On the concrete hardness of Learning with Errors

Martin R. Albrecht @martinralbrecht
joint work with Rachel Player and Sam Scott

ACE Seminar, UCL. May 7, 2015

Information Security Group Royal Holloway, University of London

Outline

Outline

Lattice

A lattice is a discrete additive subgroup of $\mathbb{R}^m$.

Basis

Let $\mathbf{B} = \{\mathbf{b}_1, ..., \mathbf{b}_m\}$ be a set of m linearly independent vectors in $\mathbb{R}^m$.
Then $L(\mathbf{B}) = \{\sum_{i=1}^{m} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z}\}$ is the lattice generated by this basis.

Dual Lattice

Given a lattice $L(\mathbf{B}) \subset \mathbb{R}^m$, define its dual as $\{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{x}\mathbf{B} \in \mathbb{Z}^m\}$.

We'll only use scaled-by-q dual lattices, i.e. $\{\mathbf{x} \in \mathbb{Z}_q^m \mid \mathbf{x}\mathbf{B} \equiv \mathbf{0}\}$

The Learning with Errors (LWE) problem was defined by Oded Regev [Reg05].

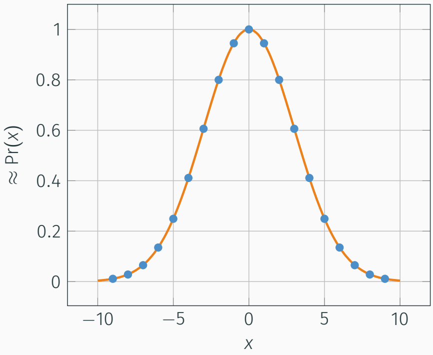
- Suppose a public matrix $\mathbf{A}$ and a secret vector $\mathbf{s}$.
- If we were also given $\mathbf{b} = \mathbf{A}\mathbf{s}$ we could compute $\mathbf{s}$ by linear algebra.
- Now imagine this is noisy: $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e}$ with $\mathbf{e}$ small.
- From $\mathbf{A}$ and $\mathbf{c}$ can we find $\mathbf{s}$? Was $\mathbf{c}$ even computed this way?

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

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

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

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 a discrete Gaussian with width

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

Search LWE

From samples $(\mathbf{A}, \mathbf{c})$ recover $\mathbf{s}$.

Decision LWE

Determine if samples $(\mathbf{A}, \mathbf{c})$ are LWE or uniformly random.

These problems are polynomial-time equivalent.

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.

Switching Moduli

Let $(\mathsf{a}, c) = (\mathsf{a}, \langle \mathsf{a}, \mathsf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ 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 s. If $p < q$ then

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

follows a distribution close to an LWE distribution with parameters $n, \sqrt{2}\alpha, p$.

Why care?

Learning With Errors

- is assumed to be a hard problem like discrete logarithms, factoring, etc.
- reduces to hard problems on lattices, such as GapSVP.
- is assumed to have resistance against quantum computers, unlike discrete logarithms and factoring.
- is remarkably versatile for constructing cryptographic schemes.

Identity-based encryption [GPV08]

Ciphertexts are of the form

$$(\mathbf{p}, c) = (\mathbf{A}\mathbf{s} + \mathbf{e}, \mathbf{u} \cdot \mathbf{s} + e + b \cdot \lfloor q/2 \rfloor)$$

where $H(id) = \mathbf{u} = \mathbf{x}^T\mathbf{A}$ is the public key for the private key $\mathbf{x}$.

Decryption is done by

$$c - \langle \mathbf{x}, \mathbf{p} \rangle = -\langle \mathbf{x}, \mathbf{e} \rangle + e + b \cdot \lfloor q/2 \rfloor.$$

Fully homomorphic encryption [BV11, AFFP11]

Think of LWE encryptions

$$(\mathbf{a}_i, c_i) = (\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i + b_i \cdot \lfloor q/2 \rfloor)$$

as noisy linear polynomials

$$-c_i + \sum a_{ij} x_j.$$

Add, multiply and decrypt as usual.

How hard is LWE?

Given n (and α, q) how many operations does it take to solve?

