An intro to lattices and learning with errors

A way to keep your secrets secret in a post-quantum world

Daniel Apon – Univ of Maryland

Some images in this talk authored by me
Many, excellent lattice images in this talk authored by Oded Regev
and available in papers and surveys on his personal website
http://www.cims.nyu.edu/~regev/ (as of Sept 29, 2012)

Introduction to LWE

1. Learning with Errors
   - Let $p = p(n) \leq poly(n)$. Consider the noisy linear equations:

$$\begin{aligned} \langle \mathbf{a}_1, \mathbf{s} \rangle &\approx_\chi b_1 \pmod{p} \\ \langle \mathbf{a}_2, \mathbf{s} \rangle &\approx_\chi b_2 \pmod{p} \\ &\vdots \end{aligned}$$

for $\mathbf{s} \in \mathbb{Z}_p^n$, $\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_p^n$, $b_i \in \mathbb{Z}_p$, and error $\chi : \mathbb{Z}_p \to \mathbb{R}^+$ on $\mathbb{Z}_p$.

1. Learning with Errors
   - Let $p = p(n) \leq poly(n)$. Consider the noisy linear equations:

$$\begin{aligned} \langle \mathbf{a}_1, \mathbf{s} \rangle &\approx_\chi b_1 \pmod{p} \\ \langle \mathbf{a}_2, \mathbf{s} \rangle &\approx_\chi b_2 \pmod{p} \\ &\vdots \end{aligned}$$

     for $\mathbf{s} \in \mathbb{Z}_p^n$, $\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_p^n$, $b_i \in \mathbb{Z}_p$, and error $\chi : \mathbb{Z}_p \to \mathbb{R}^+$ on $\mathbb{Z}_p$.
   - Goal: Recover $\mathbf{s}$.

Introduction to LWE

1. Learning with Errors
   - Let $p = p(n) \leq poly(n)$. Consider the noisy linear equations:

$$\begin{aligned} \langle \mathbf{a}_1, \mathbf{s} \rangle &\approx_\chi b_1 \pmod{p} \\ \langle \mathbf{a}_2, \mathbf{s} \rangle &\approx_\chi b_2 \pmod{p} \\ &\vdots \end{aligned}$$

     for $\mathbf{s} \in \mathbb{Z}_p^n$, $\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_p^n$, $b_i \in \mathbb{Z}_p$, and error $\chi : \mathbb{Z}_p \to \mathbb{R}^+$ on $\mathbb{Z}_p$.
   - Goal: Recover $\mathbf{s}$.
2. Why we care:
   - Believed hard for quantum algorithms
   - Average-case = worst-case
   - Many crypto applications!

Talk Overview. Up next: Intro to lattices

What's a lattice?

- A lattice is a discrete additive subgroup of $\mathbb{R}^n$

What's a lattice?

- A lattice is a discrete additive subgroup of $\mathbb{R}^n$
- A lattice is a set of points in n-dimensional space with a periodic structure

What's a lattice?

- A lattice is a discrete additive subgroup of $\mathbb{R}^n$
- A lattice is a set of points in n-dimensional space with a periodic structure
- Given n linearly independent vectors $\mathbf{v}_1, ..., \mathbf{v}_n \in \mathbb{R}^n$, the lattice they generate is the set of vectors

$$L(\mathbf{v}_1, ..., \mathbf{v}_n) \stackrel{\text{def}}{=} \left\{ \sum_{i=1}^{n} \alpha_i \mathbf{v}_i \;\middle|\; \alpha_i \in \mathbb{Z} \right\}.$$

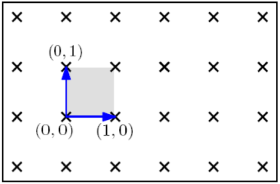
What's a lattice?

- A lattice is a discrete additive subgroup of $\mathbb{R}^n$
- A lattice is a set of points in n-dimensional space with a periodic structure
- Given n linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$, the lattice they generate is the set of vectors

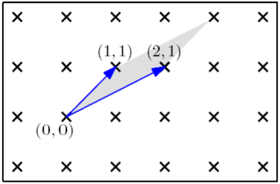
$$L(\mathbf{v}_1, \ldots, \mathbf{v}_n) \stackrel{\text{def}}{=} \left\{ \sum_{i=1}^{n} \alpha_i \mathbf{v}_i \;\middle|\; \alpha_i \in \mathbb{Z} \right\}.$$

- The basis $\mathbf{B} = \begin{pmatrix} | & | & \cdots & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & \cdots & | \end{pmatrix}$ generates the lattice $L(\mathbf{B})$.

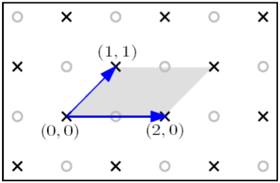
More on lattice bases



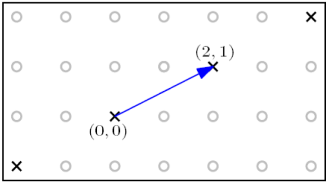
(a) A basis of $\mathbb{Z}^2$



(b) Another basis of $\mathbb{Z}^2$

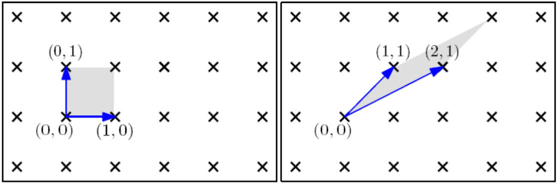


(c) Not a basis of $\mathbb{Z}^2$

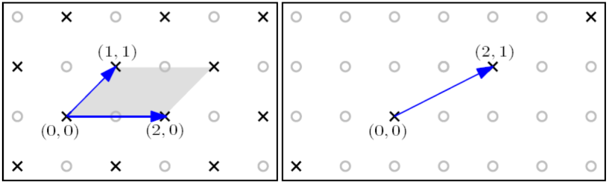


(d) Not a full-rank lattice

More on lattice bases



(a) A basis of $\mathbb{Z}^2$ (b) Another basis of $\mathbb{Z}^2$



(c) Not a basis of $\mathbb{Z}^2$ (d) Not a full-rank lattice

The gray-shaded region is the fundamental parallelepiped, given by $P(\mathbf{B}) = \{\mathbf{B}x \mid x \in [0,1)^n\}$.

