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)
1. Learning with Errors

- Let $p = p(n) \leq \text{poly}(n)$. Consider the noisy linear equations:

\[
\langle a_1, s \rangle \approx_{\chi} b_1 \pmod{p} \\
\langle a_2, s \rangle \approx_{\chi} b_2 \pmod{p} \\
\vdots
\]

for $s \in \mathbb{Z}_p^n$, $a_i \leftarrow \mathbb{Z}_p^n$, $b_i \in \mathbb{Z}_p$, and error $\chi : \mathbb{Z}_p \rightarrow \mathbb{R}^+$ on $\mathbb{Z}_p$. 

Goal: Recover $s$. 

2. Why we care:

- Believed hard for quantum algorithms
- Average-case = worst-case
- Many crypto applications!
1. Learning with Errors

- Let \( p = p(n) \leq \text{poly}(n) \). Consider the noisy linear equations:

\[
\langle a_1, s \rangle \approx_{\chi} b_1 \pmod{p} \\
\langle a_2, s \rangle \approx_{\chi} b_2 \pmod{p} \\
\vdots
\]

for \( s \in \mathbb{Z}_p^n, a_i \leftarrow \mathbb{Z}_p^n, b_i \in \mathbb{Z}_p \), and error \( \chi : \mathbb{Z}_p \rightarrow \mathbb{R}^+ \) on \( \mathbb{Z}_p \).

- **Goal:** Recover \( s \).
1. Learning with Errors
   ▶ Let $p = p(n) \leq \text{poly}(n)$. Consider the noisy linear equations:

   \[
   \langle a_1, s \rangle \approx_{\chi} b_1 \pmod{p} \\
   \langle a_2, s \rangle \approx_{\chi} b_2 \pmod{p} \\
   \vdots
   \]

   for $s \in \mathbb{Z}_p^n$, $a_i \leftarrow \mathbb{Z}_p^n$, $b_i \in \mathbb{Z}_p$, and error $\chi : \mathbb{Z}_p \rightarrow \mathbb{R}^+$ on $\mathbb{Z}_p$.

   ▶ **Goal**: Recover $s$.

2. Why we care:
   ▶ Believed hard for quantum algorithms
   ▶ Average-case $=$ worst-case
   ▶ Many crypto applications!
1. Intro to lattices
   1.1 What’s a lattice?
   1.2 Hard lattice problems
2. Gaussians and lattices
3. From lattices to learning
4. From learning to crypto
A lattice is a **discrete additive subgroup** of $\mathbb{R}^n$. 

Given $n$ linearly independent vectors $v_1, \ldots, v_n \in \mathbb{R}^n$, the lattice they generate is the set of vectors

$$L(v_1, \ldots, v_n) = \{ \sum_{i=1}^{n} \alpha_i v_i \mid \alpha_i \in \mathbb{Z} \}.$$ 

The basis $B = (|v_1| |v_2| \cdots |v_n|)$ generates the lattice $L(B)$. 

Daniel Apon – Univ of Maryland

An intro to lattices and learning with errors
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 \textbf{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

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

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

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

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

(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 
\[ P(B) = \{ Bx \mid x \in [0, 1]^n \}. \]
More on the fundamental parallelepiped

Useful facts:

- For bases $B_1, B_2$, $L(B_1) = L(B_2) \Rightarrow \text{vol}(P(B_1)) = \text{vol}(P(B_2))$
More on the fundamental parallelepiped

Useful facts:

- For bases $\mathbf{B}_1, \mathbf{B}_2$, $L(\mathbf{B}_1) = L(\mathbf{B}_2) \Rightarrow \text{vol}(P(\mathbf{B}_1)) = \text{vol}(P(\mathbf{B}_2))$
- $\text{vol}(P(\mathbf{B})) = \det(\mathbf{B})$
More on the fundamental parallelepiped

Useful facts:

- For bases $B_1, B_2$, $L(B_1) = L(B_2) \Rightarrow \text{vol}(P(B_1)) = \text{vol}(P(B_2))$
- $\text{vol}(P(B)) = \det(B)$
- $\det(B_1) = \det(B_2)$ iff $B_1 = B_2U$ for a unimodular matrix $U$
More on the fundamental parallelepiped

Useful facts:

- For bases $B_1, B_2$, $L(B_1) = L(B_2) \Rightarrow \text{vol}(P(B_1)) = \text{vol}(P(B_2))$
- $\text{vol}(P(B)) = \det(B)$
- $\det(B_1) = \det(B_2)$ iff $B_1 = B_2U$ for a unimodular matrix $U$
- A matrix $U$ is unimodular if it is integral and $\det(U) = \pm1$. 