- Problem 1. Algorithms/attacks are not well understood in terms of concrete running times.
  - Runtimes are given asymptotically.
  - Algorithms are better in practice than the theoretical bounds.
  - Many heuristic assumptions.
- Problem 2. Many variables
  - dimension, modulus, secret size
  - distribution of the secret
  - number of samples available to an attacker
  - variants of the problem (e.g. small secrets, BinaryError-LWE)

What do people do currently?

Often, in the literature, the following assumptions were made when estimating concrete security of an LWE-based scheme:

- the best attack is a lattice-based distinguishing attack;
- BKZ runs in roughly the time given in [LP11];
- the use of a small secret makes no difference for attacks.

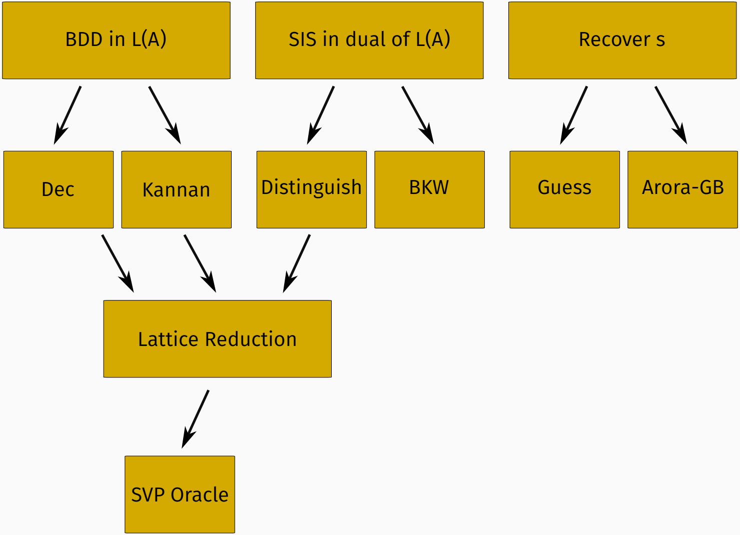
All three of these assumptions turn out not to be correct.

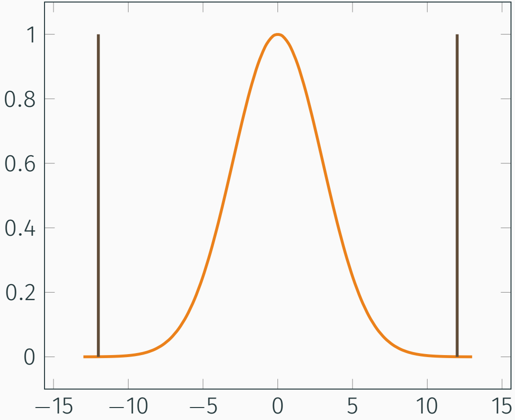
So, what did we do?

- Overview the strategies for attacking LWE.
- Analyse and present running times.
- Produce concrete estimates for attack timings for parameters sets.

The estimation code is available as a Sage module.

Strategies for Solving LWE





- The error is from a small subset of $\mathbb{Z}_q$, say, $(-\tau \cdot \sigma, \ldots, \tau \cdot \sigma)$
- Each candidate gives rise to one linear equation.
- Construct equations of degree $2\,\tau \cdot \sigma + 1$ encoding that one of these linear equations must hold.
- Solve the system using Gröbner bases.

Arora-GB

Arora-Ge (Linearisation) with $\sigma = \sqrt{n}$

$$\mathcal{O}\left(2^{\omega\, n \log(8n \log n) - \omega n \log n}\right)$$

Gröbner Bases with $\sigma = \sqrt{n}$

$$\mathcal{O}\left(2^{2.82\, \omega\, n}\right)$$

under some regularity assumption.

M.A., Carlos Cid, Jean-Charles Faugère, and Ludovic Perret.
Algebraic algorithms for LWE.
Cryptology ePrint Archive, Report 2014/1018, 2014.
`http://eprint.iacr.org/2014/1018`.

Short Integer Solutions (SIS)

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

$$\mathbf{y}\mathbf{B} \equiv \mathbf{0} \pmod{q}.$$

- Recall the dual lattice: $L^* = \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{x}\mathbf{B} \in \mathbb{Z}^m\}$.
- Then the scaled dual lattice, qL^* has the property that $\mathbf{x}\mathbf{B} \equiv 0 \pmod{q}$ for all $\mathbf{x} \in qL^*$.
- Therefore, a short vector of qL^* is equivalent to solving SIS on $\mathbf{B}$.

