COS 597A: Quantum Cryptography
Lecture 10 (October 15, 2018)
Lecturer: Mark Zhandry

Princeton University

Scribe: James Bartusek

Notes for Lecture 10

Recall the LWE and SIS problems from the last couple of lectures. Today we'll study the question: why are SIS/LWE thought to be hard for quantum computers to break? Here are some reasons:

- A couple dozen (very smart) people have tried and failed to find quantum algorithms breaking SIS/LWE
- Their exact versions are NP-hard, and it is widely believed that quantum computers cannot be used to solve every problem in NP in polynomial time
- They are related to the Hidden Subgroup Problem over *non-abelian* groups

We will focus on this last point in today's lecture.

1 Hidden Subgroup Problem

We are given a group G that contains a secret subgroup H, and a function f such that $f(g \cdot h) = f(g)$ iff $h \in H$, where $\cdot$ denotes the group operation. Recall that quantum computers can be used to efficiently solve this problem in the case that G is commutative (abelian). However, there is no known efficient algorithm in the case of a non-abelian G.

We'll consider perhaps the simplest example of a non-abelian group, namely, the Dihedral group D_n. D_n is a group of order $2n$ which consists of the symmetries of a regular n-gon (there are n rotations and n reflections). There are a few subgroups of D_n, one of which is the group of rotations. However, we'll let the secret subgroup H consist of just the identity and some secret reflection.

This 'Dihedral Subgroup Problem' is actually related to SIS/LWE, but before getting to the connections, we'll first describe the problem in a more intuitive way. The following is a description of the 'Hidden Shift Problem'. Let $f_0, f_1 : \mathbb{Z}_n \to S$ be injective functions with arbitrary range S such that there exists a hidden shift s such that $f_0(x) = f_1(x + s \bmod n)$ for all x. The goal will be to find s. To map back to the dihedral subgroup problem, we define $f(b, x) = f_b(x)$ where $b \in \{0, 1\}$ and $x \in \mathbb{Z}_n$.

2 Algortihm for HSP?

We'll follow the blueprint of previously seen quantum algorithms in an attempt to solve the hidden shift problem, and along the way we'll see connections to LWE. We assume that the underlying shift s has been chosen uniformly at random from $\mathbb{Z}_n$.

1. Start with the uniform superposition over the domain

$$\frac{1}{\sqrt{2n}}\sum_{b,x}|b,x\rangle$$

2. Apply f in superposition

$$\frac{1}{\sqrt{2n}}\sum_{b,x}|b,x,f_b(x)\rangle$$

3. Measure $f_b(x) \rightarrow y$. In this case there are only two inputs that map to y due to the injectivity of f_0 and f_1

$$\frac{1}{\sqrt{2}}|0,x\rangle + \frac{1}{\sqrt{2}}|1,x+s \bmod n\rangle$$

4. Now we usually apply the QFT over the domain, but previously our domain has been something like $\mathbb{Z}_n$, which is commutative. One can define the QFT over non-abelian groups, however, we don't know an efficient quantum algorithm for computing it. Now, it is known that given enough - $O(\log(|G|))$ - samples that result from step 3, we can *inefficiently* find s. Instead, we'll consider applying the QFT mod n to the final $\log(n)$ qubits (all except the first). This results in

$$\frac{1}{\sqrt{2n}}\sum_{y}|0,y\rangle\omega_n^{xy} + \frac{1}{\sqrt{2n}}\sum_{y}|1,y\rangle\omega_n^{(x+s)y} = \frac{1}{\sqrt{2n}}\sum_{b,y}|b,y\rangle\omega_n^{(x+bs)y}$$

Now we repeat steps 1-4 k times and tensor the results together to form the state

$$\frac{1}{(\sqrt{2n})^k}\sum_{\vec{b},\vec{y}}|\vec{b},\vec{y}\rangle\omega_n^{(\vec{x}+s\vec{b})\cdot\vec{y}}$$

where $\vec{b} \in \{0,1\}^k, \vec{x}, \vec{y} \in \mathbb{Z}_n^k$.