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.

Daniel Apon – Univ of Maryland
An intro to lattices and learning with errors
Useful facts:

- For bases $B_1, B_2$, $L(B_1) = L(B_2) \Rightarrow \text{vol}(P(B_1)) = \text{vol}(P(B_2))$
- $\text{vol}(P(B)) = \text{det}(B)$
- $\text{det}(B_1) = \text{det}(B_2)$ iff $B_1 = B_2U$ for a unimodular matrix $U$
- A matrix $U$ is unimodular if it is integral and $\text{det}(U) = \pm 1$.

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

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

Equivalently, the dual of a lattice $L \in \mathbb{R}^n$ is given by

$$L^* = \left\{ y \in \mathbb{R}^n \mid \langle x, y \rangle \in \mathbb{Z}, \text{ for all } x \in L \right\}.$$
The dual of a lattice

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

- Equivalently, the dual of a lattice $L \in \mathbb{R}^n$ is given by

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

- **Fact.** For any $L = L(B)$, $L^* = L(B^*)$,

$$|\text{vol}(P(B))| = \left| \frac{1}{\text{vol}(P(B^*))} \right|. $$
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. $\text{GapSVP}_\gamma$
   - Input: $n$-dimensional lattice $L$ and a number $d > 0$
   - Output: YES if $\lambda_1(L) \leq d$; NO if $\lambda_1(L) > \gamma(n) \cdot d$
Lattice problems

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

1. **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. **CVP**$_{L^*,d}$
   - **Input:** $n$-dimensional (dual) lattice $L^*$ and a point $x \in \mathbb{R}^n$ within distance $d$ of $L^*$
   - **Output:** the closest vector in $L^*$ to $x$
Lattice problems

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

1. **\text{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. **\text{CVP}_{L^*,d}**
   - **Input:** \( n \)-dimensional (dual) lattice \( L^* \) and a point \( x \in \mathbb{R}^n \)
     within distance \( d \) of \( L^* \)
   - **Output:** the closest vector in \( L^* \) to \( x \)

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)
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.
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 $x \in \mathbb{Z}$?

![Number line diagram](image-url)
**Question**: How do you uniformly sample over an unbounded range?

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

**Answer**: You can’t!
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 \( \mathbb{Z}_p \)
**Question:** How do you uniformly sample from $\mathbb{R}^n$?

The "lattice answer": Sample uniformly from the fundamental parallelepiped of a lattice.
Question: How do you uniformly sample from $\mathbb{R}^n$?

Answer: You can’t!
**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.
A related question: What does a lattice look like when you “smudge” the lattice points with Gaussian-distributed 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
Denote by $D_{L,r}$ the discrete Gaussian on a lattice $L$ of width $r$. 

**Important fact.** $\eta(\mathcal{L}) = \omega(\sqrt{\log n}) \approx \Theta(\sqrt{n})$. 

Daniel Apon – Univ of Maryland

An intro to lattices and learning with errors
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 $\epsilon$-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 $\epsilon$-far from the continuous Gaussian (over $L$).

- Important fact. $\eta_{\text{negl}}(n)(L) = \omega(\sqrt{\log n}) \approx \Theta(\sqrt{n})$
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 sketch

1. **Our goal:** Prove LWE is hard
2. Reduction outline
   2.1 Why quantum?
3. Classical step: $D_{L,r} + \text{LWE oracle} \rightarrow \text{CVP}_{L^*,\alpha p/r}$ oracle
4. Quantum step: $\text{CVP}_{L^*,\alpha p/r}$ oracle $\rightarrow D_{L,r\sqrt{n}/(\alpha p)}$
   4.1 **Note:** $(\eta_e(L) \approx) \alpha p > 2\sqrt{n} \rightarrow D_{L,r\sqrt{n}/(\alpha p)} \approx D_{L,<r/2}$
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.
LLL Basis Reduction algorithm: In polytime, given an arbitrary \( L(B) \) outputs a new basis \( B' \) 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.
LLL Basis Reduction algorithm: In polytime, given an arbitrary $L(B)$ outputs a new basis $B'$ 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 $B'$, and using $B'$, draw $n^c$ samples from $D_{L,r_{3n}}$.  