Strategy

- Find a short $\mathbf{y}$ solving SIS on $\mathbf{A}$.
- Given LWE samples $\mathbf{A}, \mathbf{c}$ where either $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e}$ or $\mathbf{c}$ uniformly random.
- Compute $\langle \mathbf{y}, \mathbf{c} \rangle$.
  - If $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e}$, then $\langle \mathbf{y}, \mathbf{c} \rangle = \langle \mathbf{y}\mathbf{A}, \mathbf{s} \rangle + \langle \mathbf{y}, \mathbf{e} \rangle \equiv \langle \mathbf{y}, \mathbf{e} \rangle \pmod{q}$.
  - If $\mathbf{c}$ is uniformly random, so is $\langle \mathbf{y}, \mathbf{c} \rangle$.
- If $\mathbf{y}$ is sufficiently short, since $\mathbf{e}$ is also small, then $\langle \mathbf{y}, \mathbf{e} \rangle$ will also be short, and can be distinguished from uniform values.

Distinguish (Lattice Reduction)

A reduced lattice basis is made of short vectors, in particular the first vector.

1. Construct a basis of the dual from the instance.
2. Feed to a lattice reduction algorithm to obtain short vectors $\mathbf{v}_i$.
3. Check if $\mathbf{v}_i\,\mathbf{A}$ are small.

Daniele Micciancio and Oded Regev.
Lattice-based cryptography.
In Bernstein et al. [BBD09], pages 147–191.

We revisit Gaussian elimination:

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

$$\stackrel{?}{=} \left(\begin{array}{c|c|cccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & \langle a_1, s\rangle + e_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & \langle a_2, s\rangle + e_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} & \langle a_m, s\rangle + e_m \end{array}\right)$$

$$\Rightarrow \left(\begin{array}{c|c|ccc|l} \mathbf{a}_{11} & \mathbf{a}_{12} & \mathbf{a}_{13} & \cdots & \mathbf{a}_{1n} & \langle \mathbf{a}_1, \mathbf{s}\rangle + \mathbf{e}_1 \\ 0 & \tilde{\mathbf{a}}_{22} & \tilde{\mathbf{a}}_{23} & \cdots & \tilde{\mathbf{a}}_{2n} & \langle \tilde{\mathbf{a}}_2, \mathbf{s}\rangle + \mathbf{e}_2 - \frac{\mathbf{a}_{21}}{\mathbf{a}_{11}}\mathbf{e}_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \\ 0 & \tilde{\mathbf{a}}_{m2} & \tilde{\mathbf{a}}_{m3} & \cdots & \tilde{\mathbf{a}}_{mn} & \langle \tilde{\mathbf{a}}_m, \mathbf{s}\rangle + \mathbf{e}_m - \frac{\mathbf{a}_{m1}}{\mathbf{a}_{11}}\mathbf{e}_1 \end{array}\right)$$

- $\frac{\mathbf{a}_{i1}}{\mathbf{a}_{11}}$ is essentially random in $\mathbb{Z}_q$ wiping all “smallness”.
- If $\frac{\mathbf{a}_{i1}}{\mathbf{a}_{11}}$ is 1 noise-size doubles because of the addition.

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

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

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

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

For each $\mathbf{z} \in \mathbb{Z}_q^b$ find row in $\mathbf{A}$ which contains $\mathbf{z}$ as a subvector at the target indices.

BKW Algorithm

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

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

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

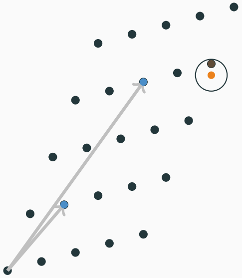
Time and memory complexity of $\mathcal{O}\left(2^{n/(2-1/c)}\right)$ for $q \approx n^c$.

M.A., Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret.
On the complexity of the BKW algorithm on LWE.
Designs, Codes and Cryptography, 74:325–354, 2015.

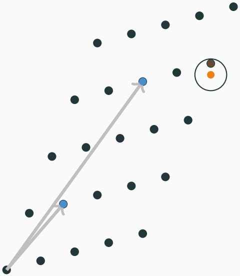
Alexandre Duc, Florian Tramèr, and Serge Vaudenay.
Better Algorithms for LWE and LWR.
In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 173–202. Springer, April 2015.

