QIC 890/891 Selected Advanced Topics in Quantum Information

Module 1: *Quantum Algorithms for Number Theory Problems*

Lecture 1

Lecturer: Fang Song, University of Waterloo *May 4, 2015*

► **Theme of this crash course**. In this short module, we will look at a central problem in quantum computing, the Hidden Subgroup Problem (HSP) on an *abelian* group G. We will see efficient quantum algorithms for HSP instances where G is finite abelian, $\mathbb{Z}$, $\mathbb{R}$ and $\mathbb{R}^n$. Then we will discuss how quantum computers can solve efficiently a few interesting problems from number theory, e.g., factoring large integers and finding the unit group of a number field, by reducing them to the HSP problem. These number theory problems are usually believed hard classically and there are many crypto-systems based on them. We will show a few examples of crypto-systems that will become broken by quantum computers.

A brief syllabus:

- Lecture 1. HSP on finite abelian groups, QFT, and phase estimation.
- Lecture 2. Period finding on $\mathbb{Z}$ and $\mathbb{R}$, and solving factoring and Pell's equation.
- Lecture 3. Period finding on $\mathbb{R}^n$ for arbitrary n.
- Lecture 4. Computing the unit group in a number field, and connections to lattice-based crypto.

Basic familiarity with quantum information and quantum algorithms is assumed.

Supporting materials for this course:

- Lecture notes by Andrew Childs [**?**] and Umesh Vazirani [**?**].
- Research papers as we go along.
- and of course our loyal friend: QIQC by Nielsen & Chuang [**?**]

► **Content of this lecture**. This lecture will review quantum Fourier transform, phase estimation, and HSP on finite abelian groups. We will also see how to apply these HSP algorithms to break a construction of (psudo-)random permutation based on Feistel networks, and any crypto-systems based on discrete-logarithm.

1 HSP on (finite) abelian groups

All groups we consider in this course are abelian. Let's first study the case that G is finite.

Definition 1 (HSP on a finite abelian group)**.** *Given a black-box function $f : G \to S$, where G is a known group (use addition '$+$' as group operation) and S is a set. f satisfies the promise that:*

$$f(x) = f(y) \quad \textit{if and only if} \quad x \in y + H$$

for some unknown subgroup $H \leq G$. The goal is to find H (i.e., to compute a generating set for H) using queries to f.

We usually say that f *hides* H. We will denote such an instance as $\mathrm{HSP}(f, G)$. The complexity of an algorithm for solving it is parameterized by $\log |G|$. For example, an algorithm is considered efficient if it runs in time $\mathrm{poly}(\log |G|)$.

▶ **Quantum Fourier Transform**. The central tool we need, which is perhaps the most important unitary transformation in quantum computing, is the *quantum Fourier transform* (QFT). For finite abelian G, QFT is

$$\mathcal{F}_G := \frac{1}{|G|} \sum_{x \in G} \sum_{y \in \hat{G}} \chi_y(x) |y\rangle\langle x|$$

The objects appearing in the expression are explained below.

- χ_y: yth character of G. A *character* is a homomorphism from G to $\mathbb{C}$. Hence $\chi_y(x)$ is the value of χ_y evaluated at $x \in G$.
- $\hat{G} := \{\chi_y\}$ is the set of characters of G and is called the dual group of G. G and $\hat{G}$ are isomorphic, so it is usually convenient to label $\hat{G}$ by elements of G.

$\boxed{\text{Fact:}}$ Distinct characters are orthogonal in the sense that

$$\forall y, y' \in \hat{G}, \quad \frac{1}{|G|} \sum_{x \in G} \chi_y(x) \chi^*_{y'}(x) = \delta_{y,y'}$$

We will see later how to implement $\mathcal{F}_G$ by a poly-size quantum circuit. Now we apply QFT to solve $\mathrm{HSP}(f, G)$.

1.1 Quantum algorithm for $\mathrm{HSP}(f, G)$

▶ **The algorithm**.

1. create uniform superposition over the group (by $\mathcal{F}_G$ on $|0\rangle$) and evaluate f, we have

$$|0, 0\rangle \mapsto \frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |f_x\rangle$$

2. measure the 2nd register we get a coset state

$$|x + H\rangle := \frac{1}{\sqrt{|H|}} \sum_{h \in H} |x + h\rangle$$

for a random $x \in G$.

3. Apply $\mathcal{F}_G$ on $|x + H\rangle$, we get

$$\frac{1}{\sqrt{|H| \cdot |G|}} \sum_{y \in \hat{G}} \sum_{h \in H} \chi_y(x + h) |y\rangle = \sqrt{\frac{|H|}{|G|}} \sum_{y \in \hat{G}} \chi_y(x) \chi_y(H) |y\rangle$$

with $\chi_y(H) := \frac{1}{|H|} \sum_{h \in H} \chi_y(h)$.

4. measure and get y.