More on the fundamental parallelepiped

Useful facts:

- For bases $\mathbf{B}_1, \mathbf{B}_2$, $L(\mathbf{B}_1) = L(\mathbf{B}_2) \Rightarrow \text{vol}(P(\mathbf{B}_1)) = \text{vol}(P(\mathbf{B}_2))$

More on the fundamental parallelepiped

Useful facts:

- For bases $\mathbf{B}_1, \mathbf{B}_2$, $L(\mathbf{B}_1) = L(\mathbf{B}_2) \Rightarrow \mathrm{vol}(P(\mathbf{B}_1)) = \mathrm{vol}(P(\mathbf{B}_2))$
- $\mathrm{vol}(\mathrm{P}(\mathbf{B})) = \det(\mathbf{B})$

More on the fundamental parallelepiped

Useful facts:

- For bases $\mathbf{B}_1, \mathbf{B}_2$, $L(\mathbf{B}_1) = L(\mathbf{B}_2) \Rightarrow \text{vol}(P(\mathbf{B}_1)) = \text{vol}(P(\mathbf{B}_2))$
- $\text{vol}(\text{P}(\mathbf{B})) = \det(\mathbf{B})$
- $\det(\mathbf{B}_1) = \det(\mathbf{B}_2)$ *iff* $\mathbf{B}_1 = \mathbf{B}_2\mathbf{U}$ for a unimodular matrix $\mathbf{U}$

More on the fundamental parallelepiped

Useful facts:

- For bases $\mathbf{B}_1, \mathbf{B}_2$, $L(\mathbf{B}_1) = L(\mathbf{B}_2) \Rightarrow \text{vol}(P(\mathbf{B}_1)) = \text{vol}(P(\mathbf{B}_2))$
- $\text{vol}(\text{P}(\mathbf{B})) = \det(\mathbf{B})$
- $\det(\mathbf{B}_1) = \det(\mathbf{B}_2)$ *iff* $\mathbf{B}_1 = \mathbf{B}_2\mathbf{U}$ for a unimodular matrix $\mathbf{U}$
- A matrix $\mathbf{U}$ is unimodular if it is integral and $\det(\mathbf{U}) = \pm 1$.

More on the fundamental parallelepiped

Useful facts:

- For bases $\mathbf{B}_1, \mathbf{B}_2$, $L(\mathbf{B}_1) = L(\mathbf{B}_2) \Rightarrow \text{vol}(P(\mathbf{B}_1)) = \text{vol}(P(\mathbf{B}_2))$
- $\text{vol}(\mathrm{P}(\mathbf{B})) = \det(\mathbf{B})$
- $\det(\mathbf{B}_1) = \det(\mathbf{B}_2)$ *iff* $\mathbf{B}_1 = \mathbf{B}_2\mathbf{U}$ for a unimodular matrix $\mathbf{U}$
- A matrix $\mathbf{U}$ is unimodular if it is integral and $\det(\mathbf{U}) = \pm 1$.

Moral of the story: All lattices have countably infinitely many bases, and given some fixed lattice, all of its possible bases are related by "volume-preserving" transformations.

The dual of a lattice

- Given a lattice $L = L(\mathbf{B})$, the dual lattice $L^* \stackrel{\text{def}}{=} L(\mathbf{B}^*)$ is generated by the dual basis $\mathbf{B}^*$; the unique basis s.t. $\mathbf{B}^T\mathbf{B}^* = \mathbf{I}$.

The dual of a lattice

- Given a lattice $L = L(\mathbf{B})$, the dual lattice $L^* \stackrel{\text{def}}{=} L(\mathbf{B}^*)$ is generated by the dual basis $\mathbf{B}^*$; the unique basis s.t. $\mathbf{B}^T\mathbf{B}^* = \mathbf{I}$.
- Equivalently, the dual of a lattice $L \in \mathbb{R}^n$ is given by

$$L^* = \left\{ \mathbf{y} \in \mathbb{R}^n \;\middle|\; \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}, \text{ for all } \mathbf{x} \in L \right\}.$$

The dual of a lattice

- Given a lattice $L = L(\mathbf{B})$, the dual lattice $L^* \stackrel{\text{def}}{=} L(\mathbf{B}^*)$ is generated by the dual basis $\mathbf{B}^*$; the unique basis s.t. $\mathbf{B}^T\mathbf{B}^* = \mathbf{I}$.
- Equivalently, the dual of a lattice $L \in \mathbb{R}^n$ is given by

$$L^* = \left\{ \mathbf{y} \in \mathbb{R}^n \;\middle|\; \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}, \text{ for all } \mathbf{x} \in L \right\}.$$

- Fact. For any $L = L(\mathbf{B}), L^* = L(\mathbf{B}^*)$,

$$|\mathrm{vol}(P(\mathbf{B}))| = \left| \frac{1}{\mathrm{vol}(P(\mathbf{B}^*))} \right|.$$

Lattice problems

Defn: $\lambda_1(L)$ is the length of the shortest nonzero vector in L

Lattice problems

Defn: $\lambda_1(L)$ is the length of the shortest nonzero vector in L

1. $\textsc{GapSVP}_\gamma$
   - $\textsc{Input}$: n-dimensional lattice L and a number $d > 0$
   - $\textsc{Output}$: YES if $\lambda_1(L) \leq d$; NO if $\lambda_1(L) > \gamma(n) \cdot d$

Lattice problems

Defn: $\lambda_1(L)$ is the length of the shortest nonzero vector in L

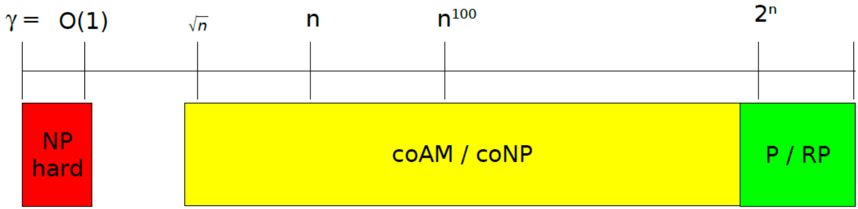
1. $\textsc{GapSVP}_\gamma$
   - INPUT: n-dimensional lattice L and a number $d > 0$
   - OUTPUT: YES if $\lambda_1(L) \leq d$; NO if $\lambda_1(L) > \gamma(n) \cdot d$
2. $\mathrm{CVP}_{L^*,d}$
   - INPUT: n-dimensional (dual) lattice L^* and a point $\mathbf{x} \in \mathbb{R}^n$ within distance d of L^*
   - OUTPUT: the closest vector in L^* to $\mathbf{x}$

Lattice problems

Defn: $\lambda_1(L)$ is the length of the shortest nonzero vector in L

1. $\textsc{GapSVP}_\gamma$
   - INPUT: n-dimensional lattice L and a number $d > 0$
   - OUTPUT: YES if $\lambda_1(L) \leq d$; NO if $\lambda_1(L) > \gamma(n) \cdot d$
2. $\mathrm{CVP}_{L^*,d}$
   - INPUT: n-dimensional (dual) lattice L^* and a point $\mathbf{x} \in \mathbb{R}^n$ within distance d of L^*
   - OUTPUT: the closest vector in L^* to $\mathbf{x}$
3. Other common lattice problems:
   - Shortest Independent Vectors Problem (SIVP), Covering Radius Problem (CRP), Bounded Distance Decoding (BDD), Discrete Gaussian Sampling Problem (DGS), Generalized Independent Vectors Problem (GIVP)

Complexity of (Gap)SVP and (Gap)CVP (and SIVP)



Moral of the story: We can get $\tilde{O}(2^n)$-approximate solutions in polynomial time. Constant-factor approximations are NP-hard. The best algorithms for anything in between require $\Omega(2^n)$ time.

