Hardness of Lattice Problems, Multiplying Quantum States, and Quantum Advantage Relative to a Random Oracle

Today we'll cover 3 results:

① [Regev05]: Solve LWE → Solve "Short Integer Solution" (SIS)
(Quantum reduction)

② [Ajtai96]: Solve SIS → Solve worst-case lattice problems
(Classical reduction)

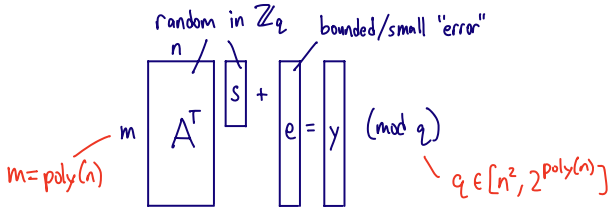
Goal of ①+②: Why do we believe Learning with Errors (LWE) is quantumly hard?

Then, we'll use the main insight behind ① for a very different purpose:

③ [Yamakawa-Zhandry22]: Verifiable Quantum Advantage Relative to a Random Oracle

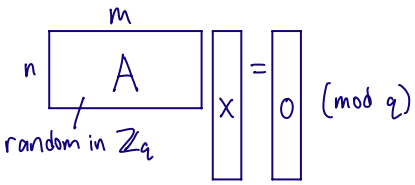
Part 1: Regev's SIS-to-LWE Reduction

Learning with Errors (LWE) Problem [Regev05]



Task: Given A, y, find s.
(for simplicity, we'll assume s is uniquely determined)

Short Integer Solution (SIS) Problem [Ajtai96]



Task: Given A and a bound $poly(n) \ll q$, find x s.t.
- $Ax = 0$
- $\|x\|_\infty \leq poly(n)$ (i.e., x is "short")

How do these problems compare?

① Both SIS and LWE are about finding short vectors
- SIS: Given $A \in \mathbb{Z}_q^{n \times m}$, find short $x \in \mathbb{Z}_q^m$ in kernel of A.
- LWE: Given $A^T, y = A^T s + e$, find s such $A^T s - y$ is short.

② However, LWE has a "planted" short vector in $span(A^T, y)$, while SIS asks for a short vector in the $\ker(A)$.

Overall, LWE seems more "structured" than SIS, and this plays a central role in many applications.
(ex: encryption, delegation of quantum computation, etc.)

Nevertheless, Regev shows:
Any classical or quantum LWE solver implies a quantum SIS solver.

To illustrate the main idea, we'll assume we have a perfect LWE solver $\mathcal{O}_{LWE}$.

$\mathcal{O}_{LWE}(A, y)$ outputs smallest $\hat{e}$ (in $\|\cdot\|_2$ norm) such that $\hat{e} = y - A \cdot s$ for some $s \in \mathbb{Z}_q^n$.

(HW: extend to non-perfect, quantum attacks)

How can we use $\mathcal{O}_{LWE}(A, y)$ to solve SIS?

Regev's approach: Use $\mathcal{O}_{LWE}(A, y)$ to (somehow) construct

$$|goal\rangle = \sum_{\substack{x \in \mathbb{Z}_q^n \\ Ax = 0}} \exp\left(-\frac{|x|^2}{r^2}\right)|x\rangle \quad \text{for small } r$$

(We'll ignore normalization)

Can show: for some $r = poly(n)$, $|goal\rangle$ is almost entirely supported on valid SIS solutions.

Define:
- $|\ker(A)\rangle := \sum_{\substack{x \in \mathbb{Z}_q^m \\ Ax = 0}} |x\rangle$
- $|G_r\rangle = \sum_{x \in \mathbb{Z}_q^m} e^{-|x|^2/r^2}|x\rangle$
  i.e., coefficients follow a width-r Gaussian.

Fact: These states are efficiently constructable.
($|G_r\rangle$ can be constructed via [Grover-Rudolph02])

Observe: $|goal\rangle = |\ker(A)\rangle \odot |G_r\rangle$
(entrywise product)

Main Step: Use $\mathcal{O}_{LWE}$ to prepare this entrywise product state.