Bounded Distance Decoding (BDD)

Given a basis of L, a target vector $\mathbf{t}$, and a distance parameter $\beta > 0$ such that $d(\mathbf{t}, L) < \beta\lambda_1(L)$, find a $\mathbf{y} \in L$ such that $d(\mathbf{y}, \mathbf{t}) = d(L, \mathbf{t})$.



here, $\beta = 0.5$



- $\mathbf{b} = \mathbf{A}\mathbf{s}$ is a point in the lattice,
- $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$ is a perturbed point.
- Solve the BDD instance to recover $\mathbf{b}$.
- Recover $\mathbf{s}$ by linear algebra.

- Most basic is Babai's nearest planes.
- Lindner and Peikert: use multiple planes.
- Liu and Nguyen: use pruning strategy.
- No closed formula for runtime, can only be calculated numerically.

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

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

γ-uSVP

Given a lattice L s.t. $\lambda_2(L) > \gamma\lambda_1(L)$, find a shortest nonzero vector in L

1. Reduce BDD to uSVP via Kannan's embedding:
$$\mathbf{B} = \begin{pmatrix} \mathbf{A}^T & 0 \\ \mathbf{c} & t \end{pmatrix}$$
   where in practice $t = 1$.
2. Use lattice reduction to solve uSVP instance.

M.A., Robert Fitzpatrick, and Florian Göpfert.
On the efficacy of solving LWE by reduction to unique-SVP.
In Hyang-Sook Lee and Dong-Guk Han, editors, *ICISC 13*, volume 8565 of *LNCS*, pages 293–310. Springer, November 2014.

Small Secret LWE

- So far the secret vector was chosen as $\mathsf{s}_{(\mathsf{i})} \leftarrow \mathbb{Z}_q$.
- Some applications choose $\mathsf{s}_{(i)} \leftarrow \{-1, 0, 1\}$ or $\mathsf{s}_{(\mathsf{i})} \leftarrow \{0, 1\}$.
- This is for efficiency or to make certain operations possible (FHE).
- We call such an LWE instance a small secret instance.

In most algorithms, a small secret makes the instance easier.

- exhaustive search: check 2^n or 3^n elements rather than $(\alpha q)^n$.
- modulus switching: we can improve many algorithms by switching to a smaller modulus.

Small Secret BKW

Given LWE samples $\mathbf{A}, \mathbf{c}$:

- Recall that BKW finds short vectors $\mathbf{y}$ such that $\mathbf{y}\mathbf{A} = 0$.
- Instead, find short vectors $\mathbf{y}$ such that $\mathbf{y}\mathbf{A} = \mathbf{w}$ is small.
- Then $\langle \mathbf{y}, \mathbf{c} \rangle = \mathbf{y} \cdot \mathbf{A} \cdot \mathbf{s} + \langle \mathbf{y}, \mathbf{e} \rangle = \langle \mathbf{w}, \mathbf{s} \rangle + \langle \mathbf{y}, \mathbf{e} \rangle$.
- If $\mathbf{s}$ is small, so is $\langle \mathbf{w}, \mathbf{s} \rangle$.

M.A., J.-C. Faugère, R. Fitzpatrick, and L. Perret.
Lazy modulus switching for the BKW algorithm on LWE.
In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 429–445. Springer, March 2014.

- Embed LWE instance into different uSVP lattice.
- Exploits the difference between size of the secret and the size of the error by scaling.
- Has little effect for FHE case, because the noise is already very small, but dramatic effect for Regev's PKC parameters.

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

Theory [BLP+13]

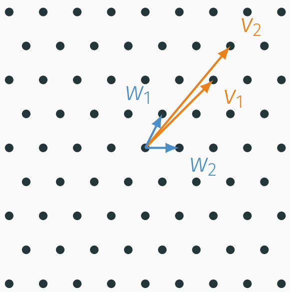
A small secret LWE instance as hard as standard LWE requires dimension $n \log q = \mathcal{O}(n \log n)$, for typical parameter choice $q = n^c$ for some small c.

Bai and Galbraith's Attack

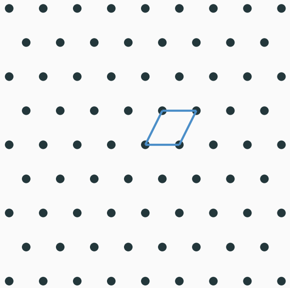
Dimension $n \log(\log n)$ may be sufficient.

