# CS 294. Quantum Computing and Lattices

Quantum computing and lattices have had a close relationship ever since the work of Regev in 2004 [Reg04] which showed a connection between the unique shortest vector problem (which we now know is equivalent to bounded distance decoding) to the hidden subgroup problem on dihedral groups. Since then, the relationship has been very productive.

On the one hand, lattices give us one of the most prominent ways to do "post-quantum" cryptography. For this to be meaningful, we need to be reasonably confident that there are no "fast" quantum algorithms for LWE/SIS. Many have tried and failed.

We will see one of the most important such attempts, Kuperberg's algorithm for the dihedral hidden subgroup problem and its connnection to LWE. Another example is the recent advances in quantum algorithms for the principal ideal problem (PIP) which we will not describe today.

On the other hand, lattices have given us ways to solve fundamental problems in cryptography and quantum computing including generating a verifiable stream of truly random coins, designing classical protocols which check that a quantum computer is doing its job correctly, and designing a quantum money scheme. (See recent works of Mahadev, of Brakerski, Christiano, Mahadev, Vazirani and Vidick, and of Zhandry.) We will unfortunately not have time to delve into any of these today, but please see the course website for pointers.

# 1 A Quantum Computing Primer

**States.** A (pure) quantum state is a *unit vector* in the Hilbert space $\mathbb{C}^N$ for some $N$. Here, $N$ is the universe under consideration. For example, for a one qubit system, $N = 2$; for two qubits, $N = 4$; and for $n$ qubits, $N = 2^n$. For every $x \in [N] = \{0,1\}^n$, we will denote the elementary states as

$$|x\rangle = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 1 \\ 0 \\ \cdots \\ 0 \end{pmatrix}$$

with a 1 in the $x^{th}$ location and 0 everywhere else. In particular, when $n = 1$, we have

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \text{and} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

The vectors $|x\rangle$ form an orthonormal basis for $\mathbb{C}^N$. (Indeed, as we will see later, any orthonormal basis is just as good). A general (pure) quantum state can be written as

$$|\Psi\rangle = \sum_{x \in [N]} \alpha_x |x\rangle$$

where $\alpha_x \in \mathbb{C}$ such that $\sum_{x \in [N]} |\alpha_x|^2 = 1$. For example,

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \text{and} \qquad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

The joint state of $n$ qubits lives in the tensor product of the corresponding Hilbert spaces. If $|x\rangle \in \mathbb{C}^{N_1}$ and $|y\rangle \in \mathbb{C}^{N_2}$ then the joint state, denoted $|x, y\rangle \in \mathbb{C}^{N_1 N_2}$.

**Operations.** Legal operations on qubits have to turn unit vectors into unit vectors; therefore, they are unitary matrices $U \in \mathbb{C}^{N \times N}$. That is,

$$U^\dagger U = I$$

where $U^\dagger$ is the conjugate transpose of $U$ (In case $U$ is a real matrix, this is simply the transpose of $U$.)

- For a single qubit, the $X$ gate is defined by

  $$\mathsf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

  This turns $|0\rangle$ into $|1\rangle$ and vice versa. The $Z$ gate is defined by

  $$\mathsf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

  This keeps $|0\rangle$ the same and turns $|1\rangle$ into $-|1\rangle$.

- For two qubits, the controlled not ($\mathsf{CNOT}$) gate turns $|a, b\rangle$ into $|a, a \oplus b\rangle$. That is, if $a = 0$, it leaves everything the same, but if $a = 1$, it flips the second bit.

  This is defined by

  $$\mathsf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- The quantum Fourier transform $\mathsf{QFT}_n$ over $\mathbb{Z}_N := \mathbb{Z}_{2^n}$ is defined by the $N \times N$ unitary matrix where the $(a, b)^{th}$ entry is $e^{2\pi i ab/N} := \omega_N^{ab}$ where $\omega_N$ denotes the primitive $N^{th}$ root of unity. (The indices run from 0 to $N - 1$.)

  For example, $\mathsf{QFT}_1$ is defined by

  $$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

  and $\mathsf{QFT}_2$ is defined by

  $$\frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

We care about unitaries on $n$ qubits that can be implemented using a quantum circuit with $\mathsf{poly}(n)$ *gates* that each act on a constant number of qubits. We will assert, but not prove, that $\mathsf{QFT}_n$ can be implemented with a poly-size quantum circuit.

