & \mathbf{t}_{1n} & c_{t,0} \\
-\lfloor \frac{q}{2} \rfloor & -\lfloor \frac{q}{2} \rfloor + 1 & \mathbf{t}_{23} & \cdots & \mathbf{t}_{2n} & c_{t,1} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \\
\lfloor \frac{q}{2} \rfloor & \lfloor \frac{q}{2} \rfloor & \mathbf{t}_{q^2 3} & \cdots & \mathbf{t}_{q^2 n} & c_{t,q^2}
\end{bmatrix}
$$

$$
\Rightarrow \quad \begin{pmatrix}
\mathbf{a}_{11} & \mathbf{a}_{12} & \mathbf{a}_{13} & \cdots & \mathbf{a}_{1n} & c_0 \\
0 & 0 & \tilde{\mathbf{a}}_{23} & \cdots & \tilde{\mathbf{a}}_{2n} & \tilde{c}_1 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \\
\mathbf{a}_{m1} & \mathbf{a}_{m2} & \mathbf{a}_{m3} & \cdots & \mathbf{a}_{mn} & c_m
\end{pmatrix}
$$

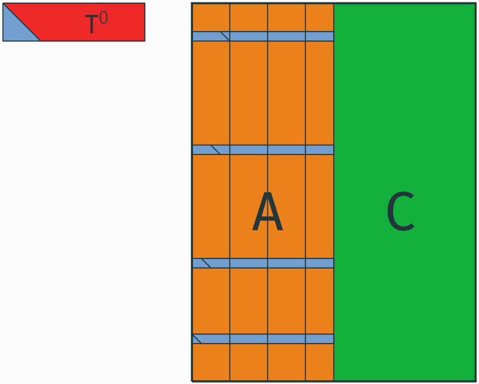
- When running the BKZ algorithm, only eliminate the most significant bits
- This can be seen as a lazy variant of modulus switching.

Martin R. Albrecht et al. 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

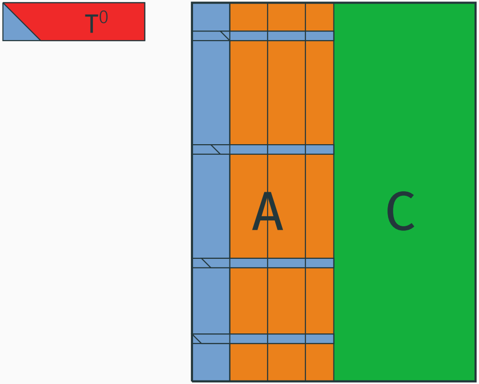
uneven noise contribution I

When eliminating higher order bits in latter tables of BKW, this leads to an increase in the noise of the components covered by earlier tables.

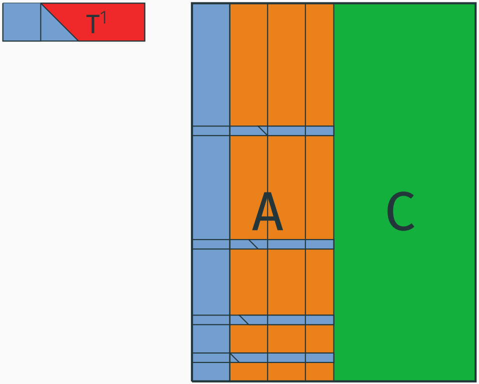
UNEVEN NOISE CONTRIBUTION II

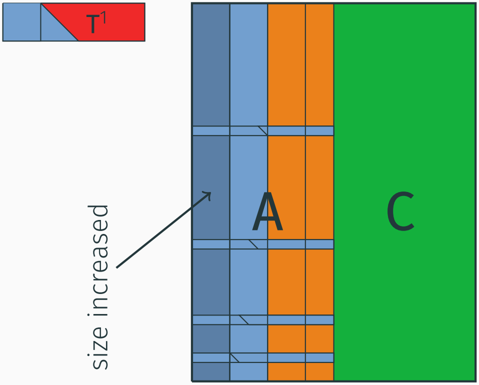


UNEVEN NOISE CONTRIBUTION III



UNEVEN NOISE CONTRIBUTION IV




T^1
A
C
size increased

- We could pick decreasing moduli (increasing noise levels) for consecutive blocks to address this problem.
- This, however, would increase the complexity which would now be dominated by the size of the table T^0.
- To compensate for this, we may choose increasing blocksizes b_i for each of the a blocks