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

- Above we made reference to lattice basis reduction algorithms.
- Examples of lattice reduction algorithms: LLL, BKZ, BKZ 2.0.
- These take as input a lattice basis and outputs a reduced basis:



- The success of a lattice reduction algorithm is characterised by the 'root-Hermite factor' δ_0.
- This is defined by $||\mathbf{b}_1|| = \delta_0^m \text{vol}\,(L)^{1/m}$.



BKZ

- Best known lattice reduction algorithm.
- BKZ is parametrised by blocksize k: bigger k mean better quality but more time.
- It can be seen as generalised LLL, which has $k = 2$.
- Literature disagrees on:
  - limiting value of δ_0 which BKZ can achieve (as a function of k);
  - runtime of BKZ (as a function of δ_0, k, or both).

We estimate BKZ as follows:

- Blocksize: Solve $\delta_0 \approx \left(\frac{k}{2\pi e}(\pi k)^{\frac{1}{k}}\right)^{\frac{1}{2(k-1)}}$ for k.
- CPU cycles for one SVP call in dimension k: $t_k = 2^{0.27k\,\log k - 1.02\,k + 16}$
- Required number of rounds: $\rho \approx \frac{n^2}{k^2}\log n$.
- Overall cost: $\rho \cdot t_k$.

Yuanmi Chen.
Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe.
PhD thesis, Paris 7, 2013.

Guillaume Hanrot, Xavier Pujol, and Damien Stehlé.
Analyzing blockwise lattice algorithms using dynamical systems.
In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 447–464. Springer, August 2011.

So many questions

- What is better for SIS: BKW or lattice reduction?
- What is the betst use for lattice reduction: Decoding, Kannan und solving SIS?
- Is there a best algorithm overall?
- What is the best small secret strategy?

So many questions

- What is better for SIS: BKW or lattice reduction?
- What is the betst use for lattice reduction: Decoding, Kannan und solving SIS?
- Is there a best algorithm overall?
- What is the best small secret strategy?

Short answer: it depends ....

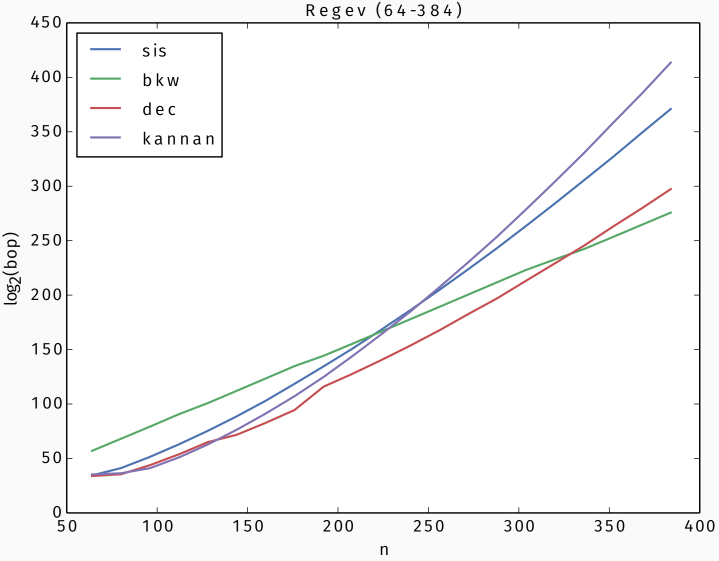
- For most algorithms, there is no sufficiently precise closed formula for runtime.
- We provide a Sage module for estimating how long various algorithms take to run.
- It takes as input parameters n, α, q.
- It outputs estimates of bit operations, memory requirements and number of calls to the LWE oracle.

We consider some 'typical' parameter sets.

