COL7160 : Quantum Computing

Lecture 17: Hidden Subgroup Problem and Discrete Logarithms

Instructor: Rajendra Kumar **Scribe:** Pushpraj

1 Recap

1.1 Abstract Formulation: The Hidden Subgroup Problem

Input: A group G, a finite set X, and a function $f : G \to X$ evaluated via a quantum oracle.
Promise: There exists a subgroup $H \leq G$ such that f is constant on the cosets of H and distinct on different cosets. This means $f(g_1) = f(g_2) \iff g_1H = g_2H$
Goal: Find a set of generators for the hidden subgroup H.

1.2 Discrete Logarithm Problem

Let p be a prime and consider the multiplicative group $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$ under multiplication modulo p. This is a cyclic group of order $p-1$. Let $g \in \mathbb{Z}_p^*$ be a generator. Given $h \in \mathbb{Z}_p^*$, find $x \in \mathbb{Z}_{p-1}$ such that

$$h \equiv g^x \pmod{p}.$$

2 Discrete Logarithm as an Abelian HSP

2.1 The Hiding Function

Define the group

$$G = \mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1}$$

with the operation being component-wise addition modulo $p-1$, i.e.,

$$(a, b) + (a', b') = (a + a',\ b + b') \pmod{p-1}.$$

Define a function

$$f : G \to \mathbb{Z}_p^*, \quad f(a, b) = g^a\, h^{-b} \pmod{p}.$$

Using $h = g^x$, we can rewrite:

$$f(a, b) = g^a (g^x)^{-b} = g^{a-xb} \pmod{p}.$$

Now consider when two inputs map to the same value:

$$f(a, b) = f(a', b').$$

This holds if and only if

$$g^{a-xb} \equiv g^{a'-xb'} \pmod{p}.$$

Since g is a generator of $\mathbb{Z}_p^*$, we obtain

$$a - xb \equiv a' - xb' \pmod{p-1}.$$

Rearranging,

$$(a - a') \equiv x(b - b') \pmod{p-1}.$$

2.2 The Hidden Subgroup

Define

$$H = \{(a, b) \in \mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1} \mid a - xb \equiv 0 \pmod{p-1}\}.$$

Equivalently,

$$H = \{(xb, b) \mid b \in \mathbb{Z}_{p-1}\}.$$

One can verify that H is a subgroup of G. Moreover, H is cyclic and generated by $(x, 1)$:

$$H = \langle (x, 1) \rangle.$$

From the equality condition,

$$f(a, b) = f(a', b') \iff (a, b) - (a', b') \in H.$$

The function f hides the subgroup

$$H = \langle (x, 1) \rangle \leq \mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1}.$$

Thus, solving the discrete logarithm problem reduces to finding the hidden subgroup H, which in turn reveals the unknown value x.

Note : In a general setting , we attempt to define a function over an Abelian group such that the function hides a subgroup. The problem then reduces to solving HSP on that group.

3 Algorithm

3.1 State Preparation

We begin with the initial state $|0, 0, 0\rangle$, where the first two registers hold the inputs $a, b \in \mathbb{Z}_{p-1}$ and the third register will store the function output. Our goal is to prepare a uniform superposition over $\mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1}$.
If $p - 1 = 2^n$, this is achieved exactly by applying Hadamard gates on n qubits in each of the first two registers:

$$H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{a=0}^{2^n-1} |a\rangle$$

Applying this to both input registers gives our desired state:

$$\frac{1}{p-1} \sum_{a,b \in \mathbb{Z}_{p-1}} |a, b, 0\rangle$$

which is a perfect uniform superposition over all of $\mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1}$.

Note : Look at discussions below when $p - 1$ is not power of 2

3.2 Function Oracle

Now we apply the Oracle over this to get

$$\frac{1}{p-1} \sum_{a,b \in \mathbb{Z}_{p-1}} |a, b, f(a, b)\rangle$$

3.3 Measurement on 3rd register

When we measure the 3rd register, we get all combinations of the first two registers that yield the same value of the function. Note that $f(a,b) = g^\delta \iff a - bx \equiv \delta \pmod{p-1}$. Hence, after measurement, our state collapses to (ignoring normalization):

$$\sum_{\substack{a,b \in \mathbb{Z}_{p-1} \\ a - bx \equiv \delta \pmod{p-1}}} |a, b, g^\delta\rangle$$

Note once we fix b, the value of a is unique under $\mathbb{Z}_{p-1}$. Hence we can rewrite as:

$$\frac{1}{\sqrt{p-1}} \sum_{b \in \mathbb{Z}_{p-1}} |\delta + bx \pmod{p-1}, b, g^\delta\rangle$$

This is precisely the coset state corresponding to $(\delta, 0) + H$.