5. Measure $\vec{y}$ to recover the state $|\psi_{s,\vec{y}}\rangle$.

$$|\psi_{s,\vec{y}}\rangle = \frac{1}{\sqrt{2^k}} \sum_{\vec{b}} |\vec{b}, \vec{y}\rangle \omega_n^{(\vec{x}+s\vec{b})\cdot\vec{y}} = \frac{\omega_n^{\vec{x}\cdot\vec{y}}}{\sqrt{2^k}} \sum_{\vec{b}} |\vec{b}, \vec{y}\rangle \omega_n^{s\vec{b}\cdot\vec{y}}$$

We can drop the overall phase term and write

$$\frac{1}{\sqrt{2^k}} \sum_{\vec{b}} |\vec{b}, \vec{y}\rangle \omega_n^{s\vec{b}\cdot\vec{y}}$$

6. Now we'll write out the density matrix for the state which results from the measurement, which considers all possible values of s and of $\vec{y}$.

$$\sum_{s,\vec{y}} \frac{1}{nn^k} |\psi_{s,\vec{y}}\rangle\langle\psi_{s,\vec{y}}| = \sum_{s,\vec{y}} \frac{1}{n(2n)^k} \sum_{\vec{b},\vec{b}'} |\vec{b}, \vec{y}\rangle\langle\vec{y}, \vec{b}'| \omega_n^{s\cdot\vec{b}\cdot\vec{y}-s\cdot\vec{b}'\cdot\vec{y}}$$

$$= \frac{1}{n(2n)^k} \sum_{\vec{b},\vec{b}',\vec{y}} |\vec{b}, \vec{y}\rangle\langle\vec{b}', \vec{y}| \sum_{s} \omega_n^{s(\vec{b}-\vec{b}')\cdot\vec{y}} = \frac{1}{(2n)^k} \sum_{\substack{\vec{y} \\ \vec{b}\cdot\vec{y}=\vec{b}'\cdot\vec{y}}} |\vec{b}, \vec{y}\rangle\langle\vec{b}', \vec{y}|$$

Where the last equality follows since

$$\sum_{s} \omega_n^{s(\vec{b}-\vec{b}')\cdot\vec{y}} = \begin{cases} n & \text{if } (\vec{b}-\vec{b}')\cdot\vec{y}' = 0 \\ 0 & \text{if } (\vec{b}-\vec{b}')\cdot\vec{y}' \neq 0 \end{cases}$$

7. We notice that the above state is what results when setting up the uniform state $\sum_{\vec{y},\vec{b},\vec{b}'} |\vec{b}, \vec{y}\rangle\langle\vec{b}', \vec{y}|$ and measuring $\vec{y}\cdot\vec{b}$. Thus we can imagine an alternative interpretation of the algorithm where we choose a random $\vec{y}$, create the state $\frac{1}{\sqrt{2^n}} \sum_{\vec{b}\in\{0,1\}^n} |\vec{b}\rangle$, compute and measure $\vec{y}\cdot\vec{b} = c$, and end up with the state

$$\sum_{\substack{\vec{b}\in\{0,1\}^n \\ \text{s.t. } \vec{y}\cdot\vec{b}=c \bmod n}} |\vec{b}\rangle$$

Now, relabel $\vec{y}$ as a matrix M (with one row) and say for now that $c = 0$. Then what we have is a superposition over vectors in the kernel of M, like we saw last time. One major difference is that the weight is uniform rather than Gaussian. This can be remedied, but we won't discuss it today.

8. Apply QFT to get a superposition of LWE samples (like we saw last time). Recall that c becomes a phase term, no its not important that it could have been non-zero.

9. Apply a decisional LWE algortihm to distinguish the state from random, thus solving the decisional HSP problem: distinguish between f_0, f_1 s.t. $\exists s$ s.t. $f_0(x) = f_1(x + s)$ OR f_0, f_1 with disjoint images.

   The above algorithm shows that an algortihm solving LWE can be used to solve HSP. The converse is also roughly true: if an algorithm can solve HSP using techniques similar to what we just saw, then it can be used to solve LWE.