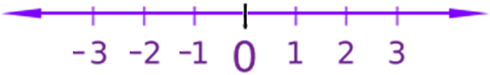
Talk Overview. Up next: Gaussians and lattices

1. Intro to lattices
2. Gaussians and lattices
   2.1 Uniformly sampling space
   2.2 $D_{L,r}$: The discrete Gaussian of width r on a lattice L
3. From lattices to learning
4. From learning to crypto

Uniformly sampling space

Question: How do you uniformly sample over an unbounded range?

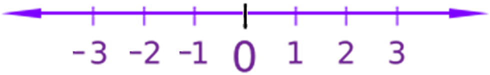
- Eg, how do you uniformly sample $x \in \mathbb{Z}$?



Uniformly sampling space

Question: How do you uniformly sample over an unbounded range?

- Eg, how do you uniformly sample $x \in \mathbb{Z}$?

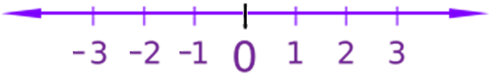


Answer: You can't!

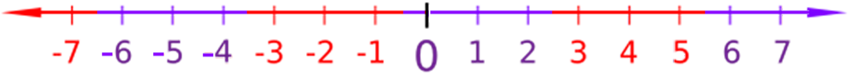
Uniformly sampling space

Question: How do you uniformly sample over an unbounded range?

- Eg, how do you uniformly sample $x \in \mathbb{Z}$?

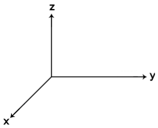


Answer: You can't!
The "lattice answer": Sample uniformly from $\mathbb{Z}_p$; view $\mathbb{Z}$ as being partitioned by copies of $\mathbb{Z}_p$



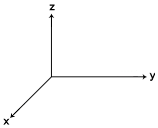
Uniformly sampling space

Question: How do you uniformly sample from $\mathbb{R}^n$?



Uniformly sampling space

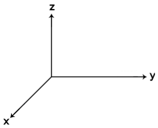
Question: How do you uniformly sample from $\mathbb{R}^n$?



Answer: You can't!

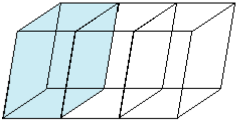
Uniformly sampling space

Question: How do you uniformly sample from $\mathbb{R}^n$?



Answer: You can't!

The "lattice answer": Sample uniformly from the fundamental parallelepiped of a lattice.

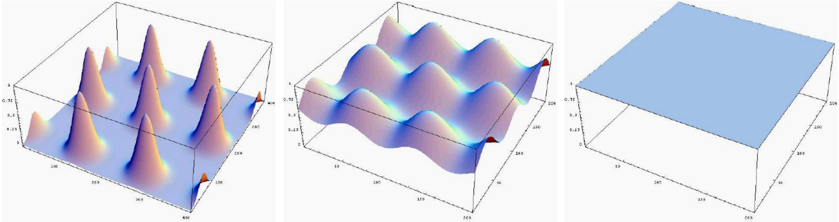


Lattices with Gaussian noise

A related question: What does a lattice look like when you "smudge" the lattice points with Gaussian-distributed noise?

Lattices with Gaussian noise

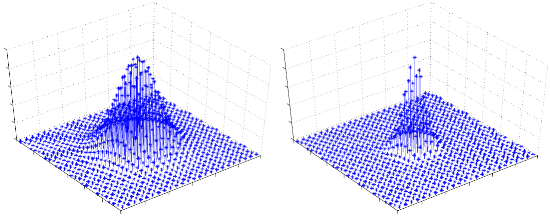
A related question: What does a lattice look like when you "smudge" the lattice points with Gaussian-distributed noise?
Answer: $\mathbb{R}^n$



- Left-to-right: PDFs of Gaussians centered at lattice points with increasing standard deviation

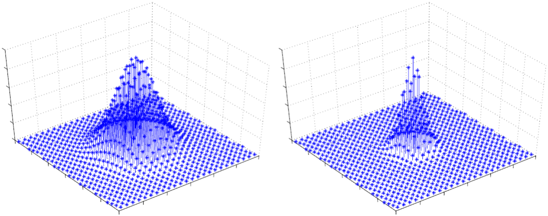
The discrete Gaussian: $D_{L,r}$

- Denote by $D_{L,r}$ the discrete Gaussian on a lattice L of width r



The discrete Gaussian: $D_{L,r}$

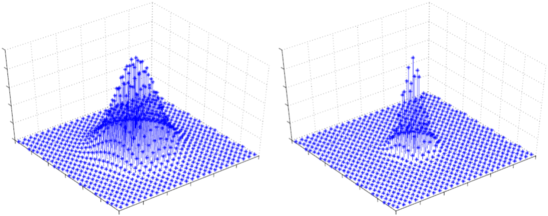
- Denote by $D_{L,r}$ the discrete Gaussian on a lattice L of width r



- Define the smoothing parameter, $\eta_\epsilon(L)$, as the least width s.t. $D_{L,r}$ is at most ϵ-far from the continuous Gaussian (over L).

The discrete Gaussian: $D_{L,r}$

- Denote by $D_{L,r}$ the discrete Gaussian on a lattice L of width r



- Define the smoothing parameter, $\eta_\epsilon(L)$, as the least width s.t. $D_{L,r}$ is at most ϵ-far from the continuous Gaussian (over L).
- Important fact. $\eta_{\mathrm{negl}(n)}(L) = \omega(\sqrt{\log n}) \approx \Theta(\sqrt{n})$

Talk Overview. Up next: From lattices to learning

1. Intro to lattices
2. Gaussians and lattices
3. From lattices to learning
   3.1 Reduction sketch: GapSVP to LWE
4. From learning to crypto

Reduction from GapSVP to LWE – Overview

Reduction sketch

1. Our goal: Prove LWE is hard
2. Reduction outline
   2.1 Why quantum?
3. Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ $\mathrm{CVP}_{L^*,\alpha p/r}$ oracle
4. Quantum step: $\mathrm{CVP}_{L^*,\alpha p/r}$ oracle $\rightarrow$ $D_{L,r\sqrt{n}/(\alpha p)}$
   4.1 Note: $(\eta_\epsilon(L) \approx)\ \alpha p > 2\sqrt{n} \rightarrow D_{L,r\sqrt{n}/(\alpha p)} \approx D_{L,<r/2}$
5. Conclude: Either LWE is hard, or the complexity landscape turns into a war zone
   5.1 "War zone:" At least 4 or 5 good complexity classes had to give their lives to ensure stability – that sort of thing.