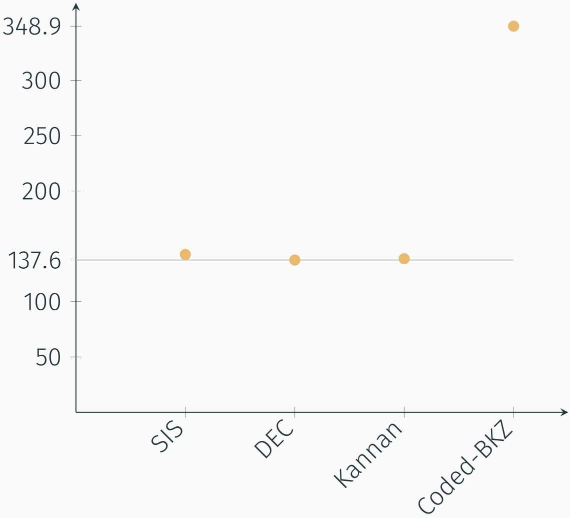
Paul Kirchner and Pierre-Alain Fouque. An Improved BKW Algorithm for LWE with Applications to Cryptography and Lattices. In: *CRYPTO 2015, Part I*. ed. by Rosario Gennaro and Matthew J. B. Robshaw. Vol. 9215. LNCS. Springer, Heidelberg, Aug. 2015, pp. 43–62. DOI: 10.1007/978-3-662-47989-6_3

This approach can be generalised

- Consider our 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.

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 II



Introduction

Warm Up

Modulus Switching

Coded-BKW

Swapping Error and Secret

A Different Embedding Approach

Exploiting Sparse Secrets

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

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

Benny Applebaum et al. 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

- We are given a random $m \times n$ matrix A mod q, and also an m-vector

$$\mathsf{c} = \mathsf{A} \cdot \mathsf{s} + \mathsf{e} \bmod q.$$

- Let A_0 denotes the first n rows of A, A_1 the next n rows, etc.
- $\mathsf{e}_0, \mathsf{e}_1, \dots$ are the corresponding parts of the error vector and $\mathsf{c}_0, \mathsf{c}_1, \dots$ the corresponding parts of c.
- We have $\mathsf{c}_0 = \mathsf{A}_0 \cdot \mathsf{s} + \mathsf{e}_0$ or $\mathsf{A}_0^{-1} \cdot \mathsf{c}_0 = \mathsf{s} + \mathsf{A}_0^{-1}\mathsf{e}_0$.
- Also, for $i > 0$ we have $\mathsf{c}_i = \mathsf{A}_i \cdot \mathsf{s} + \mathsf{e}_i$, which together with the above gives

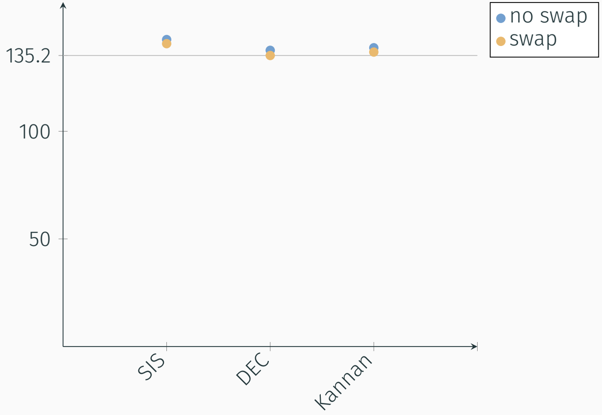
$$\mathsf{A}_i\mathsf{A}_0^{-1}\mathsf{c}_0 - \mathsf{c}_i = \mathsf{A}_i\mathsf{A}_0^{-1}\mathsf{e}_0 - \mathsf{e}_i.$$