3.4 QFT on first 2 registers

Now applying $QFT_{p-1} \otimes QFT_{p-1}$ we get (ignoring the 3rd register):

$$\frac{1}{(p-1)^{3/2}} \sum_{b \in \mathbb{Z}_{p-1}} \left(\sum_{k_1 \in \mathbb{Z}_{p-1}} \omega_{p-1}^{k_1(\delta + bx)} |k_1\rangle \right) \otimes \left(\sum_{k_2 \in \mathbb{Z}_{p-1}} \omega_{p-1}^{k_2 b} |k_2\rangle \right)$$

$$\frac{1}{(p-1)^{3/2}} \sum_{k_1 \in \mathbb{Z}_{p-1}} \sum_{k_2 \in \mathbb{Z}_{p-1}} \omega_{p-1}^{k_1 \delta} \left(\sum_{b \in \mathbb{Z}_{p-1}} \omega_{p-1}^{(k_2 + k_1 x) b} \right) |k_1, k_2\rangle$$

For a particular $|k_1, k_2\rangle$ the sum is non-zero only when $k_2 + k_1 x \equiv 0 \pmod{p-1}$

$$\frac{1}{\sqrt{p-1}} \sum_{\substack{k_1, k_2 \in \mathbb{Z}_{p-1} \\ k_2 + k_1 x \equiv 0 \pmod{p-1}}} \omega_{p-1}^{k_1 \delta} |k_1, k_2\rangle$$

Observe : The condition $k_2 + k_1 x \equiv 0 \pmod{p-1}$ corresponds to a subgroup .

3.5 Final Measurement

Now measuring the state we get k_1, k_2 that satisfy $k_2 + k_1 x \equiv 0 \pmod{p-1}$. Note that to isolate x, we need $\gcd(k_1, p-1) = 1$ to compute the modular inverse.
If this isn't the case, we run the algorithm again to collect a system of equations, repeating the measurements r times to get pairs $(k_1^1, k_2^1), \ldots, (k_1^r, k_2^r)$. We can uniquely determine the value of x provided the collective greatest common divisor is 1:

$$\gcd(k_1^1, k_1^2, \ldots, k_1^r, p-1) = 1 \quad \text{(see Exercise 1)}$$

4 Approximations

Note the weaknesses at two places in the algorithm , both requiring $p - 1 = 2^n$. The solution is to choose powers of 2 , $q = 2^m$ that are larger than $p - 1$ and then work with two input registers ranging over $\mathbb{Z}_q \times \mathbb{Z}_q$ rather than $\mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1}$.
See Exercise 2.

5 Abstract Structure of the HSP

The algorithm above is one instance of the general *abelian hidden subgroup problem* [Chi13, Chi11].

Theorem 1 (Fundamental Theorem of Finite Abelian Groups)**.** *Every finite abelian group is isomorphic to a direct product of cyclic groups.*

This theorem is the structural reason the abelian HSP is tractable. It allows us to decompose the Abelian group G into cyclic groups, and the quantum Fourier transform over G decomposes into Fourier transforms over those cyclic groups [Chi13, Chi11].
Therefore, if G is a finite abelian group of known structure, we may write

$$G \cong \mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_\ell}.$$

We now turn to examples involving non-abelian groups.

6 Graph Isomorphism:

Input: $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$
Goal: Decide whether there exists a permutation π such that

$$(i, j) \in E_1 \iff (\pi(i), \pi(j)) \in E_2.$$

6.1 Facts:

Graph Isomorphism (GI) is in NP. However, it is not known whether GI is in P, and it is also not known to be NP-complete [Bab15, dW23].
A major breakthrough due to **László Babai** shows that GI admits a quasipolynomial-time *classical* algorithm [Bab15]. Its running time is of the form

$$\exp\big((\log n)^{O(1)}\big) = n^{\mathrm{polylog}(n)}.$$

6.2 Symmetric Groups

For $n \geq 1$, the *symmetric group* S_n is the group of all permutations of

$$[n] := \{1, 2, \ldots, n\},$$

with group operation given by composition of permutations. In particular, $|S_n| = n!$.

6.3 Reduction to HSP

Graph Isomorphism can be cast as an instance of the *non-abelian* Hidden Subgroup Problem [dW23, EH99].
The relevant group is S_{2n}. See Exercise 3.
The reduction above leads to an HSP over a *non-abelian* group, for the symmetric-group case relevant to GI, no efficient quantum algorithm is known [dW23].

> Can one solve the non-abelian HSP efficiently enough to obtain an efficient quantum algorithm for Graph Isomorphism?

7 Learning With Errors (LWE)

Fix integers $n, m, q \geq 1$ and an error distribution χ over $\mathbb{Z}_q$ that is concentrated on "small" values. The *search-LWE* problem is defined as follows [Reg05].
Choose a secret vector

$$s \in \mathbb{Z}_q^n.$$

Sample a matrix

$$A \leftarrow \mathbb{Z}_q^{m \times n}$$

uniformly at random, and sample an error vector

$$e \leftarrow \chi^m.$$

Given

$$(A, b), \qquad \text{where } b = As + e \pmod{q},$$

the goal is to recover the hidden vector s.
Equivalently, one may view the input as m independent samples

$$(a_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q,$$

where each a_i is chosen uniformly from $\mathbb{Z}_q^n$ and

$$b_i = \langle a_i, s \rangle + e_i \pmod{q},$$

with $e_i \leftarrow \chi$.