**Measurement.** When measuring a qubit $|v\rangle$ in an orthonormal basis $(|b_0\rangle, |b_1\rangle)$, you get 0 with probability $|\langle v, b_0\rangle|^2$ and 1 with probability $|\langle v, b_1\rangle|^2$. For example, let

$$|v\rangle = \mathsf{cos}(\theta)|0\rangle + \mathsf{sin}(\theta)|1\rangle$$

Measuring it in the $|0\rangle, |1\rangle$ basis gives us 0 with probability $\mathsf{cos}^2(\theta)$ and 1 with probability $\mathsf{sin}^2(\theta)$. Measuring it in the $|+\rangle, |-\rangle$ basis gives us 0 $(+)$ with probability $\mathsf{cos}^2(\theta - \pi/4)$ and 1 $(-)$ with probability $\mathsf{sin}^2(\theta - \pi/4)$.

Measuring collapses the state to one of the two basis vectors.

Measuring one qubit in a multi-qubit system does something very similar. Let us show just an example. Let $|v\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle)$ be a state (this is a so-called Bell state.) Measuring the second qubit in the standard basis $(|0\rangle, |1\rangle)$ gives us a uniform bit, and the state collapses to either $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) := |+\rangle$ or $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) := |-\rangle$.

What happens when you measure the second qubit in the Hadamard basis, that is, $|+\rangle, |-\rangle$?

# 2 Dihedral Hidden Subgroup Problem and LWE

## 2.1 The Hidden Subgroup Problem

For a finite set $S$, let $|S\rangle$ denote the state

$$|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{s \in S} |s\rangle$$

In the hidden subgroup problem, we have a known finite group $G$ (presented explicitly), a finite set $S$, and (black-box access to) a function $f : G \to S$ which is constant on all (right) cosets of a *hidden* subgroup $H \leq G$. The goal is to discover $H$, say as a set of its generators.

Nearly all quantum algorithms that achieve superpolynomial speedup do so by solving a hidden subgroup problem, typically over Abelian groups). Examples are Simon's algorithm over $\mathsf{Z}_2^n$ and Shor's algorithm over $Z_N$ where $N$ is a composite number that we wish to factor.

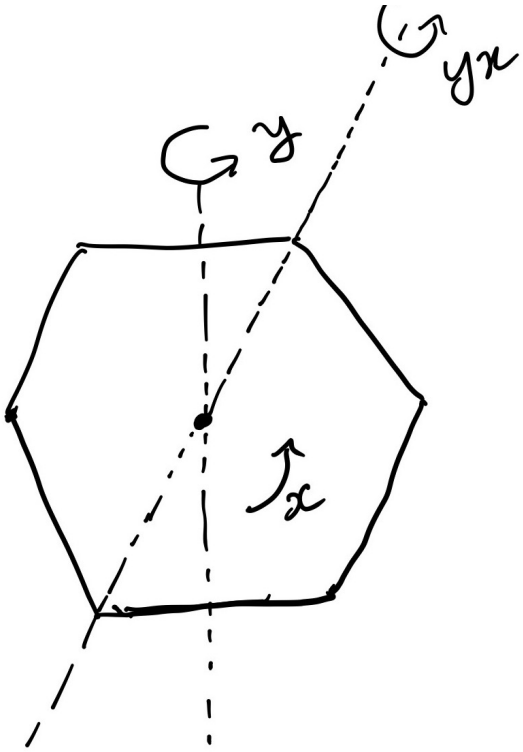



**Figure 1**: Dihedral group $D_6$ as a group of symmetries of a regular hexagon. $y$ denotes reflection across the vertical axis; $x$ denotes a counter-clockwise rotation; and $yx$ (as the reader should check) is reflection across the slanted line.

**Dihedral Group.** This is a (non-Abelian) group of order $2N$ generated by two elements $x$ and $y$ such that

$$x^N = 1,\ y^2 = 1,\ \text{and } yxy = x^{-1}$$

We think of the dihedral group as the group of symmetries of a regular $N$-gon; $x$ is a rotation and $y$ is a reflection along the vertical axis. See the accompanying figure for an illustration.

The subgroups of $D_N$ are either:

1. cyclic subgroups of $Z_N$, consisting only of rotations;
2. **subgroups of order 2 generated by some reflection** $yx^s$; or
3. subgroups generated by a reflection and a copy of a subgroup of $Z_N$.

(1) is Abelian and therefore easy. (3) can be shown to be essentially as easy as (2). If $H$ contains a copy of a cycle $C$, one can reduce the problem to an HSP where $H' = H/C$ on the group $G' = D_N/C \equiv D_{N'}$. So, we will focus on (2).

