Proving Hardness of LWE
(based on [R05, J. of the ACM])

Oded Regev
Tel Aviv University, CNRS, ENS-Paris
Outline

- Introduction to lattices
- Main theorem: hardness of LWE
- Proof of main theorem
  - Overview
  - Part I: Quantum
  - Part II: Classical
Lattices

Basis:
\( \mathbf{v}_1, \ldots, \mathbf{v}_n \) vectors in \( \mathbb{R}^n \)

The lattice \( L \) is
\[
L = \{a_1 \mathbf{v}_1 + \ldots + a_n \mathbf{v}_n \mid a_i \text{ integers}\}
\]

The dual lattice of \( L \) is
\[
L^* = \{x \mid \forall y \in L, \langle x, y \rangle \in \mathbb{Z}\}.
\]
Shortest Independent Vectors Problem (SIVP)

- SIVP: Given a lattice, find a ‘short’ set of $n$ linearly independent lattice vectors (say within factor $n$ of shortest)
SIVP Seems Hard

- **Best known algorithm runs in time** $2^n$
  
  [AjtaiKumarSivakumar01,...]

- **No better quantum algorithm known!**

- **On the other hand, not believed to be NP-hard**
  
  [GoldreichGoldwasser00, AharonovR04]
Bounded Distance Decoding

- $\text{BDD}_d$: Given a lattice and a target vector within distance $d$, find the closest lattice point.
Main Theorem

Hardness of LWE
LWE

• Fix some $p < \text{poly}(n)$
• Let $s \in \mathbb{Z}_p^n$ be a secret
• We have random equations modulo $p$ with error:

\[ 2s_1 + 0s_2 + 2s_3 + 1s_4 + 2s_5 + 4s_6 + \ldots + 4s_n \approx 2 \]
\[ 0s_1 + 1s_2 + 5s_3 + 0s_4 + 6s_5 + 6s_6 + \ldots + 2s_n \approx 4 \]
\[ 6s_1 + 5s_2 + 2s_3 + 0s_4 + 5s_5 + 2s_6 + \ldots + 0s_n \approx 2 \]
\[ 6s_1 + 4s_2 + 4s_3 + 4s_4 + 3s_5 + 3s_6 + \ldots + 1s_n \approx 5 \]

\ldots

\ldots

\ldots
• More formally, we need to learn $s$ from samples of the form $(t, st+e)$ where $t$ is chosen uniformly from $\mathbb{Z}_p^n$ and $e$ is chosen from $\mathbb{Z}_p$

• Easy algorithms need $2^{O(n \log n)}$ equations/time

• Best algorithm needs $2^{O(n)}$ equations/time
  [BlumKalaiWasserman’00]

• Subexponential algorithm if noise $< \sqrt{n}$ [AroraGe’11]
Main Theorem

LWE is as hard as worst-case lattice problems using a quantum reduction

• In other words: solving LWE implies an efficient quantum algorithm for lattices
Why Quantum?

• As part of the reduction, we need to perform a certain algorithmic task on lattices
• We do not know how to do it classically, only quantumly!
Why Quantum?

- We are given an oracle that solves $BDD_d$ for some small $d$
- As far as I can see, the only way to generate inputs to this oracle is:
  - Somehow choose $x \in L$
  - Let $y$ be some random vector within $dist\ d$ of $x$
  - Call the oracle with $y$
- The answer is $x$. But we already know the answer !!
- Quantumly, being able to compute $x$ from $y$ is very useful: it allows us to transform the state $|y, x\rangle$ to the state $|y, 0\rangle$ reversibly (and then we can apply the quantum Fourier transform)
Proof of the Main Theorem

Overview
Gaussian Distribution

- Recall the discrete Gaussian distribution on a lattice (normalization omitted):

\[ \forall x \in L, \quad D_r(x) = e^{-\|x/r\|^2} \]