Aside: in general, there's no way to prepare $|\psi_0\rangle \otimes |\psi_1\rangle$ from $|\psi_0\rangle$ and $|\psi_1\rangle$.
(Exercise: Show that this allows solving NP-complete problems)

Key technical ingredient: Fourier transform + convolutions

mod q Fourier transform: $\omega_q = e^{2\pi i/q}$ is q^{th} root of unity

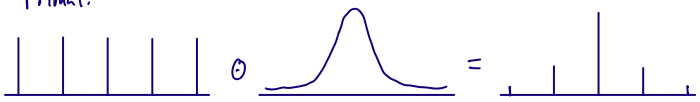
For $x \in \mathbb{Z}_q^m$: $QFT|x\rangle = \sum_{y \in \mathbb{Z}_q^m} \omega_q^{x \cdot y} |y\rangle$

Convolution Theorem:

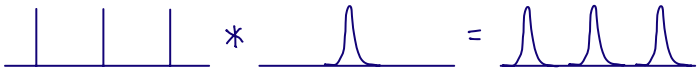
$QFT(|\psi_0\rangle \otimes |\psi_1\rangle) = QFT|\psi_0\rangle * QFT|\psi_1\rangle$

where $\sum \alpha_x |x\rangle * \sum \beta_y |y\rangle = \sum \alpha_x \beta_y |x+y\rangle$

Primal:



Fourier:



Therefore: $QFT|goal\rangle = QFT(|\ker(A)\rangle \otimes |G_r\rangle)$
$= (QFT|\ker(A)\rangle) * (QFT|G_r\rangle)$

Useful properties of Fourier transform:

- $QFT|\ker(A)\rangle = QFT|\text{rowspan}(A)\rangle = \sum_{s \in \mathbb{Z}_q^n} |A^T s\rangle$
- $QFT|G_r\rangle = QFT|G_t\rangle = \sum_{e \in \mathbb{Z}_t^m} \exp(-\frac{|e|^2}{t^2})|e\rangle$
  (where $r \cdot t \approx q$)

Thus, $QFT|goal\rangle = \sum_{\substack{s \in \mathbb{Z}_q^n \\ e \in \mathbb{Z}_t^m}} \exp(\frac{-|x|^2}{t^2})|A^T s + e\rangle$

We'll do this in 3 steps:

1) Prepare $QFT|\ker(A)\rangle \otimes QFT|G_r\rangle = \sum_{\substack{s \in \mathbb{Z}_q^{n_2} \\ e \in \mathbb{Z}_t^m}} \exp(\frac{-|x|}{t^2})|A^T s\rangle|e\rangle$

2) Compute $A^T s + e$ in superposition:

$$\sum_{\substack{s \in \mathbb{Z}_q^n \\ e \in \mathbb{Z}_t^m}} \exp(\tfrac{-|x|^2}{t^2})|A^T s\rangle|e\rangle|A^T s + e\rangle$$

3) Key step: Use $\mathcal{O}_{LWE}$ to uncompute $A^T s, e$.

$$|A^T s\rangle|e\rangle|A^T s + e\rangle \xrightarrow{\text{easy}} |0\rangle|e\rangle|A^T s + e\rangle$$
$$\xrightarrow{\mathcal{O}_{LWE}} |0\rangle|e + \mathcal{O}_{LWE}(A^T s + e)\rangle|A^T s + e\rangle$$
$$= |0\rangle|0\rangle|A^T s + e\rangle$$

That's it! Then take QFT, measure to solve SIS.

We've seen that LWE is as hard as SIS, but how do we know that SIS is hard?

Next up: brief digression from quantum. We'll sketch why breaking SIS/LWE is enough to solve worst-case lattice problems.

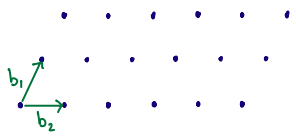
(This will be entirely classical, but it provides a foundation for much of post-quantum/quantum crypto!)