5. Repeat the above $O(\log|G|)$ times and recover H from the samples $\{\chi_y\}$ by classical post-processing.

The quantum circuit is shown below.

Quantum circuit for HSP

▶ **Analysis**. Note that χ_y restricted on H is also a character on H. We claim that in step 4 we measure $y \in \hat{G}$ if only if $\chi_y(h) = 1, \forall h \in H$. This is because if $\chi_y(h) \neq 1$ for some $h \in H$, which means that χ_y is not the trivial character $\chi_0 : \forall x \in G, \chi_0(x) = 1$, then by orthogonality of characters

$$\chi_y(H) = \frac{1}{|H|}\sum_{x\in G}\chi_y(x) = \frac{1}{|H|}\sum_{x\in G}\chi_y(x)\cdot\chi_0(x) = 0\,.$$

This means that we get random samples of χ_y whose kernel $\{x \in G : \chi_y(x) = 1\}$ contains H. With sufficiently many $O(\log|G|)$ samples, we can find H with high probability by efficiently computing the intersection of all the kernels. The details of the classical post-processing are standard (see for example [**?**]).

1.2 Examples and Applications in Cryptography

▶ $G = \mathbb{Z}_2^n$**: Simon's problem**. In Simon's problem, we have a function $f : \mathbb{Z}_2^n \to S$, such that $f(x) = f(y)$ iff. $x = y + s$ for some unknown $s \in \{0,1\}^n$. This readily reduces to HSP on $G = \mathbb{Z}_2^n$, where $H = \{0, s\}$. Hence we can find s efficiently on a quantum computer. To the contrary, it is easy to argue that any probabilistic classical algorithm needs $\Omega(2^{n-1})$ queries to f.

Simon's algorithm has an interesting application in attacking a construction of pseudo-random permutations from pseudo-random functions in classical cryptography.

3-round Feistel network

Fact: If each $f_i, i = 1,2,3$ is chosen independently from $\{f : \{0,1\}^n \to \{0,1\}^n\}$ uniformly at random, then any probabilistic algorithm making $o(2^{cn})$ queries can not distinguish $\Pi(f_i)$ from a truly random permutation $P : \{0,1\}^{2n} \to \{0,1\}^{2n}$.

Remarks. If we substitute pseudo-random function for each random round function f_i, we obtain a pseudo-random permutation.

However, this constructions fails against quantum algorithms. Kuwakado and Morii [**?**] observed that the 3-round Feistel construction actually possess certain structure that allows one to define an instance of Simon's problem from $\Pi(f_i)$ with high probability. Meanwhile, a truly random permutation has no such structure. Therefore a quantum distinguisher can tell them apart efficiently.

▶ $G = \mathbb{Z}_N \times \mathbb{Z}_N$**: discrete logarithm**. Discrete logarithm in $L = \langle g\rangle$. Let $N = |L|$ be the order of g. Given $y \in L$ find $x \in \mathbb{Z}_N$ such that $g^x = y$. Denote such an x as $\log_g y$.

It reduces to HSP(f, G), where $G = \mathbb{Z}_N \times \mathbb{Z}_N$ and f is defined as follows:

$$f(a,b) = x^a g^b\,.$$

Hidden subgroup $H = L_0 := \{(a,b) \in \mathbb{Z}_N^2 : x^a g^b = 1\}$

$$f(a,b) = f(a',b') \Leftrightarrow x^a g^b = x^{a'} g^{b'} \Leftrightarrow x^{a-a'} g^{b-b'} = 1 \Leftrightarrow (a-a', b-b') \in L_0\,.$$

$N \times N$ grid of coset lines

2 Implementing QFT

► $G = \mathbb{Z}_2^n$. In this case, $\mathcal{F}_{\mathbb{Z}_2^n} = \frac{1}{\sqrt{2^n}} \sum_{x,y \in \mathbb{Z}_2^n} (-1)^{x \cdot y} |y\rangle\langle x| = H^{\otimes n}$.

► $G = \mathbb{Z}_N, N = 2^n$. For any $m \in \mathbb{Z}$, denote $\omega_m := e^{\frac{2\pi i}{m}}$. There is a simple product formula for $\mathcal{F}_{\mathbb{Z}_{2^n}}$:

$$\mathcal{F}_{\mathbb{Z}_{2^n}} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_{2^n}} \omega_{2^n}^{x \cdot y} |y\rangle = \otimes_{k=0}^{n-1} |z_k\rangle$$

with $|z_k\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \omega_{2^n}^{x \cdot 2^k}|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i (x_0 2^{k-n} + \ldots + x_{n-1-k} 2^{-1})}|1\rangle)$.

Let R_ℓ be the single-qubit rotation operator

$$R_\ell := \begin{pmatrix} 1 & 0 \\ 0 & \omega_{2^\ell} \end{pmatrix}.$$

Then $\mathcal{F}_{\mathbb{Z}_{2^n}}$ can be implemented by $O(n^2)$ Hadamard gates and controlled rotation gates.