Reduction outline: GapSVP to LWE

- LLL Basis Reduction algorithm: In polytime, given an arbitrary $L(\mathbf{B})$ outputs a new basis $\mathbf{B}'$ of length at most 2^n times the shortest basis.

GOAL: Given an arbitrary lattice L, output a very short vector, or decide none exist.

Reduction outline: GapSVP to LWE

- LLL Basis Reduction algorithm: In polytime, given an arbitrary $L(\mathbf{B})$ outputs a new basis $\mathbf{B}'$ of length at most 2^n times the shortest basis.

GOAL: Given an arbitrary lattice L, output a very short vector, or decide none exist.

- Let r_i denote $r \cdot (\alpha p/\sqrt{n})^i$ for $i = 3n, 3n-1, \ldots, 1$ and $r \geq O(n/\alpha)$. (Imagine $\alpha \approx 1/n^{1.5}$, so $r \approx n^{1.5} \cdot n$.)
- Using LLL, generate $\mathbf{B}'$, and using $\mathbf{B}'$, draw n^c samples from $D_{L,r_{3n}}$.

Reduction outline: GapSVP to LWE

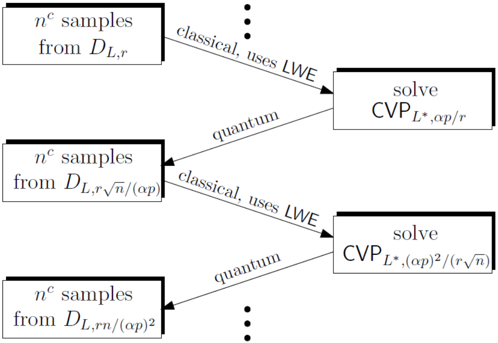
- LLL Basis Reduction algorithm: In polytime, given an arbitrary $L(\mathbf{B})$ outputs a new basis $\mathbf{B}'$ of length at most 2^n times the shortest basis.

GOAL: Given an arbitrary lattice L, output a very short vector, or decide none exist.

- Let r_i denote $r \cdot (\alpha p/\sqrt{n})^i$ for $i = 3n, 3n-1, ..., 1$ and $r \geq O(n/\alpha)$. (Imagine $\alpha \approx 1/n^{1.5}$, so $r \approx n^{1.5} \cdot n$.)
- Using LLL, generate $\mathbf{B}'$, and using $\mathbf{B}'$, draw n^c samples from $D_{L,r_{3n}}$.
- For $i = 3n, ...1$,
  - Call IterativeStep n^c times, using the n^c samples from D_{L,r_i} to produce 1 sample from $D_{L,r_{i-1}}$ each time.
- Output a sample from $D_{L,r_0} = D_{L,r}$.

The iterative step

Two steps: (1) classical, (2) quantum



Why quantum?

- Let L be a lattice. Let $d \ll \lambda_1(L)$.
- You are given an oracle $\mathcal{O}$ that, on input $\mathbf{x} \in \mathbb{R}^n$ within distance d from L, outputs the closest lattice vector to $\mathbf{x}$.
- (Caveat: If $\mathbf{x}$ of distance $> d$ from L, $\mathcal{O}$'s output is arbitrary.)
- How do you use $\mathcal{O}$?

Why quantum?

- Let L be a lattice. Let $d \ll \lambda_1(L)$.
- You are given an oracle $\mathcal{O}$ that, on input $\mathbf{x} \in \mathbb{R}^n$ within distance d from L, outputs the closest lattice vector to $\mathbf{x}$.
- (Caveat: If $\mathbf{x}$ of distance $> d$ from L, $\mathcal{O}$'s output is arbitrary.)
- How do you use $\mathcal{O}$?
- One idea: Choose some lattice vector $\mathbf{y} \in L$. Let $\mathbf{x} = \mathbf{y} + \mathbf{z}$ with $||\mathbf{z}|| \leq d$. Give $\mathbf{x}$ to $\mathcal{O}$.

Why quantum?

- Let L be a lattice. Let $d \ll \lambda_1(L)$.
- You are given an oracle $\mathcal{O}$ that, on input $\mathbf{x} \in \mathbb{R}^n$ within distance d from L, outputs the closest lattice vector to $\mathbf{x}$.
- (Caveat: If $\mathbf{x}$ of distance $> d$ from L, $\mathcal{O}$'s output is arbitrary.)
- How do you use $\mathcal{O}$?
- One idea: Choose some lattice vector $\mathbf{y} \in L$. Let $\mathbf{x} = \mathbf{y} + \mathbf{z}$ with $||\mathbf{z}|| \leq d$. Give $\mathbf{x}$ to $\mathcal{O}$.
- But then $\mathcal{O}(\mathbf{x}) = \mathbf{y}$!

Why quantum?

- Let L be a lattice. Let $d \ll \lambda_1(L)$.
- You are given an oracle $\mathcal{O}$ that, on input $\mathbf{x} \in \mathbb{R}^n$ within distance d from L, outputs the closest lattice vector to $\mathbf{x}$.
- (Caveat: If $\mathbf{x}$ of distance $> d$ from L, $\mathcal{O}$'s output is arbitrary.)
- How do you use $\mathcal{O}$?
- One idea: Choose some lattice vector $\mathbf{y} \in L$. Let $\mathbf{x} = \mathbf{y} + \mathbf{z}$ with $||\mathbf{z}|| \leq d$. Give $\mathbf{x}$ to $\mathcal{O}$.
- But then $\mathcal{O}(\mathbf{x}) = \mathbf{y}$!
- But quantumly, knowing how to compute $\mathbf{y}$ given only $\mathbf{y} + \mathbf{z}$ is useful – it allows us to uncompute a register containing $\mathbf{y}$.

Classical step: $D_{L,r}$ + LWE oracle $\to$ CVP$_{L^*,\alpha p/r}$ oracle

- Let D be a probability distribution on a lattice L. Consider the Fourier transform $f : \mathbb{R}^n \to \mathbb{C}$, given by

$$f(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{\mathbf{y} \in L} D(\mathbf{y}) exp(2\pi i \langle \mathbf{x}, \mathbf{y} \rangle) = \mathbb{E}_{\mathbf{y} \leftarrow D}[exp(2\pi i \langle \mathbf{x}, \mathbf{y} \rangle)]$$

Classical step: $D_{L,r}$ + LWE oracle $\to$ CVP$_{L^*,\alpha p/r}$ oracle

- Let D be a probability distribution on a lattice L. Consider the Fourier transform $f : \mathbb{R}^n \to \mathbb{C}$, given by

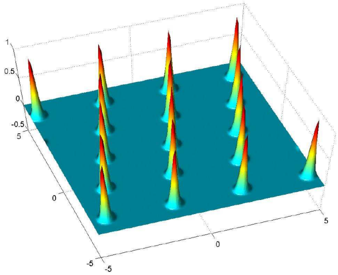
$$f(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{\mathbf{y}\in L} D(\mathbf{y}) exp(2\pi i \langle \mathbf{x}, \mathbf{y} \rangle) = \mathbb{E}_{\mathbf{y}\leftarrow D}[exp(2\pi i \langle \mathbf{x}, \mathbf{y} \rangle)]$$

- Using Hoeffding's inequality, if $\mathbf{y}_1, ..., \mathbf{y}_N$ are $N = poly(n)$ independent samples from D, then w.h.p.