Part 2: Worst-Case to Average-Case Hardness of Lattice Problems

Background on Lattices

Def: Lattice L = discrete subgroup of $\mathbb{R}^n$

$n=2$



L = all integer combinations of some basis $B = (\vec{b}_1 \cdots \vec{b}_n)$

Notation: $L(B) := B \cdot \mathbb{Z}^n = \{\sum_{i \in [n]} x_i \vec{b}_i : x_i \text{ are integers}\}$

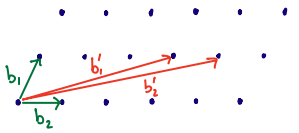
Example Lattice Problems

Shortest Vector Problem (SVP):
Given basis B, find the shortest non-zero vector in $L(B)$.

Shortest Independent Vector Problem (SIVP)
Given basis B, find shortest basis of $L(B)$.
(length of basis $\vec{b}_1, \ldots, \vec{b}_n$ means $\max_{i \in [n]} \|\vec{b}_i\|_2$)

Key point: hard to find short vector/basis when given highly distorted basis as input.

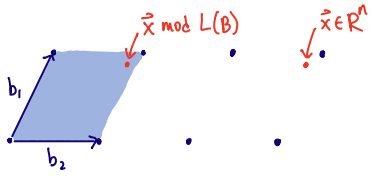
$n=2$



SVP/SIVP might look easy for $n=2$, but we believe hardness scales exponentially in n.

Best known poly(n)-time algorithm for (approximate) SVP finds a $2^{n\lg\lg n/\lg n}$-approximate shortest vector.

Def: $P(B) := B\cdot[0,1)^n$ is the "fundamental parallelepiped" of B.



Observe: 1) $\vec{0}$ is the only lattice point in $P(B)$.
2) $\vec{x}\in L(B)$ iff $\vec{x} \bmod L(B) = \vec{0}$.

Def: G_s is the mean 0, width s, continuous Gaussian on $\mathbb{R}^n$, i.e., G_s outputs $\vec{x}\in\mathbb{R}^n$ with prob $\sim \exp(-\|\vec{x}\|/s^2)$

Smoothing Lemma

For $s \approx$ length of the shortest basis of $L(B)$,

$\vec{x} \bmod L(B)$ for $\vec{x}\leftarrow G_s$ $\approx_{negl}$ uniformly distributed on $P(B)$.

See Vinod Vaikuntanathan's lecture notes for a proof.

This s is called the "smoothing parameter"

Reducing Worst-Case Lattice Problems to SIS

[Ajtai96, Micciancio-Regev05]

We'll show (sketch) how solving SIS can be used to solve worst-case SIVP (find short basis).*

*Caveat: It's not known how to reduce SVP to SIS or LWE — this is a major open question!
However, it is known how to reduce from GapSVP: determine if shortest vector is $\le a$ or $\ge b$ (promise problem)

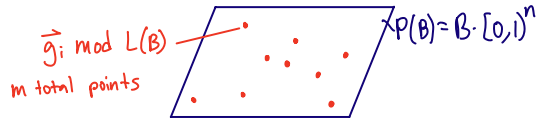
Goal: Given (bad) basis B for $L(B)$, generate short lattice vector.

Intuition: Sample many $\vec{g}_i\leftarrow G_s$, reduce mod $L(B)$, run SIS solver. Then SIS solution x gives us short $\sum x_i\vec{g}_i$ in the lattice. (not quite accurate)

High-Level Template

1) Sample $m=\text{poly}(n)$ i.i.d. Gaussian vectors $\vec{g}_i\leftarrow G_s$ (smoothing parameter)

Write $\vec{g}_i = B\cdot(\vec{c}_i + \vec{a}_i)$. Note: $B\cdot\vec{a}_i = \vec{g}_i \bmod L(B)$.
$\vec{c}_i\in\mathbb{Z}^n$, $\vec{a}_i\in[0,1)^n$



By smoothing lemma, $\vec{g}_i \bmod L(B)$ is uniform in $P(B)=B\cdot[0,1)^n$.
Equivalently, each $\vec{a}_i$ is uniform in $[0,1)^n$.