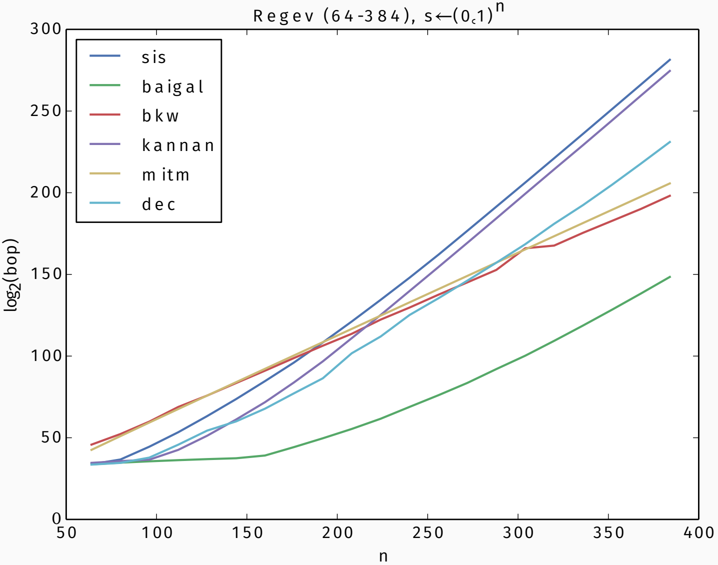
Regev These are Regev's example choices for parameters from [Reg09]. We use [AFC+13] to pick $q \approx n^2$ and $\alpha = 1/(\sqrt{2\pi n}\log_2^2 n)$.

FHE Given n and the multiplicative depth L we set $q = 2^{16.5 \cdot L + 5.4} \cdot 8^{2L-3} \cdot n^L$ and $\alpha = \sqrt{2\pi} \cdot 3.2/q$ inspired by parameters suggested in [GHS12].

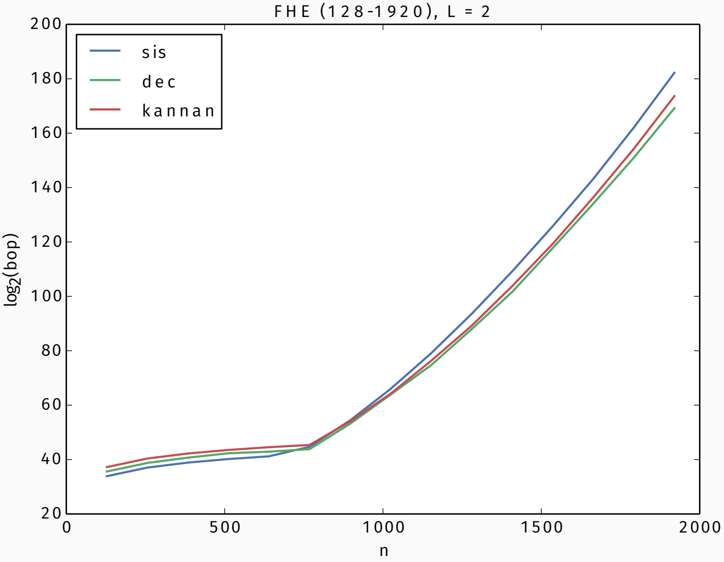
For small secrets we always assume $\mathbf{s}_{(i)} \in \{0, 1\}$.


Regev (64-384)
sis
bkw
dec
kannan
log2(bop)
n
0
50
100
150
200
250
300
350
400
450
50
100
150
200
250
300
350
400

Regev: Small Secret



FHE


FHE (128-1920), L = 2
sis
dec
kannan
log2(bop)
200
180
160
140
120
100
80
60
40
20
0
500
1000
1500
2000
n

So what?

Results

- No one algorithm always outperforms all others.
- Parameters are paramount.
- Small secrets matter.

Open Problems

- Is there an algorithm in $2^{\mathcal{O}(n)}$ time but less than $2^{\mathcal{O}(n)}$ memory?
- How long does lattice reduction actually take?
- Can we bridge the gap between theory and practice for small secrets?

Questions?

survey `http://eprint.iacr.org/2015/046`

estimator `https://bitbucket.org/malb/lwe-estimator`

M.A., Robert Fitzpatrick, Daniel Cabracas, Florian Göpfert, and Michael Schneider.
A generator for LWE and Ring-LWE instances, 2013.
available at `http://www.iacr.org/news/files/2013-04-29lwe-generator.pdf`.

M.A., Pooya Farshim, Jean-Charles Faugère, and Ludovic Perret.
Polly cracker, revisited.
In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 179–196. Springer, December 2011.

Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors.
Post-Quantum Cryptography.
Springer, 2009.

Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé.

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

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

Oded Regev.
On lattices, learning with errors, random linear codes, and cryptography.
J. ACM, 56(6), 2009.