Daniel Apon – Univ of Maryland

An intro to lattices and learning with errors
LLL Basis Reduction algorithm: In polytime, given an arbitrary $L(B)$ outputs a new basis $B'$ 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 $B'$, and using $B'$, draw $n^c$ samples from $D_{L,r_{3n}}$.

For $i = 3n, ..., 1$,

- Call $\text{IterativeStep}$ $n^c$ 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 $D_{L,r_0} = D_{L,r}$. 

Daniel Apon – Univ of Maryland

An intro to lattices and learning with errors
The iterative step

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

$n^c$ samples from $D_{L,r}$

$solve$ $CVP_{L^*,\alpha/r}$

$n^c$ samples from $D_{L,r\sqrt{n}/(\alpha p)}$

classical, uses LWE

$solve$ $CVP_{L^*,(\alpha p^2)/(r\sqrt{n})}$

$n^c$ samples from $D_{L,r n/(\alpha p)^2}$

quantum
Why quantum?

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

You are given an oracle $O$ that, on input $x \in \mathbb{R}^n$ within distance $d$ from $L$, outputs the closest lattice vector to $x$.

(Caveat: If $x$ of distance $> d$ from $L$, $O$’s output is arbitrary.)

How do you use $O$?

One idea: Choose some lattice vector $y \in L$. Let $x = y + z$ with $\|z\| \leq d$. Give $x$ to $O$. 
Why quantum?

Let $L$ be a lattice. Let $d \ll \lambda_1(L)$.

You are given an oracle $O$ that, on input $x \in \mathbb{R}^n$ within distance $d$ from $L$, outputs the closest lattice vector to $x$.

(Caveat: If $x$ of distance $> d$ from $L$, $O$'s output is arbitrary.)

How do you use $O$?

One idea: Choose some lattice vector $y \in L$. Let $x = y + z$ with $\|z\| \leq d$. Give $x$ to $O$.

But then $O(x) = y$!
Why quantum?

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

$$f(x) \overset{\text{def}}{=} \sum_{y \in L} D(y) \exp(2\pi i \langle x, y \rangle) = \mathbb{E}_{y \leftarrow D}[\exp(2\pi i \langle x, y \rangle)]$$
Let $D$ be a probability distribution on a lattice $L$. Consider the Fourier transform $f : \mathbb{R}^n \rightarrow \mathbb{C}$, given by

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

Using Hoeffding’s inequality, if $y_1, \ldots, y_N$ are $N = \text{poly}(n)$ independent samples from $D$, then w.h.p.

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

$$f_{1/r}(x) \approx \exp(-\pi(r \cdot \text{dist}(L^*, x))^2)$$
Classical step: $D_{L,r} + \text{LWE oracle} \rightarrow \text{CVP}_{L^*,\alpha p/r} \text{ oracle}$

- **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} + \text{LWE oracle} \rightarrow \text{CVP}_{L^*,\alpha p/r} \text{ 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 $\text{CVP}_{L^*,1/r}$. (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: $D_{L,r} + \text{LWE oracle} \rightarrow \text{CVP}_{L^*,\alpha p/r} \text{ oracle}$

- **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 $\text{CVP}_{L^*,1/r}$. (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: $D_{L,r} + \text{LWE oracle} \rightarrow \text{CVP}_{L^*,\alpha p/r} \text{ oracle}$

- 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: $D_{L,r} + \text{LWE oracle} \rightarrow \text{CVP}_{L^*,\alpha p/r}$ 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 $\mathbb{Z}_p^n$. 
Classical step: $D_{L,r} + \text{LWE oracle} \rightarrow \text{CVP}_{L^*,\alpha p/r}$ 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 $L + La/p$, for $a \in \mathbb{Z}_p^n$.
  - For any choice of $a \in \mathbb{Z}_p^n$, $L + La/p$ (modulo the parallelepiped) corresponds to a choice of translate.
This motivates defining a new distribution, $\tilde{D}$ with samples $(a, y)$ obtained by:

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

1. $y \leftarrow D_{L/p, r/p}$
2. $a \in \mathbb{Z}_p^n$ s.t. $y \in L + La/p$ (Complicated to analyze..?)