- Set $\mathbf{B} = (\mathbf{A}_0^{-1} \mid \mathbf{A}_1 \cdot \mathbf{A}_0^{-1} \mid \dots)$ and $\mathbf{z} = (\mathbf{A}_0^{-1}\mathbf{c}_0 \mid \mathbf{A}_1\mathbf{A}_0^{-1}\mathbf{c}_1 \mid \dots)$, and also $\mathbf{f} = (\mathbf{s}|\mathbf{e}_1 \mid \dots)$ then we get the LWE instance

$$\mathbf{z} = \mathbf{B} \cdot \mathbf{e}_0 + \mathbf{f}$$

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

Applied to our rolling example:



Introduction

Warm Up

Modulus Switching

- Coded-BKW

Swapping Error and Secret

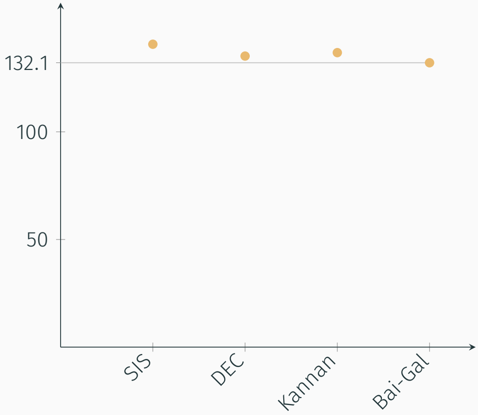
A Different Embedding Approach

Exploiting Sparse Secrets

- Let $m' = m + n$.
- We may embed our LWE lattice into a different lattice with uSVP structure:
$$L = \{\mathbf{v} \in \mathbb{Z}^{m'} | \mathbf{A}'\mathbf{v} \equiv 0 \bmod q\}$$
where
$$\mathbf{A}' = (\mathbf{A}|\mathbf{I}_m).$$
- The target short vector is now $(\mathbf{s}||\mathbf{e})$
- When $|\mathbf{s}| \ll |\mathbf{e}|$, the vector $(\mathbf{s}||\mathbf{e})$ is uneven.
- We may want to rescale the first components to have same size as the last components.

- When $\mathbf{s} \leftarrow_{\$} \mathcal{B}^{-}$, after an appropriate rescaling, the volume of the lattice is increased by σ^n.
- When $\mathbf{s} \leftarrow_{\$} \mathcal{B}^{+}$ the volume is increased by $(2\sigma)^n$ because we can scale by 2σ and then rebalance.
- When $\mathbf{s} \leftarrow_{\$} \mathcal{B}^{\pm}_{hw}$ the volume increases further based on the *hw*.

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



Note: We don't know the performance of this algorithm in the low advantage regime.

Introduction

Warm Up

Modulus Switching

Coded-BKW

Swapping Error and Secret

A Different Embedding Approach

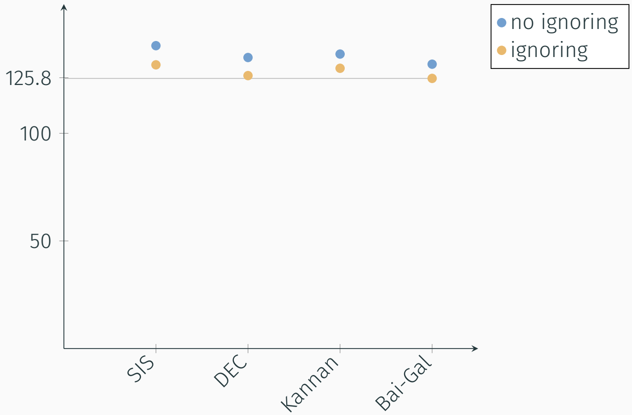
Exploiting Sparse Secrets

- All approaches so far tried to exploit small secrets. However, in our rolling example, the secret is sparse, i.e. most components are zero.
- In our example, the probability that a random coordinate is non-zero is $64/2048 = 1/32 \Rightarrow$ with probability $1 - 1/32$ a coordinate is zero.
- Ignoring k random components will ignore only non-zero components with probability

$$P_k = \prod_{i=0}^{k-1} \left(1 - \frac{64}{n-i}\right)$$

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

IGNORING COMPONENTS II



To summarise the results for our rolling example, we get:

- $\approx 2^{137.6}$ operations when ignoring small, sparse secret
- $\approx 2^{125.5}$ operations when exploiting small, sparse secret

Questions?