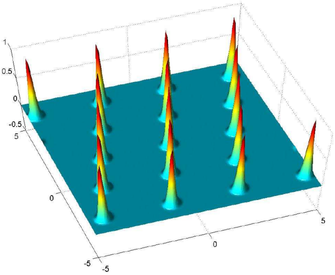
$$f(\mathbf{x}) \approx \frac{1}{N} \sum_{j=1}^{N} exp(2\pi i \langle \mathbf{x}, \mathbf{y}_j \rangle)$$

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ $\mathrm{CVP}_{L^*,\alpha p/r}$ oracle

- Applying this idea to $D_{L,r}$, we get a good approximation of its Fourier transform, denoted $f_{1/r}$. Note $f_{1/r}$ is L^*-periodic.



- Applying this idea to $D_{L,r}$, we get a good approximation of its Fourier transform, denoted $f_{1/r}$. Note $f_{1/r}$ is L^*-periodic.



- It can be shown that $1/r \ll \lambda_1(L^*)$, so we have

$$f_{1/r}(\mathbf{x}) \approx exp(-\pi(r \cdot \mathrm{dist}(L^*, \mathbf{x}))^2)$$

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ CVP$_{L^*,\alpha p/r}$ oracle

- Attempt #1: Using samples from $D_{L,r}$, we repeatedly compute approximations to $f_{1/r}$ and attempt to "walk uphill" to find the peak (a dual lattice point).

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ CVP$_{L^*,\alpha p/r}$ oracle

- Attempt #1: Using samples from $D_{L,r}$, we repeatedly compute approximations to $f_{1/r}$ and attempt to "walk uphill" to find the peak (a dual lattice point).
- The problem: This procedure only gives a method to solve CVP$_{L^*,1/r}$. (Beyond that distance, the value of $f_{1/r}$ becomes negligible.)
- Plugging this into our iterative step means we go from $D_{L,r}$ to $D_{L,r\sqrt{n}}$, which is the wrong direction!

Classical step: $D_{L,r}$ + LWE oracle $\to$ CVP$_{L^*,\alpha p/r}$ oracle

- Attempt #1: Using samples from $D_{L,r}$, we repeatedly compute approximations to $f_{1/r}$ and attempt to "walk uphill" to find the peak (a dual lattice point).
- The problem: This procedure only gives a method to solve CVP$_{L^*,1/r}$. (Beyond that distance, the value of $f_{1/r}$ becomes negligible.)
- Plugging this into our iterative step means we go from $D_{L,r}$ to $D_{L,r\sqrt{n}}$, which is the wrong direction!
- Goal: We need a FATTER Fourier transform!

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ CVP$_{L^*,\alpha p/r}$ oracle

- Equivalently, we need tighter samples!
- Attempt #2: Take samples from $D_{L,r}$ and just divide every coordinate by p. This gives samples from $D_{L/p,r/p}$, where L/p is L scaled down by a factor of p.

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ CVP$_{L^*,\alpha p/r}$ oracle

- Equivalently, we need tighter samples!
- Attempt #2: Take samples from $D_{L,r}$ and just divide every coordinate by p. This gives samples from $D_{L/p,r/p}$, where L/p is L scaled down by a factor of p.
- That is, the lattice L/p consists of p^n translates of L.
  - Label these p^n translates by vectors from $\mathbb{Z}_p^n$.

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ $\text{CVP}_{L^*,\alpha p/r}$ oracle

- Equivalently, we need tighter samples!
- Attempt #2: Take samples from $D_{L,r}$ and just divide every coordinate by p. This gives samples from $D_{L/p,r/p}$, where L/p is L scaled down by a factor of p.
- That is, the lattice L/p consists of p^n translates of L.
  - Label these p^n translates by vectors from $\mathbb{Z}_p^n$.
- It can be shown that $r/p > \eta_\epsilon(L)$, which implies $D_{L/p,r/p}$ is uniform over the set of $L + L\mathbf{a}/p$, for $\mathbf{a} \in \mathbb{Z}_p^n$
  - For any choice of $\mathbf{a} \in \mathbb{Z}_p^n$, $L + L\mathbf{a}/p$ (modulo the parallelepiped) corresponds to a choice of translate

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ CVP$_{L^*,\alpha p/r}$ oracle

- This motivates defining a new distribution, $\tilde{D}$ with samples $(\mathbf{a}, \mathbf{y})$ obtained by:
  1. $\mathbf{y} \leftarrow D_{L/p,r/p}$
  2. $\mathbf{a} \in \mathbb{Z}_p^n$ s.t. $\mathbf{y} \in L + L\mathbf{a}/p$ ($\leftarrow$ Complicated to analyze..?)

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ CVP$_{L^*,\alpha p/r}$ oracle

- This motivates defining a new distribution, $\tilde{D}$ with samples $(\mathbf{a}, \mathbf{y})$ obtained by:
  1. $\mathbf{y} \leftarrow D_{L/p,r/p}$
  2. $\mathbf{a} \in \mathbb{Z}_p^n$ s.t. $\mathbf{y} \in L + L\mathbf{a}/p$ ($\leftarrow$ Complicated to analyze..?)
- From the previous slide, we know that we can obtain $\tilde{D}$ from $D_{L,r}$.

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ $\mathrm{CVP}_{L^*,\alpha p/r}$ oracle

- This motivates defining a new distribution, $\tilde{D}$ with samples $(\mathbf{a}, \mathbf{y})$ obtained by:
  1. $\mathbf{y} \leftarrow D_{L/p,r/p}$
  2. $\mathbf{a} \in \mathbb{Z}_p^n$ s.t. $\mathbf{y} \in L + L\mathbf{a}/p$ ($\leftarrow$ Complicated to analyze..?)
- From the previous slide, we know that we can obtain $\tilde{D}$ from $D_{L,r}$.