As we will see, dihedral HSP is very closely related to LWE.

## 2.2 Dihedral HSP

Let $H$ be (the order-2 subgroup) generated by an element $y^b x^s$ in $D_N$ which we will denote by the pair $(b, s) \in \mathsf{Z}_2 \times \mathsf{Z}_N$. Note that with this notation, the group law can be written as

$$(b, s) \odot (b', s') = (b \oplus b', (-1)^{b'} s + s')$$

where $\oplus$ denotes mod-2 addition. Note that in the non-trivial case we are considering, $b = 1$. Finally, note that order of operation matters here as the group is non-Abelian.

The (right) cosets of $H$ are

$$Ha := \{(0, a), (1, s + a)\}$$

for all $a \in \mathsf{Z}_N$.

**Generating coset states (Coset sampling).** First, create a superposition $|G\rangle$ over all elements of $G := D_N$.

$$\sum_{b' \in \{0,1\}, s' \in \mathbb{Z}_N} |b', s'\rangle$$

Tensor this with the singleton state to produce

$$\sum_{b' \in \{0,1\}, s' \in \mathbb{Z}_N} |b', s', 0\rangle$$

Compute the function $f : G \to S$ in the description of the hidden subgroup problem coherently in superposition.

$$\sum_{b' \in \{0,1\}, s' \in \mathbb{Z}_N} |b', s', f(b', s')\rangle$$

Measure the third register to get the state

$$\sum_{(b', s') \in Ha} |b', s'\rangle = |0, a\rangle + |1, s + a\rangle$$

for some (unknown) $a \in \mathsf{Z}_N$.

### 2.3 From LWE to (Robust) Dihedral HSP

We start by showing the relevance of the dihedral HSP by showing how to reduce LWE to it. In fact, we will crucially need a stronger version of dihedral HSP which we call robust dihedral HSP. In this variant, the function $f$ can make an error with some small probability $\varepsilon$ so that the coset states are "correct" with probability $1 - \varepsilon$, and are a singleton state (which is not a valid coset state) with probability $\varepsilon$.

Jumping ahead, we remark that we will show a subexponential algorithm for dihedral HSP in a little bit. But the algorithm seems to be not particularly noise-tolerant. Rather frustratingly (or shall we say fortunately), this is what prevents us from using the reduction this section together with Kuperberg's algorithm to come up with a subexponential quantum algorithm for LWE!

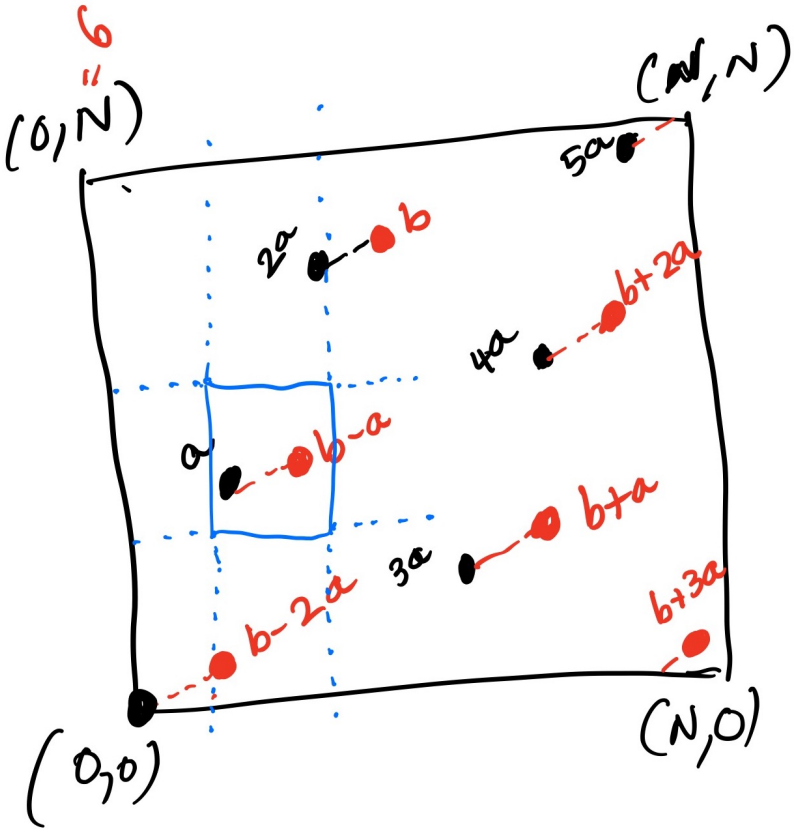
For simplicity, we will reduce from a one-dimensional version of LWE (which, for appropriate parameters, can be shown to be as hard as LWE itself) to dihedral HSP.