From the previous slide, we know that we can obtain $\tilde{D}$ from $D_{L,r}$.
This motivates defining a new distribution, $\tilde{D}$ with samples $(a, y)$ obtained by:

1. $y \leftarrow D_{L/p, r/p}$
2. $a \in \mathbb{Z}_p^n$ s.t. $y \in L + L \frac{a}{p}$ (← 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, $a \leftarrow \mathbb{Z}_p^n$ (← Ahh! Much nicer. :))
2. Then, $y \leftarrow D_{L+La/p, r/p}$
Classical step: $D_{L,r} + \text{LWE oracle} \rightarrow \text{CVP}_{L^*,\alpha p/r \text{ oracle}}$

This motivates defining a new distribution, $\tilde{D}$ with samples $(a,y)$ obtained by:

1. $y \leftarrow D_{L/p,r/p}$
2. $a \in \mathbb{Z}_p^n$ s.t. $y \in L + La/p$ (← 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, $a \leftarrow \mathbb{Z}_p^n$ (← Ahh! Much nicer. :))
2. Then, $y \leftarrow D_{L+La/p,r/p}$

The width of the discrete Gaussian samples $y$ is tighter now!..
Classical step: $D_{L,r} + \text{LWE oracle} \rightarrow \text{CVP}_{L^*,\alpha p/r} \text{ oracle}$

How about the Fourier transform of $\tilde{D}$? It’s wider now! But...
Classical step: $D_{L,r} + \text{LWE oracle} \rightarrow \text{CVP}_{L^*,\alpha p/r \text{ oracle}}$

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 $D_{L+La/p,r/p}$ with $a=(0,0)$ (left) and $a=(1,1)$ (right).
Key observation #1:

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

\[
\langle a, \tau(x) \rangle = p \langle x, y \rangle \mod p
\]

for \( a \leftarrow \mathbb{Z}_p^n \). After about \( n \) equations, we can use Gaussian elimination to recover \( \tau(x) \in \mathbb{Z}_p^n \).

▶ What if \( x \not\in L^* \)?
Key observation #2:

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

\[
\langle \mathbf{a}, \tau(\mathbf{x}) \rangle \approx \lfloor p \langle \mathbf{x}, \mathbf{y} \rangle \rfloor \mod p
\]

for \( \mathbf{a} \leftarrow \mathbb{Z}_p^n \). After \( \text{poly}(n) \) equations, we use the LWE oracle to recover \( \tau(\mathbf{x}) \in \mathbb{Z}_p^n \). (NOTE: \( |\text{error}| = ||\tau(\mathbf{x})||_2 \))
Key observation #2:

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

$$\langle a, \tau(x) \rangle \approx \lfloor p\langle x, y \rangle \rfloor \mod p$$

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

- This lets us compute the phase $\exp(2\pi i \langle a, \tau(x) \rangle / p)$, and hence recover the closest dual lattice vector to $x$.

- Classical step DONE.
Quantum step: $\text{CVP}_{L^*, \alpha p/r} \Rightarrow D_{L,r\sqrt{n}/(\alpha p)}$

Observe: $\text{CVP}_{L^*, \alpha p/r} \Rightarrow D_{L,r\sqrt{n}/(\alpha p)} = \text{CVP}_{L^*, \sqrt{n}/r} \Rightarrow D_{L,r}$
Ok, let’s give a solution for $\text{CVP}_{L^*, \sqrt{n}/r} \rightarrow D_{L,r}$. 
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.
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 $L^*$: $\sum_{x \in L^*} |x\rangle$. 