  Also, we know that $\tilde{D}$ is equivalently obtained by:
  1. First, $\mathbf{a} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^n$ ($\leftarrow$ Ahh! Much nicer. :))
  2. Then, $\mathbf{y} \leftarrow D_{L+L\mathbf{a}/p,r/p}$

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ $\text{CVP}_{L^*,\alpha p/r}$ oracle

- This motivates defining a new distribution, $\tilde{D}$ with samples $(\mathbf{a}, \mathbf{y})$ obtained by:
  1. $\mathbf{y} \leftarrow D_{L/p,r/p}$
  2. $\mathbf{a} \in \mathbb{Z}_p^n$ s.t. $\mathbf{y} \in L + L\mathbf{a}/p$ ($\leftarrow$ Complicated to analyze..?)
- From the previous slide, we know that we can obtain $\tilde{D}$ from $D_{L,r}$.

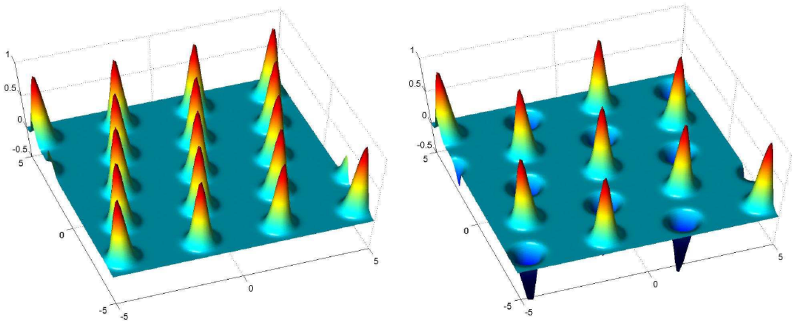
  Also, we know that $\tilde{D}$ is equivalently obtained by:
  1. First, $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_p^n$ ($\leftarrow$ Ahh! Much nicer. :))
  2. Then, $\mathbf{y} \leftarrow D_{L+L\mathbf{a}/p,r/p}$
- The width of the discrete Gaussian samples $\mathbf{y}$ is tighter now!..

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ CVP$_{L^*,\alpha p/r}$ oracle

How about the Fourier transform of $\tilde{D}$? It's wider now! But...

Classical step: $D_{L,r}$ + LWE oracle $\to$ CVP$_{L^*,\alpha p/r}$ oracle

How about the Fourier transform of $\tilde{D}$? It's wider now! But...
The problem: Each hill of $f_{p/r}$ now has its own "phase." Do we climb up or down?



- Two examples of the Fourier transform of $D_{L+L\mathbf{a}/p,r/p}$ with $\mathbf{a}$=(0,0) (left) and $\mathbf{a}$=(1,1) (right).

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ CVP$_{L^*,\alpha p/r}$ oracle

Key observation #1:

- For $\mathbf{x} \in L^*$, each sample $(\mathbf{a}, \mathbf{y}) \leftarrow \tilde{D}$ gives a linear equation

$$\langle \mathbf{a}, \tau(\mathbf{x}) \rangle = p\langle \mathbf{x}, \mathbf{y} \rangle \bmod p$$

  for $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_p^n$. After about n equations, we can use Gaussian elimination to recover $\tau(\mathbf{x}) \in \mathbb{Z}_p^n$.
- What if $\mathbf{x} \notin L^*$?

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ CVP$_{L^*,\alpha p/r}$ oracle

Key observation #2:

- For $\mathbf{x}$ close to L^*, each sample $(\mathbf{a}, \mathbf{y}) \leftarrow \tilde{D}$ gives a linear equation with error

$$\langle \mathbf{a}, \tau(\mathbf{x}) \rangle \approx \lfloor p \langle \mathbf{x}, \mathbf{y} \rangle \rceil \bmod p$$

for $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_p^n$. After $poly(n)$ equations, we use the LWE oracle to recover $\tau(\mathbf{x}) \in \mathbb{Z}_p^n$. (NOTE: $|error| = ||\tau(\mathbf{x})||_2$)

Classical step: $D_{L,r}$ + LWE oracle $\rightarrow$ $\text{CVP}_{L^*,\alpha p/r}$ oracle

Key observation #2:

- For $\mathbf{x}$ close to L^*, each sample $(\mathbf{a}, \mathbf{y}) \leftarrow \tilde{D}$ gives a linear equation with error

$$\langle \mathbf{a}, \tau(\mathbf{x}) \rangle \approx \lfloor p \langle \mathbf{x}, \mathbf{y} \rangle \rceil \bmod p$$

  for $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_p^n$. After $poly(n)$ equations, we use the LWE oracle to recover $\tau(\mathbf{x}) \in \mathbb{Z}_p^n$. (NOTE: $|error| = ||\tau(\mathbf{x})||_2$)
- This lets us compute the phase $exp(2\pi i \langle \mathbf{a}, \tau(\mathbf{x}) \rangle / p)$, and hence recover the closest dual lattice vector to $\mathbf{x}$.
- Classical step DONE.

Quantum step: $\mathrm{CVP}_{L^*,\alpha p/r}$ oracle $\rightarrow D_{L,r\sqrt{n}/(\alpha p)}$

Observe: $\mathrm{CVP}_{L^*,\alpha p/r} \rightarrow D_{L,r\sqrt{n}/(\alpha p)} = \mathrm{CVP}_{L^*,\sqrt{n}/r} \rightarrow D_{L,r}$

New quantum step: $\mathrm{CVP}_{L^*,\sqrt{n}/r} oracle \rightarrow D_{L,r}$

Ok, let's give a solution for $\mathrm{CVP}_{L^*,\sqrt{n}/r} \rightarrow D_{L,r}$.

New quantum step: $\mathrm{CVP}_{L^*,\sqrt{n}/r}$ *oracle* $\rightarrow D_{L,r}$

Ok, let's give a solution for $\mathrm{CVP}_{L^*,\sqrt{n}/r} \rightarrow D_{L,r}$.

GOAL: Get a quantum state corresponding to $f_{1/r}$ (on the dual lattice) and use the quantum Fourier transform to get $D_{L,r}$ (on the primal lattice). We will use the promised CVP oracle to do so.

New quantum step: $\text{CVP}_{L^*,\sqrt{n}/r}$ *oracle* $\rightarrow D_{L,r}$

Ok, let's give a solution for $\text{CVP}_{L^*,\sqrt{n}/r} \rightarrow D_{L,r}$.

GOAL: Get a quantum state corresponding to $f_{1/r}$ (on the dual lattice) and use the quantum Fourier transform to get $D_{L,r}$ (on the primal lattice). We will use the promised CVP oracle to do so.

1. Create a uniform superposition on L^*: $\sum_{\mathbf{x} \in L^*} |\mathbf{x}\rangle$.

New quantum step: $\text{CVP}_{L^*,\sqrt{n}/r}$ *oracle* $\rightarrow D_{L,r}$

Ok, let's give a solution for $\text{CVP}_{L^*,\sqrt{n}/r} \rightarrow D_{L,r}$.