2) Let $A=(\vec{a}_1\ \vec{a}_2\cdots\vec{a}_m)\in[0,1)^{n\times m}$.

Intuition: If we find $x\in\mathbb{Z}^m$ s.t. $Ax=0$ and $\|x\|_\infty=\text{poly}(n)$ then $\sum_{i\in[m]} x_i\vec{g}_i$ would be a short lattice vector:

- $\sum_i x_i\vec{g}_i = B\big(\underbrace{\sum_i x_i\vec{c}_i}_{\in\mathbb{Z}^n} + \underbrace{\sum_i x_i\vec{a}_i}_{=0^n}\big)\in L(B)$
- $\sum_i x_i\vec{g}_i$ is short since $\|x\|_\infty=\text{poly}(n)$ and $\|\vec{g}_i\|_2 \approx$ smoothing parameter s $\approx$ length of the shortest basis of $L(B)$.

However, integer solutions $x\in\mathbb{Z}^m$ to $Ax=0$ don't exist!
In the actual reduction to SIS mod q, we discretize:

- Multiply A by q, round entrywise to nearest integer.
- Get new $A'\in\mathbb{Z}_q^{n\times m}$ — this is random in $\mathbb{Z}_q^{n\times m}$!
- So we can run SIS solver to find short x s.t. $A'x=0 \bmod q$.
- Output $\sum_i x_i B(\vec{c}_i+\vec{a}_i')$, $\vec{a}_i' = \vec{a}_i$ rounded to multiple of $1/q$.

Additional details:

- Rounding incurs error that scales with B/q so this only works if B is "not too distorted"
- Fortunately, LLL algorithm gives a way to prepare such a B given any initial basis.
- Repeat this whole process many times; eventually get a short basis!

Together with Regev's SIS-to-LWE reduction, this tells us LWE is as hard as worst-case lattice problems!

Up next: Return to quantum world. What else is Regev's technique good for?

Part 3: Verifiable Quantum Advantage w.r.t. a Random Oracle

Want problem that quantum computers solve superpolynomially faster than any classical algorithm.
Moreover, solution should be classically verifiable.

Note: Factoring has a fast quantum algorithm, but we can only conjecture classical hardness (unless P≠NP)

Since we can't prove P≠NP, we often turn to oracle problems, where proving classical hardness is possible.

Examples: - Simon's problem (find hidden period in oracle f)
- Welded trees problem (find sink in oracle-defined "welded tree" graph)

However, these problems require highly structured oracles.

Natural Question: Are structured oracles necessary for verifiable quantum advantage?
[Yamakawa-Zhandry 22]: No! Random Oracles suffice.

[YZ22] design a verifiable task relative to a random oracle by "replacing" the role of $\mathcal{O}_{LWE}$ in Regev's reduction with an efficient classical algorithm.

The YZ Problem

- Let C be a subspace of $\mathbb{Z}_q^n$, $q = \Omega(2^n)$
- Let $H_1, \dots, H_n$ be random oracles where $H_i : \mathbb{Z}_q \to \{0,1\}$

Task: Find $(c_1, \dots, c_n) \in C$ where $H_i(c_i) = 0$ for all i.
C will satisfy some additional properties (TBD).
Notation: $\vec{c} = (c_1, \dots, c_n)$. $H(\vec{c}) = H_1(c_1), \dots, H_n(c_n)$.

Like SIS, the YZ problem also has the form:

	linear system	non-linear constraint
YZ:	$\vec{c}$ in C	$H_i(c_i) = 0 \quad \forall i$
SIS:	$\vec{x}$ in $\ker(A)$	$\lvert x_i \rvert \leq \text{poly}(n) \quad \forall i$

Regev's reduction shows we can (quantumly) compute a solution to
"$x \in \ker(A)$" + "$|x_i|$ short"
if we can solve a decoding task in the Fourier domain.