So, you are given

$$(\mathbf{a}, \mathbf{b} := s\mathbf{a} + \mathbf{e}) \in \mathsf{Z}_N^{1\times m} \times \mathsf{Z}_N^{1\times m}$$

for some $N$ (which you should think of as exponential in the security parameter $\lambda$) and $m$ (which you should think of as polynomial in $\lambda$). Your goal is to recover $s \in \mathsf{Z}_N$.

Partition the space into cubes of side-length $\ell$. We will set $\ell$ so that $\ell\sqrt{m} < \lambda_1(L) \approx O(q)$, but so that $\ell \gg ||\mathbf{e}||$. (Note that this already places a limit on the LWE error for which we can solve it, i.e., $||\mathbf{e}|| \ll q/\sqrt{m}$.) Let $\phi$ be the function that maps a point in $\mathsf{Z}_N^m$ to its associated cube.



Create the state

$$|0\rangle \sum_{t \in \mathbb{Z}_N} |t, \phi(t\mathbf{a})\rangle + |1\rangle \sum_{t \in \mathbb{Z}_N} |t, \phi(\mathbf{b} + t\mathbf{a})\rangle = |0\rangle \sum_{t \in \mathbb{Z}_N} |t, \phi(t\mathbf{a})\rangle + |1\rangle \sum_{t \in \mathbb{Z}_N} |t - s, \phi(\mathbf{e} + t\mathbf{a})\rangle$$

Measure the second register which will give us the name of a subcube. The rest of the state will either collapse to a singleton (when there is either a lattice point or a shifted lattice point in the cube, but not both) or a superposition of two points (when there is both a lattice point and a shifted lattice point in the cube). It is easy to check that the way we set up parameters, there will never be two lattice points (resp. two shifted lattice points in the same cube).

So, in the good case, we get

$$|0\rangle|t\rangle + |1\rangle|t - s\rangle$$

Starting from our LWE sample, we can produce as many of these states as we like. As we saw a few minutes ago, these are precisely the coset states of the dihedral HSP. So, any algorithm that solves the dihedral HSP by coset sampling will give us an algorithm for LWE.

The one wrinkle in this reduction is that sometimes we get singleton states which are not valid coset states for the dihedral HSP (and we never know when we got those, so we can't throw them away.) How often do we get singleton states?

I will leave it to you to check that this happens with probability roughly $||\mathbf{e}|| \cdot n^{1.5}/N$.

## 2.4 Kuperberg's Algorithm for Dihedral HSP

We will show the algorithm for $N = 2^n$. This can be generalized to any $N$ with essentially the same complexity.

Let $H$ be (the order-2 subgroup) generated by an element $y^b x^s$ in $D_N$ which we will denote by the pair $(b, s) \in \mathsf{Z}_2 \times \mathsf{Z}_N$. Note that with this notation, the group law can be written as

$$(b, s) \odot (b', s') = (b \oplus b', (-1)^{b'} s + s')$$

where $\oplus$ denotes mod-2 addition. Note that in the non-trivial case we are considering, $b = 1$. Finally, note that order of operation matters here as the group is non-Abelian.

The (right) cosets of $H$ are

$$Ha := \{(0, a), (1, s + a)\}$$

for all $a \in \mathsf{Z}_N$.

**Generating coset states.** First, create a superposition $|G\rangle$ over all elements of $G := D_N$.

$$\sum_{b' \in \{0,1\}, s' \in \mathbb{Z}_N} |b', s'\rangle$$

Tensor this with the singleton state to produce

$$\sum_{b' \in \{0,1\}, s' \in \mathbb{Z}_N} |b', s', 0\rangle$$

Compute the function $f : G \to S$ in the description of the hidden subgroup problem coherently in superposition.

$$\sum_{b' \in \{0,1\}, s' \in \mathbb{Z}_N} |b', s', f(b', s')\rangle$$

Measure the third register to get the state

$$\sum_{(b', s') \in Ha} |b', s'\rangle = |0, a\rangle + |1, s + a\rangle$$

for some (unknown) $a \in \mathsf{Z}_N$.