GOAL: Get a quantum state corresponding to $f_{1/r}$ (on the dual lattice) and use the quantum Fourier transform to get $D_{L,r}$ (on the primal lattice). We will use the promised CVP oracle to do so.

1. Create a uniform superposition on L^*: $\sum_{\mathbf{x} \in L^*} |\mathbf{x}\rangle$.
2. On a separate register, create a "Gaussian state" of width $1/r$: $\sum_{\mathbf{z} \in \mathbb{R}^n} exp(-\pi ||r\mathbf{z}||^2)|\mathbf{z}\rangle$.

New quantum step: $\mathrm{CVP}_{L^*,\sqrt{n}/r}$ *oracle* $\rightarrow D_{L,r}$

Ok, let's give a solution for $\mathrm{CVP}_{L^*,\sqrt{n}/r} \rightarrow D_{L,r}$.

GOAL: Get a quantum state corresponding to $f_{1/r}$ (on the dual lattice) and use the quantum Fourier transform to get $D_{L,r}$ (on the primal lattice). We will use the promised CVP oracle to do so.

1. Create a uniform superposition on L^*: $\sum_{\mathbf{x} \in L^*} |\mathbf{x}\rangle$.
2. On a separate register, create a "Gaussian state" of width $1/r$: $\sum_{\mathbf{z} \in \mathbb{R}^n} exp(-\pi||r\mathbf{z}||^2)|\mathbf{z}\rangle$.
3. The combined system state is written:

$$\sum_{\mathbf{x} \in L^*, \mathbf{z} \in \mathbb{R}^n} exp(-\pi||r\mathbf{z}||^2)|\mathbf{x}, \mathbf{z}\rangle.$$

New quantum step: $\mathrm{CVP}_{L^*,\sqrt{n}/r} oracle \rightarrow D_{L,r}$

Key rule: All quantum computations must be reversible.

New quantum step: $\mathrm{CVP}_{L^*,\sqrt{n}/r}$*oracle* $\rightarrow D_{L,r}$

Key rule: All quantum computations must be reversible.

1. Add the first register to the second (reversible) to obtain: $\sum_{\mathbf{x}\in L^*,\mathbf{z}\in\mathbb{R}^n} exp(-\pi||r\mathbf{z}||^2)|\mathbf{x},\mathbf{x}+\mathbf{z}\rangle$.

New quantum step: $\mathrm{CVP}_{L^*,\sqrt{n}/r}$ *oracle* $\rightarrow D_{L,r}$

Key rule: All quantum computations must be reversible.

1. Add the first register to the second (reversible) to obtain: $\sum_{\mathbf{x}\in L^*,\mathbf{z}\in\mathbb{R}^n} exp(-\pi||r\mathbf{z}||^2)|\mathbf{x},\mathbf{x}+\mathbf{z}\rangle$.
2. Since we have a $\mathrm{CVP}_{L^*,\sqrt{n}/r}$ oracle we can compute $\mathbf{x}$ from $\mathbf{x}+\mathbf{z}$. Therefore, we can uncompute the first register:

$$\sum_{\mathbf{x}\in L^*,\mathbf{z}\in\mathbb{R}^n} exp(-\pi||r\mathbf{z}||^2)|\mathbf{x}+\mathbf{z}\rangle \approx \sum_{\mathbf{z}\in\mathbb{R}^n} f_{1/r}(\mathbf{z})|\mathbf{z}\rangle.$$

New quantum step: $\text{CVP}_{L^*,\sqrt{n}/r}$ *oracle* $\rightarrow D_{L,r}$

Key rule: All quantum computations must be reversible.

1. Add the first register to the second (reversible) to obtain: $\sum_{\mathbf{x}\in L^*,\mathbf{z}\in\mathbb{R}^n} exp(-\pi||r\mathbf{z}||^2)|\mathbf{x},\mathbf{x}+\mathbf{z}\rangle$.
2. Since we have a $\text{CVP}_{L^*,\sqrt{n}/r}$ oracle we can compute $\mathbf{x}$ from $\mathbf{x}+\mathbf{z}$. Therefore, we can uncompute the first register:

$$\sum_{\mathbf{x}\in L^*,\mathbf{z}\in\mathbb{R}^n} exp(-\pi||r\mathbf{z}||^2)|\mathbf{x}+\mathbf{z}\rangle \approx \sum_{\mathbf{z}\in\mathbb{R}^n} f_{1/r}(\mathbf{z})|\mathbf{z}\rangle.$$

3. Finally, apply the quantum Fourier transform to obtain

$$\sum_{\mathbf{y}\in L} D_{L,r}(\mathbf{y})|\mathbf{y}\rangle,$$

and measure it to obtain a sample from $\approx D_{L,r}$.

Talk Overview. Up next: From learning to crypto

1. Intro to lattices
2. Gaussians and lattices
3. From lattices to learning
4. From learning to crypto
   4.1 Regev's PKE scheme from LWE

Decisional Learning with Errors (DLWE)

- For positive integers n and $q \geq 2$, a secret $\mathbf{s} \in \mathbb{Z}_q^n$, and a distribution χ on $\mathbb{Z}$, define $A_{\mathbf{s},\chi}$ as the distribution obtained by drawing $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ uniformly at random and a noise term $e \xleftarrow{\$} \chi$, and outputting $(\mathbf{a}, b) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} + e \rangle \pmod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Decisional Learning with Errors (DLWE)

- For positive integers n and $q \geq 2$, a secret $\mathbf{s} \in \mathbb{Z}_q^n$, and a distribution χ on $\mathbb{Z}$, define $A_{\mathbf{s},\chi}$ as the distribution obtained by drawing $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ uniformly at random and a noise term $e \xleftarrow{\$} \chi$, and outputting $(\mathbf{a}, b) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} + e\rangle \pmod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.
- ($\mathrm{DLWE}_{n,q,\chi}$). An adversary gets oracle access to *either* $A_{\mathbf{s},\chi}$ or $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ and aims to distinguish (with non-negligible advantage) which is the case.

Decisional Learning with Errors (DLWE)

- For positive integers n and $q \geq 2$, a secret $\mathbf{s} \in \mathbb{Z}_q^n$, and a distribution χ on $\mathbb{Z}$, define $A_{\mathbf{s},\chi}$ as the distribution obtained by drawing $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ uniformly at random and a noise term $e \xleftarrow{\$} \chi$, and outputting $(\mathbf{a}, b) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} + e \rangle \pmod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.
- ($\mathsf{DLWE}_{n,q,\chi}$). An adversary gets oracle access to *either* $A_{\mathbf{s},\chi}$ or $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ and aims to distinguish (with non-negligible advantage) which is the case.
- **Theorem**. Let $B \geq \omega(\log n) \cdot \sqrt{n}$. There exists an efficiently sampleable distribution χ with $|\chi| < B$ (meaning, χ is supported only on $[-B, B]$) s.t. if an efficient algorithm solves the average-case $\mathsf{DLWE}_{n,q,\chi}$ problem, then there is an efficient quantum algorithm that solves $\mathsf{GapSVP}_{\tilde{O}(n \cdot q/B)}$ on any n-dimensional lattice.