Primal	Fourier
"$x \in \ker(A)$"	"$A^T s \in \text{rowspan}(A)$"
"$\lvert x_i \rvert$ short" (gaussian)	"$\lvert e_i \rvert$ short" (gaussian)
entrywise product	convolution.

Computing the convolution reduces to "decoding" $A^T s + e$ given $A^T s + e$.

In [YZ22], the corresponding decoding problem is easy by design!
Necessary property of C: vectors in $C^\perp$ are efficiently "error correctable."
That is, given $\vec{c} + \vec{e}$ ($\vec{c} \in C^\perp$, $\vec{e}$ random vector w/ relative Hamming weight $1/2$), can recover $\vec{c} \leftarrow \text{Dec}(\vec{c} + \vec{e})$.

Quantum Algorithm for YZ

Let $|C\rangle := \sum_{\substack{\vec{c} \in \mathbb{Z}_q^n \\ \vec{c} \in C}} |\vec{c}\rangle$ and $|H^{-1}(0)\rangle := \sum_{\substack{\vec{x} \in \mathbb{Z}_q^n \\ H_i(x_i) = 0 \ \forall i}} |\vec{x}\rangle$

We want to prepare

$$|\text{goal}\rangle = \sum_{\substack{\vec{c} \in \mathbb{Z}_q^n \\ \vec{c} \in C \\ H_i(c_i) = 0 \ \forall i}} |\vec{c}\rangle = |C\rangle \otimes |H^{-1}(0)\rangle.$$

Note: Both $|C\rangle$ and $|H^{-1}(0)\rangle$ are efficiently preparable.

- For $i \in [n]$, let $|H_i^{-1}(0)\rangle = \sum_{\substack{x \in \mathbb{Z}_q \\ H_i(x) = 0}} |x\rangle$.
- $|H^{-1}(0)\rangle = |H_1^{-1}(0)\rangle \otimes |H_2^{-1}(0)\rangle \otimes \cdots \otimes |H_n^{-1}(0)\rangle$

  can prepare each $|H_i^{-1}(0)\rangle$ by evaluating H_i on $\sum_{x \in \mathbb{Z}_q} |x\rangle$ until $H_i(x) = 0$ is measured ($1/2$ prob)

Inspired by Regev, we'll try to prepare

$$QFT|\text{goal}\rangle = (QFT|C\rangle) * (QFT|H^{-1}(0)\rangle).$$

What is $QFT|goal\rangle$?

$QFT|C\rangle$ is simply $|C^\perp\rangle$. (superposition over the orthogonal subspace $C^\perp$)

But what is $QFT|H^{-1}(0)\rangle$? Suffices to understand $QFT|H_i^{-1}(0)\rangle$.

- Observe: $|H_i^{-1}(0)\rangle \approx \underbrace{\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{q}}\sum_{x\in\mathbb{Z}_q}|x\rangle\right)}_{\text{uniform superposition}} + \underbrace{\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{q}}\sum_{x\in\mathbb{Z}_q}(-1)^{H_i(x)}|x\rangle\right)}_{\text{(essentially) random state}}$

  Aside: it turns out that $\sum_x (-1)^{H(x)}|x\rangle$ is statistically close to a "Haar-random" quantum state.

- So $QFT|H_i^{-1}(0)\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|random\rangle$.

- Thus, $QFT|H^{-1}(0)\rangle$ is a superposition of vectors $\vec{e}\in\mathbb{Z}_q^n$ with Hamming weight $\approx \frac{1}{2}$, and random values in the non-zero entries.