**Quantum Fourier Transform over $\mathsf{Z}_N$.** QFT gives us the state

$$|0\rangle \sum_{x\in\mathsf{Z}_N} \omega_N^{ax}|x\rangle + |1\rangle \sum_{x\in\mathsf{Z}_N} \omega_N^{x(s+a)}|x\rangle$$
$$= \eta \cdot \left( \sum_{x\in\mathsf{Z}_N} |0,x\rangle + \omega_N^{xs}|1,x\rangle \right)$$

where $\eta$ is a global phase (which no measurement can distinguish and can be ignored.)

Measure the second register to get a value $x \in \mathsf{Z}_N$ and the state

$$|\Psi_x\rangle := |0\rangle + \omega_N^{xs}|1\rangle$$

This can be repeated many times to generate a random $x \in \mathsf{Z}_N$ together with the state $\Psi_x$.

**Kuperberg Sieve.** We now have many copies $(x, |\Psi_x\rangle)$ and wish to find $s$, therefore solving the HSP. We will now focus on finding *a single bit* of $s$, namely its least significant bit. This can later be iterated with every single bit of $s$ to recover the entire value.

That is, our goal will be to somehow produce the state

$$|0\rangle + (-1)^{s \bmod 2}|1\rangle$$

from which $s \bmod 2$ can be recovered by measuring in the $|+\rangle, |-\rangle$ basis.

What we will next do should remind you of an algorithm we have already seen in class. (Can you remember which one?)

We produce $Q$ such states. We note that if we take two such states

$$|00\rangle + \omega_N^{x_1 s}|10\rangle + \omega_N^{x_2 s}|01\rangle + \omega_N^{(x_1+x_2)s}|11\rangle$$

and apply a CNOT operator (with the first qubit as the control), we get

$$|00\rangle + \omega_N^{x_1 s}|11\rangle + \omega_N^{x_2 s}|01\rangle + \omega_N^{(x_1+x_2)s}|10\rangle$$

Measure the second qubit to get

$$|0\rangle + \omega_N^{(x_1+x_2)s}|1\rangle \quad \text{or} \quad |0\rangle + \omega_N^{(x_1-x_2)s}|1\rangle$$

where the former happens if the measurement resulted in 0 and the latter if it resulted in 1.

Here is the consequence. Assume that $k$ least significant bits of $x_1$ and $x_2$ were the same to begin with. This gives us a procedure to take two qubits and with probability $1/2$ produce a single qubit $|0\rangle + \omega_N^{xs}|1\rangle$ where $k$ of the least significant bits of $x$ are 0. We can continue this procedure to "clear out" more and more of the LSBs and eventually keep just the MSB of $x$. In this case, it is easy to check that the resulting state is $|0\rangle + \omega_N^{s \bmod 2}|1\rangle$ if the MSB of $x$ is 1.

Each step "consumes" on average four qubits to produce a better qubit.

- If $k$ is too small, we need many qubits to get to the end, roughly $4^{n/k}$.
- If $k$ is too large, we may not be able to "pair up" the qubits with friends so that the least significant bits of the corresponding $x$ match. Roughly speaking, we need about $2^k$ qubits to make sure, by the coupon collector bound, that *most qubits have friends.*

Fortunately, there is a point in the tradeoff space, and as the calculation suggests, the right thing to do is set $2n/k \approx k$ or $k \approx \sqrt{2n}$.

Thus, we start with producing $Q_0 = (k + n/4k) \cdot 2^k \cdot 4^{n/k}$ qubits

$$(x, |\Psi_x\rangle)$$

Each step potentially loses $2^k$ qubits which cannot be paired, and roughly a factor 4 because of the sieving step. So, we get in expectation $Q_1 = (Q_0 - 2^k)/4$ qubits. At the end,

$$Q_{n/k} = Q_0 \cdot 4^{-n/k} - 2^k \cdot (n/4k) \approx k2^k \gg O(1)$$

qubits remain. Since none of the operations "looked at" the MSB of the $x$, the resulting bits will have $MSB(x) = 0$ or 1 nearly equiprobably. In the event that $MSB(x) = 1$, we obtain the desired outcome, i.e., the LSB of $s$.

**Setting the Parameters.** Balancing $k$ and $n/k$ gives us a roughly

$$2^{O(\sqrt{n})} = 2^{O(\sqrt{\log N})}$$

time quantum algorithm. The memory consumption is nearly the same as time, but this has been improved subsequently by Kuperberg to use $2^{O(\sqrt{\log N})}$ time and *classical space* but only $O(\log N)$ quantum memory.

## References

[Reg04] Oded Regev. Quantum computation and lattice problems. *SIAM J. Comput.*, 33(3):738–760, 2004.