- We can efficiently sample from \( D_r \) for large \( r = 2^n \)
The Reduction

- Assume the existence of an algorithm for LWE for $p = 2\sqrt{n}$

- Our lattice algorithm:
  - $r = 2^n$
  - Take $\text{poly}(n)$ samples from $D_r$
  - Repeat:
    - Given $\text{poly}(n)$ samples from $D_r$, compute $\text{poly}(n)$ samples from $D_{r/2}$
    - Set $r \leftarrow r/2$
  - When $r$ is small, output a short vector
$D_{r/2}$
Obtaining $D_{r/2}$ from $D_r$

- **Lemma 1:**
  Given poly(n) samples from $D_r$, and an LWE oracle, we can solve $\text{BDD}_{p/r}$ in $L^*$
  - Classical

- **Lemma 2:**
  Given a solution to $\text{BDD}_d$ in $L^*$, we can obtain samples from $D_{\sqrt{n/d}}$
  - Quantum
  - Based on the quantum Fourier transform

$p=2\sqrt{n}$
Samples from $D_r$ in $L$

Samples from $D_{r/2}$ in $L$

Samples from $D_{r/4}$ in $L$

Solution to $BDD_{p/r}$ in $L^*$

Solution to $BDD_{2p/r}$ in $L^*$

Solution to $BDD_{4p/r}$ in $L^*$

Classical, uses LWE oracle

Quantum
Dual world (L*)

Primal world (L)

$D_r$

$D_{r/2}$

$f_1/r$

$f_2/r$
Fourier Transform

- The Fourier transform of $D_r$ is given by
  \[ f_{1/r}(x) \approx e^{-\|r \cdot \text{dist}(x, L^*)\|^2} \]

- Its value is
  - 1 for $x$ in $L^*$,
  - $e^{-1}$ at points of distance $1/r$ from $L^*$,
  - $\approx 0$ at points far away from $L^*$. 
Proof of the Main Theorem

Lemma 2: Obtaining $D_{\sqrt{n/d}}$ from $BDD_d$
Assume we can solve $\text{BDD}_d$; we’ll show how to obtain samples from $D_{\sqrt{n/d}}$

**Step 1:**
Create the quantum state

$$\sum_{x \in \mathbb{R}^n} \frac{f_d}{\sqrt{n}}(x) |x\rangle$$

by adding a Gaussian to each lattice point and **uncomputing** the lattice point using the BDD algorithm
From BDD$_d$ to D$_{\sqrt{n/d}}$

- **Step 2:**
  Compute the quantum Fourier transform of
  \[ \sum_{x \in \mathbb{R}^n} f_d/\sqrt{n}(x) |x\rangle \]
  It is exactly D$_{\sqrt{n/d}}$ !!

- **Step 3:**
  Measure and obtain one sample from D$_{\sqrt{n/d}}$

- By repeating this process, we can obtain poly(n) samples
Proof of the Main Theorem

Lemma 1: Solving $\text{BDD}_{p/r}$ given samples from $D_r$ and an LWE oracle
It’s enough to approximate $f_{p/r}$

- **Lemma**: being able to approximate $f_{p/r}$ implies a solution to $\text{BDD}_{p/r}$

- **Proof Idea** – walk uphill:
  - $f_{p/r}(x) > \frac{1}{4}$ for points $x$ of distance $< p/r$
  - Keep making small modifications to $x$ as long as $f_{p/r}(x)$ increases
  - Stop when $f_{p/r}(x)=1$ (then we are on a lattice point)
What’s ahead in this part

- For warm-up, we show how to approximate \( f_{1/r} \) given samples from \( D_r \)
  - No need for the LWE oracle
  - This is main idea in [AharonovR’04]

- Then we show how to approximate \( f_{2/r} \) given samples from \( D_r \) and an LWE oracle (for \( p=2 \))

- Approximating \( f_{p/r} \) is similar
Warm-up: approximating $f_{1/r}$

- Let's write $f_{1/r}$ in its Fourier representation:
  \[ f_{1/r}(x) = \sum_{w \in L} \hat{f}_{1/r}(w) \cos(2\pi \langle w, x \rangle) \]
  \[ = \sum_{w \in L} D_r(w) \cos(2\pi \langle w, x \rangle) \]
  \[ = E_{w \sim D_r} [\cos(2\pi \langle w, x \rangle)] \]