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

$$\sum_{x \in L^*, z \in \mathbb{R}^n} \exp(-\pi \|rz\|^2) |x, z\rangle.$$
New quantum step: $\text{CVP}_{L^*, \sqrt{n/r}} \rightarrow D_{L,r}$

**Key rule:** All quantum computations must be reversible.
Key rule: All quantum computations must be reversible.

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

2. Since we have a $\text{CVP}_{L^*, \sqrt{n}/r}$ oracle we can compute $x$ from $x + z$. Therefore, we can **uncompute** the first register:
\[
\sum_{x \in L^*, z \in \mathbb{R}^n} \exp(-\pi \| rz \|^2) |x + z\rangle \approx \sum_{z \in \mathbb{R}^n} f_{1/r}(z) |z\rangle.
\]
New quantum step: $\text{CVP}_{L^*, \sqrt{n/r}} \text{oracle} \to D_{L,r}$

**Key rule:** All quantum computations must be reversible.

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

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

3. Finally, apply the quantum Fourier transform to obtain
   $$\sum_{y \in L} D_{L,r}(y)|y\rangle,$$
   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
For positive integers $n$ and $q \geq 2$, a secret $s \in \mathbb{Z}_q^n$, and a distribution $\chi$ on $\mathbb{Z}$, define $A_{s,\chi}$ as the distribution obtained by drawing $a \leftarrow \mathbb{Z}_q^n$ uniformly at random and a noise term $e \leftarrow \chi$, and outputting $(a, b) = (a, \langle a, s + e \rangle \pmod{q}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. 

An adversary gets oracle access to either $A_{s,\chi}$ 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 $\chi$ with $|\chi| < B$ (meaning, $\chi$ is supported only on $[-B, B]$) s.t. if an efficient algorithm solves the average-case $\text{DLWE}_{n,q,\chi}$ problem, then there is an efficient quantum algorithm that solves GapSVP $\tilde{O}(n \cdot q / B)$ on any $n$-dimensional lattice.
For positive integers $n$ and $q \geq 2$, a secret $s \in \mathbb{Z}_q^n$, and a distribution $\chi$ on $\mathbb{Z}$, define $A_{s,\chi}$ as the distribution obtained by drawing $a \leftarrow \mathbb{Z}_q^n$ uniformly at random and a noise term $e \leftarrow \chi$, and outputting $(a, b) = (a, \langle a, s + e \rangle \pmod{q}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

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

$(\text{DLWE}_{n,q,\chi})$. An adversary gets oracle access to either $A_{s,\chi}$ 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 $\chi$ with $|\chi| < B$ (meaning, $\chi$ is supported only on $[-B, B]$) s.t. if an efficient algorithm solves the average-case $\text{DLWE}_{n,q,\chi}$ problem, then there is an efficient quantum algorithm that solves $\text{GapSVP}_{\tilde{O}(n \cdot q/B)}$ on any $n$-dimensional lattice.
1. **SecretKeyGen**(1^n): Sample s $\leftarrow \mathbb{Z}_q^n$. Output $sk = s$. 

2. **PublicKeyGen**(s): Let $N \text{def} = O(n \log q)$. Sample $A \leftarrow \mathbb{Z}_{N \times n}^q$ and $e \leftarrow \chi_{N \times q}$. Compute $b = A \cdot s + e \mod q$, and define $P \text{def} = [b \mid -A] \in \mathbb{Z}_{N \times (n+1)}^q$. Output $pk = P$.

3. **Enc**(pk): To encrypt a message $m \in \{0, 1\}$ using $pk = P$, sample $r \in \{0, 1\}^N$ and output the ciphertext $c = P^T \cdot r + \lfloor q/2 \rfloor \cdot m \mod q \in \mathbb{Z}_{n+1}^q$, where $m \text{def} = (m, 0, \ldots, 0) \in \{0, 1\}^{n+1}$.

4. **Dec**(sk): To decrypt $c \in \mathbb{Z}_{n+1}^q$ using secret key $sk = s$, compute $m = \lceil 2^q (\langle c, (1, s) \rangle \mod q) \rceil \mod 2$.
Regev’s PKE scheme

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

2. **PublicKeyGen($s$)**: Let $N \overset{\text{def}}{=} O(n \log q)$. Sample $A \leftarrow \mathbb{Z}_q^{N \times n}$ and $e \leftarrow \chi^N$. Compute $b = A \cdot s + e \pmod{q}$, and define

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

Output $pk = P$. 

Regev’s PKE scheme

1. **SecretKeyGen**(1^n): Sample \( s \leftarrow \mathbb{Z}_q^n \). Output \( sk = s \).