So $QFT|goal\rangle = |C^\perp\rangle * |H^{-1}(0)\rangle = \sum_{\vec{c}\in C^\perp} |\vec{c}+\vec{e}\rangle$

$\vec{e}\in Error_{y_2}$

($Error_{y_2}$ denotes set of $\vec{e}\in\mathbb{Z}_q^n$ with Hamming weight $\approx 1/2$)

Constructing $QFT|goal\rangle = \sum_{\vec{c}\in C^\perp} |\vec{c}+\vec{e}\rangle$ in 3 steps

$\vec{e}\in Error_{y_2}$

1) Prepare $QFT|C\rangle \otimes QFT|H^{-1}(0)\rangle = \sum_{\vec{c}\in C^\perp} |\vec{c}\rangle|\vec{e}\rangle$

$\vec{e}\in Error_{y_2}$

2) Add in superposition: $\sum_{\vec{c}\in C^\perp} |\vec{c}\rangle|\vec{e}\rangle|\vec{c}+\vec{e}\rangle$

$\vec{e}\in Error_{y_2}$

3) Run Dec to uncompute $\vec{c},\vec{e}$ from $\vec{c}+\vec{e}$:

$$\sum_{\vec{c}\in C^\perp} |\vec{c}\rangle|\vec{e}\rangle|\vec{c}+\vec{e}\rangle \xrightarrow{easy} \sum_{\vec{c}\in C^\perp} |\vec{c}\rangle|0\rangle|\vec{c}+\vec{e}\rangle$$

$\vec{e}\in Error_{y_2}$ $\vec{e}\in Error_{y_2}$

$$\xrightarrow{Dec} \sum_{\vec{c}\in C^\perp} |\vec{c}+Dec(\vec{c}+\vec{e})\rangle|0\rangle|\vec{c}+\vec{e}\rangle = \sum_{\vec{c}\in C^\perp} |0\rangle|0\rangle|\vec{c}+\vec{e}\rangle$$

$\vec{e}\in Error_{y_2}$ $\vec{e}\in Error_{y_2}$

$$= QFT|goal\rangle.$$

That's it*! Just take QFT and measure.

*Missing Piece: We've been ignoring normalization today, but it turns out to matter a lot! In particular, the convolution theorem has an exp(n) constant of proportionality, which blows up even a negl-decoding error. It is analyzed in both Regev and Yz, but is a lot trickier in Yz. (See [Yz22] for details).

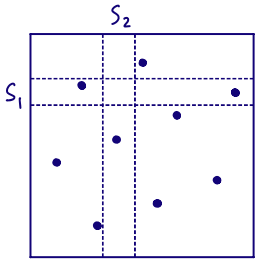
Classical Hardness:

Property: C is "rectangle-evasive"

For any subsets $S_1,\dots,S_n$ satisfying $|S_i| = poly(n)$ for all i,

$$|S_1\times S_2\times\cdots\times S_n \text{ intersect } C| \le 2^{o(n)}.$$

(looks big, but small compared to $|S_1\times S_2\times\cdots\times S_n| = (poly(n))^n$)



($n=2$ example)

Note: Any "list-recoverable" code C is rectangle-evasive.

Yz show: $\exists$ C s.t. $C^\perp$ is error-correctable and C is rectangle-evasive.

Claim:
The Yz problem is classically hard if C is rectangle-evasive.

High-level idea: attacker only learns $H(\vec{c})$ for $2^{o(n)}$ vectors in C, which is not enough to find $H(\vec{c})=0$.

We'll prove this for attackers that make non-adaptive oracle queries and outputs only queried symbols. (i.e., inputs to the oracle independant of prior outputs)

Easy: For any fixed $\vec{c}=(c_1,\dots,c_n)$

$$\Pr_H[H(\vec{c})=0] = \frac{1}{2^n}$$

Therefore, for any set $S\subseteq C$,

$$\Pr_{\{H_i\}}[\exists \vec{c}\in S \text{ s.t. } H(\vec{c})=0] \le \frac{|S|}{2^n}$$

by a union bound.

Let S_i be the set of all inputs the attacker queries H_i on.

Then the conclusion follows from rectangle-evasiveness:

$$\Pr_H\left[\exists \vec{c}\in S_1\times\cdots\times S_n \cap C \text{ s.t. } H(\vec{c})=0\right] \le \frac{2^{o(n)}}{2^n} = negl$$

HW: extend to adaptive queries (requires strengthening of rectangle-evasiveness)