► **Phase Estimation**. We need a tool, called *phase estimation* (PhE) to proceed. Kitaev [**?**] proposed the PhE problem and gave an efficient quantum algorithm for it. It turns out to be an extremely useful primitive, as we will see many applications later. (I personally feel that its potential power is not yet fully exploited.)

The PhE problem is defined as follows. Given a unitary operator U and an eigenstate $|\psi\rangle$ of U such that $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle, \theta \in [0,1)$. We want to compute (a rational approximation of) θ.

P.E circuit

1. prepare the state: $\sum_{x \in \mathbb{Z}_{2^q}} |x\rangle|\psi\rangle$.
2. apply the operator $\sum_{x \in \mathbb{Z}_{2^q}} |x\rangle\langle x| \otimes U^x$ and get

$$\frac{1}{\sqrt{2^q}} \sum_{x \in \mathbb{Z}_{2^q}} e^{2\pi i \theta x} |x\rangle|\psi\rangle$$

3. apply inverse Fourier transform $\mathcal{F}_{\mathbb{Z}_{2^q}}^{-1}$ (run the circuit above backward) on the 1st register, we get

$$\frac{1}{2^q} \sum_{y \in \mathbb{Z}_2^n} \sum_{x \in \mathbb{Z}_2^q} \omega_{2^q}^{(2^q\theta - y)x} |y\rangle|\psi\rangle \,.$$

4. measure the 1st register and get $y \in \mathbb{Z}_2^n$. Output $\hat{\theta} := y/2^q$.

Fact: We can measure an ϵ_p-approximation $\hat{\theta}$ (i.e., $|\hat{\theta} - \theta| \leq \epsilon_p$) with probability $\Omega(1 - \epsilon_{meas})$, when $2^q = \Omega(\frac{1}{\epsilon_p} \cdot \frac{1}{\epsilon_{meas}})$.

We omit the analysis here since we will see a similar one later.

Phase estimation can be interpreted more generally. For example, consider $\{|\psi_i\rangle : i = 1, ..., k\}$ be a set of mutually orthogonal states which are all eigenvectors of U with $U|\psi_i\rangle = e^{2\pi i \theta_i}|\psi_i\rangle$. Now let $|\varphi\rangle = \sum_i \alpha_i |\psi_i\rangle \in \mathrm{span}(|\psi_1\rangle, \ldots, |\psi_k\rangle)$ be some state. What happens if we input $|\varphi\rangle$ in the PhE algorithm? By linearity, we are basically picking a random i according to $\{p_i := |\alpha_i|^2\}$ and running PhE on $|\psi_i\rangle$. In other words, a projective measurement is effectively made on $|\varphi\rangle$ under $\{|\psi_i\rangle\}$ and conditioned on measuring i we get an approximation of θ_i at the same time.

▶ $G = \mathbb{Z}_N$, **arbitrary** $N \in \mathbb{Z}$. Note that it suffices to implement $\mathcal{F}_{\mathbb{Z}_N}$ on every basis vector

$$|x\rangle \mapsto |\tilde{x}\rangle := \mathcal{F}_{\mathbb{Z}_N}|x\rangle = \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle \, .$$

It is straightforward to implement $|x, 0\rangle \mapsto |x, \tilde{x}\rangle$. But we need to uncompute x to implement $|x\rangle \mapsto |\tilde{x}\rangle$. It's time to have our friend PhE to help us for the first time.

Consider $U := \sum_{x \in \mathbb{Z}_N} |x - 1 \text{ mod } N\rangle\langle x|$, we can verify that $|\tilde{x}\rangle$ is an eigenstate of U with eigenvalue $e^{2\pi i x/N}$.

$$U|\tilde{x}\rangle = \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y - 1 \text{ mod } N\rangle \overset{z:=y-1 \text{ mod } N}{=} \frac{1}{\sqrt{N}} \sum_{z \in \mathbb{Z}_N} \omega_N^{y} \omega_N^{xz} |z\rangle = e^{2\pi i x/N} |\tilde{x}\rangle \, .$$

Therefore if we feed $|0, \tilde{x}\rangle$ in the PhE circuit, we will get $|x, \tilde{x}\rangle$ (approximately). This means that if we run the PhE circuit in reverse on $|x, \tilde{x}\rangle$, we can uncompute x from $\tilde{x}$ and hence implement $\mathcal{F}_{\mathbb{Z}_N}$.

$$\boxed{\text{Quantum circuit for } \mathcal{F}_{\mathbb{Z}_N}}$$

▶ **Any finite abelian** G. Since G can be written as direct product of cyclic factors $\mathbb{Z}_{N_i}$, $\mathcal{F}_G = \otimes \mathcal{F}_{\mathbb{Z}_{N_i}}$ can be implemented by tensor product of QFT over each cyclic factor.