Regev's PKE scheme

1. SecretKeyGen(1^n): Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$. Output $sk = \mathbf{s}$.

Regev's PKE scheme

1. SecretKeyGen(1^n): Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$. Output $sk = \mathbf{s}$.
2. PublicKeyGen($\mathbf{s}$): Let $N \stackrel{\text{def}}{=} O(n \log q)$. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{N \times n}$ and $\mathbf{e} \xleftarrow{\$} \chi^N$. Compute $\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \pmod q$, and define

$$\mathbf{P} \stackrel{\text{def}}{=} [\mathbf{b} || - \mathbf{A}] \in \mathbb{Z}_q^{N \times (n+1)}.$$

   Output $pk = \mathbf{P}$.

Regev's PKE scheme

1. SecretKeyGen(1^n): Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$. Output $sk = \mathbf{s}$.
2. PublicKeyGen($\mathbf{s}$): Let $N \stackrel{\text{def}}{=} O(n \log q)$. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{N \times n}$ and $\mathbf{e} \xleftarrow{\$} \chi^N$. Compute $\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \pmod q$, and define
$$\mathbf{P} \stackrel{\text{def}}{=} [\mathbf{b} || -\mathbf{A}] \in \mathbb{Z}_q^{N \times (n+1)}.$$
Output $pk = \mathbf{P}$.
3. $\text{Enc}_{pk}(m)$: To encrypt a message $m \in \{0,1\}$ using $pk = \mathbf{P}$, sample $\mathbf{r} \in \{0,1\}^N$ and output the ciphertext
$$\mathbf{c} = \mathbf{P}^T \cdot \mathbf{r} + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{m} \bmod q \in \mathbb{Z}_q^{n+1},$$
where $\mathbf{m} \stackrel{\text{def}}{=} (m, 0, \ldots, 0) \in \{0,1\}^{n+1}$.

Regev's PKE scheme

1. SecretKeyGen(1^n): Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$. Output $sk = \mathbf{s}$.
2. PublicKeyGen($\mathbf{s}$): Let $N \stackrel{\text{def}}{=} O(n \log q)$. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{N \times n}$ and $\mathbf{e} \xleftarrow{\$} \chi^N$. Compute $\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \pmod q$, and define
$$\mathbf{P} \stackrel{\text{def}}{=} [\mathbf{b} || - \mathbf{A}] \in \mathbb{Z}_q^{N \times (n+1)}.$$
Output $pk = \mathbf{P}$.
3. $\text{Enc}_{pk}(m)$: To encrypt a message $m \in \{0, 1\}$ using $pk = \mathbf{P}$, sample $\mathbf{r} \in \{0, 1\}^N$ and output the ciphertext
$$\mathbf{c} = \mathbf{P}^T \cdot \mathbf{r} + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{m} \bmod q \in \mathbb{Z}_q^{n+1},$$
where $\mathbf{m} \stackrel{\text{def}}{=} (m, 0, ..., 0) \in \{0, 1\}^{n+1}$.
4. $\text{Dec}_{sk}(\mathbf{c})$: To decrypt $\mathbf{c} \in \mathbb{Z}_q^{n+1}$ using secret key $sk = \mathbf{s}$, compute
$$m = \left\lfloor \frac{2}{q} (\langle \mathbf{c}, (1, \mathbf{s}) \rangle \bmod q) \right\rceil \bmod 2.$$

Regev's PKE scheme: Correctness

Encryption noise. Let all parameters be as before. Then for some e where $|e| \leq N \cdot B$, $\langle \mathbf{c}, (1, \mathbf{s}) \rangle = \left\lfloor \frac{q}{2} \right\rfloor \cdot m + e \pmod{q}$.

Regev's PKE scheme: Correctness

Encryption noise. Let all parameters be as before. Then for some e where $|e| \leq N \cdot B$, $\langle \mathbf{c}, (1, \mathbf{s}) \rangle = \left\lfloor \frac{q}{2} \right\rfloor \cdot m + e \pmod{q}$.

Proof.

$$\begin{aligned}
\langle \mathbf{c}, (1, \mathbf{s}) \rangle &= \left\langle \mathbf{P}^T \cdot \mathbf{r} + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{m}, (1, \mathbf{s}) \right\rangle \pmod{q} \\
&= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \mathbf{r}^T \mathbf{P} \cdot (1, \mathbf{s}) \pmod{q} \\
&= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \mathbf{r}^T \mathbf{b} - \mathbf{r}^T \mathbf{A} \mathbf{s} \pmod{q} \\
&= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \langle \mathbf{r}, \mathbf{e} \rangle \pmod{q},
\end{aligned}$$

and $|\langle \mathbf{r}, \mathbf{e} \rangle| \leq ||\mathbf{r}||_1 \cdot ||\mathbf{e}||_\infty = N \cdot B$.

Regev's PKE scheme: Correctness

Encryption noise. Let all parameters be as before. Then for some e where $|e| \leq N \cdot B$, $\langle \mathbf{c}, (1, \mathbf{s}) \rangle = \lfloor \frac{q}{2} \rfloor \cdot m + e \pmod{q}$.

Proof.

$$\begin{aligned}
\langle \mathbf{c}, (1, \mathbf{s}) \rangle &= \left\langle \mathbf{P}^T \cdot \mathbf{r} + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{m}, (1, \mathbf{s}) \right\rangle \pmod{q} \\
&= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \mathbf{r}^T \mathbf{P} \cdot (1, \mathbf{s}) \pmod{q} \\
&= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \mathbf{r}^T \mathbf{b} - \mathbf{r}^T \mathbf{A} \mathbf{s} \pmod{q} \\
&= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \langle \mathbf{r}, \mathbf{e} \rangle \pmod{q},
\end{aligned}$$

and $|\langle \mathbf{r}, \mathbf{e} \rangle| \leq ||\mathbf{r}||_1 \cdot ||\mathbf{e}||_\infty = N \cdot B$.

Decryption noise. We're good to go as long as *noise* $\leq \lfloor q/2 \rfloor / 2$!

Regev's PKE scheme: Security

Let n, q, χ be chosen so that $\mathsf{DLWE}_{n,q,\chi}$ holds. Then for any $m \in \{0, 1\}$, the joint distribution $(\mathbf{P}, \mathbf{c})$ is computationally indistinguishable from $U\left(\mathbb{Z}_q^{N\times(n+1)} \times \mathbb{Z}_q^{n+1}\right)$, where $\mathbf{P}$ and $\mathbf{c}$ come from Regev's PKE scheme.

That's all. :)