2. **PublicKeyGen**(s): Let \( N \defeq O(n \log q) \). Sample \( A \leftarrow \mathbb{Z}_q^{N \times n} \) and \( e \leftarrow \chi^N \). Compute \( b = A \cdot s + e \pmod{q} \), and define

   \[
P \defeq [b| - A] \in \mathbb{Z}_q^{N \times (n+1)}.
   \]

   Output \( pk = P \).

3. \( \text{Enc}_{pk}(m) \): To encrypt a message \( m \in \{0,1\} \) using \( pk = P \), sample \( r \in \{0,1\}^N \) and output the ciphertext

   \[
c = P^T \cdot r + \left\lfloor \frac{q}{2} \right\rfloor \cdot m \mod q \in \mathbb{Z}_q^{n+1},
   \]

   where \( m \defeq (m, 0, \ldots, 0) \in \{0,1\}^{n+1} \).
Regev’s PKE scheme

1. **SecretKeyGen**(1\(^n\)): Sample \(s \leftarrow \mathbb{Z}_q^n\). Output \(sk = s\).

2. **PublicKeyGen**(s): Let \(N \overset{\text{def}}{=} O(n \log q)\). Sample \(A \overset{\$}{\leftarrow} \mathbb{Z}_q^{N \times n}\) and \(e \overset{\$}{\leftarrow} \mathcal{X}^N\). Compute \(b = A \cdot s + e\) (mod \(q\)), and define
   \[
P \overset{\text{def}}{=} [b || -A] \in \mathbb{Z}_q^{N \times (n+1)}.
   \]
   Output \(pk = P\).

3. **Enc\(_{pk}\)(m)**: To encrypt a message \(m \in \{0, 1\}\) using \(pk = P\), sample \(r \in \{0, 1\}^N\) and output the ciphertext
   \[
c = P^T \cdot r + \left\lfloor \frac{q}{2} \right\rfloor \cdot m \mod q \in \mathbb{Z}_q^{n+1},
   \]
   where \(m \overset{\text{def}}{=} (m, 0, \ldots, 0) \in \{0, 1\}^{n+1}\).

4. **Dec\(_{sk}\)(c)**: To decrypt \(c \in \mathbb{Z}_q^{n+1}\) using secret key \(sk = s\), compute
   \[
m = \left\lceil \frac{2}{q} (\langle c, (1, s) \rangle \mod q) \right\rceil \mod 2.
   \]
Encryption noise. Let all parameters be as before. Then for some $e$ where $|e| \leq N \cdot B$, $\langle c, (1, s) \rangle = \left\lfloor \frac{q}{2} \right\rfloor \cdot m + e \pmod{q}$. 
Encryption noise. Let all parameters be as before. Then for some $e$ where $|e| \leq N \cdot B$, $\langle c, (1, s) \rangle = \left\lfloor \frac{q}{2} \right\rfloor \cdot m + e \pmod{q}$.

Proof. $\langle c, (1, s) \rangle = \left\langle P^T \cdot r + \left\lfloor \frac{q}{2} \right\rfloor \cdot m, (1, s) \right\rangle \pmod{q}$

$= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + r^T P \cdot (1, s) \pmod{q}$

$= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + r^T b - r^T As \pmod{q}$

$= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \langle r, e \rangle \pmod{q}$,

and $|\langle r, e \rangle| \leq ||r||_1 \cdot ||e||_\infty = N \cdot B$. 
Regev’s PKE scheme: Correctness

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

**Proof.**
\[
\langle c, (1, s) \rangle = \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \langle r, e \rangle \pmod{q},
\]
and \( |\langle r, e \rangle| \leq \|r\|_1 \cdot \|e\|_\infty = N \cdot B \).

**Decryption noise.** We’re good to go as long as \( \text{noise} \leq \left\lfloor q/2 \right\rfloor/2! \).
Let $n, q, \chi$ be chosen so that $\text{DLWE}_{n,q,\chi}$ holds. Then for any $m \in \{0, 1\}$, the joint distribution $(P, c)$ is computationally indistinguishable from $U \left( \mathbb{Z}_q^{N \times (n+1)} \times \mathbb{Z}_q^{n+1} \right)$, where $P$ and $c$ come from Regev’s PKE scheme.
That’s all. :)