- Using samples from $D_r$, we can compute a good approximation to $f_{1/r}$
  (this is the main idea in [AharonovR’04])
• Consider the Fourier representation again:

\[ f_{1/r}(x) = E_{w \sim D_r} [\cos(2\pi \langle w, x \rangle)] \]

• For \( x \in L^* \), \( \langle w, x \rangle \) is integer for all \( w \) in \( L \) and therefore we get \( f_{1/r}(x) = 1 \)

• For \( x \) that is close to \( L^* \), \( \langle w, x \rangle \) is distributed around an integer. Its standard deviation can be (say) 1.
Approximating $f_{2/r}$

- **Main idea:** partition $D_r$ into $2^n$ distributions
- **For** $t \in (Z_2)^n$, denote the translate $t$ by $D^t_r$
- **Given a lattice point we can compute its** $t$
- **The probability on** $(Z_2)^n$ **obtained by sampling from** $D_r$ **and outputting** $t$ **is close to uniform**
Approximating $f_{2/r}$

- Hence, by using samples from $D_r$, we can produce samples from the following distribution on pairs $(t,w)$:
  - Sample $t \in (\mathbb{Z}_2)^n$ uniformly at random
  - Sample $w$ from $D^t_r$

- Consider the Fourier transform of $D^t_r$

$$f^t_{2/r}(x) = E_{w \sim D^t_r} [\cos(\pi \langle w, x \rangle)]$$
\[ f_{2/r}^{0,0} = f_{2/r} \]
$f^{1,1}_{2/r}$
Approximating $f_{2/r}$

- The functions $f^t_{2/r}$ look almost like $f_{2/r}$
- Only difference is that some Gaussians have their sign flipped
- Approximating $f^t_{2/r}$ is enough: we can easily take the absolute value and obtain $f_{2/r}$
- For this, however, we need to obtain several pairs $(t,w)$ for the same $t$
- The problem is that each sample $(t,w)$ has a different $t$!
Approximating $f_{2/r}$

- Fix $x$ close to $L^*$
- The sign of its Gaussian is $\pm 1$ depending on $\langle s, t \rangle \mod 2$ for $s \in (Z_2)^n$ that depends only on $x$
- The distribution of $\langle x, w \rangle \mod 2$ when $w$ is sampled from $D^r_t$ is centred around $\langle s, t \rangle \mod 2$
- Hence, we obtain equations modulo 2 with error:

\[
\begin{align*}
\langle s, t_1 \rangle & \approx \left\lfloor \langle x, w_1 \rangle \right\rfloor \mod 2 \\
\langle s, t_2 \rangle & \approx \left\lfloor \langle x, w_2 \rangle \right\rfloor \mod 2 \\
\langle s, t_3 \rangle & \approx \left\lfloor \langle x, w_3 \rangle \right\rfloor \mod 2 \\
\vdots & \\
\vdots & \\
\end{align*}
\]
Approximating $f_{2/r}$

- Using the LWE oracle, we solve these equations and obtain $s$
- Knowing $s$, we can cancel the sign
- Averaging over enough samples gives us an approximation to $f_{2/r}$
Open Problems

1. What happens for small moduli, say p=2 (learning parity with noise (LPN))? 

2. Dequantize the reduction: 
   - This would immediately improve the security of all LWE-based crypto 
   - Main obstacle: what can one do classically with a solution to \( \text{BDD}_d \)? (see [Peikert09]) 

3. Use quantum hardness assumptions to prove security of other cryptosystems
More Recent Work

• [Peikert09] classical reduction, but exponential modulus and based on GapSVP only

• [StehléSteinfeldTanakaXagawa09] direct quantum reduction from SIS to LWE using the quantum part (but gives weaker hardness of LWE), as well as a ring version of LWE

• [LyubashevskyPeikert09] Ring-LWE
Thanks !!