Remark. The standard goal in *search-LWE* is to recover the secret s. There is also a *decision-LWE* variant, where one must distinguish LWE samples from uniformly random samples in $\mathbb{Z}_q^n \times \mathbb{Z}_q$ [Reg05].

8 Dihedral Group

For $N \geq 3$, the *dihedral group* D_N is the group of symmetries of a regular N-gon. It has $|D_N| = 2N$ elements .
It has two operations rotations and reflection , where s is rotation by $2\pi/N$ and r is reflection about a fixed axis.

8.1 The group D_6

The symmetry group of a regular hexagon is D_6, with $|D_6| = 12$:

$$D_6 = \{e,\ s,\ s^2,\ s^3,\ s^4,\ s^5,\ r,\ sr,\ s^2r,\ s^3r,\ s^4r,\ s^5r\}.$$

Label the vertices 1–6 clockwise from the top, as in Figure 1. The two generators act as:

- s: rotation by 60° clockwise, sending $1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 1$.
- r: reflection about the vertical axis, fixing 1 and 4, and swapping $2 \leftrightarrow 6$ and $3 \leftrightarrow 5$.

Normal form. Every element of D_N has a unique *normal form*

$$s^x r^a, \qquad x \in \mathbb{Z}_N,\ a \in \{0, 1\},$$

So we can identify elements with pairs $(x, a) \in \mathbb{Z}_N \times \mathbb{Z}_2$. Every sequence of r, s operations can be written as $s^x r^a$.

Example 2. Try to find a, x for this sequence of operation rsr . You should get $s^5 = rsr$.

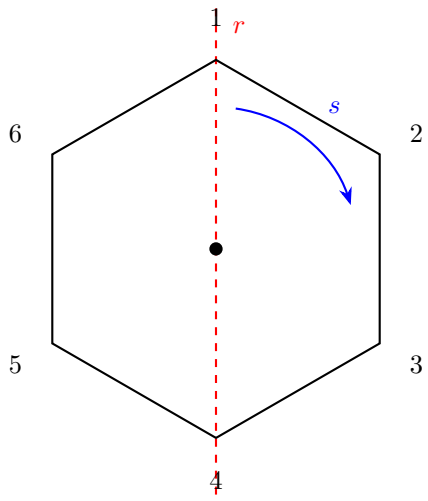

Figure 1: Regular hexagon with vertices 1–6 labeled clockwise. Blue: s rotates clockwise by $60°$. Red dashed: axis of reflection r (through vertices 1 and 4).

8.2 Subgroups of D_N

There are three families of subgroups.

Rotation subgroups. The powers of s^k form a cyclic subgroup consisting entirely of rotations:

$$\langle s^k \rangle = \{e,\ s^k,\ s^{2k}, \ldots\}.$$

Reflection subgroups. Any element of the form $s^y r$ has order 2 (since $(s^y r)^2 = e$), generating a two-element subgroup:

$$\langle s^y r \rangle = \{e,\ s^y r\}.$$

Dihedral subgroups. A rotation s^k together with a reflection $s^y r$ generate a subgroup isomorphic to a smaller dihedral group:

$$\langle s^k,\ s^y r \rangle = \{s^{jk} : j \in \mathbb{Z}\} \cup \{s^{y+jk} r : j \in \mathbb{Z}\}.$$

8.3 Connection to the Hidden Subgroup Problem

The *dihedral hidden subgroup problem* asks us to determine a hidden subgroup of D_N given it's function oracle . A standard reduction shows that it suffices to consider the case where the hidden subgroup is of the form

$$H = \langle (y, 1) \rangle = \{(0, 0), (y, 1)\},$$

for some unknown $y \in \mathbb{Z}_N$ [Kup05, Reg04a, Chi25]. See Exercise 4 for why Simon's algorithm does not apply directly.
The best known quantum algorithms for the dihedral HSP are due to Kuperberg and Regev; they run in subexponential time, but no polynomial-time quantum algorithm is known [Kup05, Reg04b].

9 Exercises

Exercise 1 (Repeated measurements for discrete logarithms)**.** Let $N = p - 1$. Recall that each run of the algorithm outputs a uniformly random pair $(k_1, k_2) \in \mathbb{Z}_N^2$ satisfying

$$k_2 + k_1 x \equiv 0 \pmod{N}.$$

(a) Show that if $\gcd(k_1, N) = 1$, then x is uniquely determined modulo N from a single sample. Deduce that the single-run success probability is

$$\Pr[\gcd(k_1, N) = 1] = \frac{\varphi(N)}{N}.$$

(b) After s independent runs, let

$$d_s = \gcd(k_1^{(1)}, \ldots, k_1^{(s)}, N).$$

Show that

$$\Pr[d_s > 1] \leq \omega(N)\, 2^{-s},$$

where $\omega(N)$ is the number of distinct prime divisors of N, and conclude that $s = O(\log \log p)$ repetitions suffice to make the failure probability bounded away from 1.

Exercise 2 (Power-of-two input registers)**.** Let $q = 2^m$ with $q > p - 1$, and replace the idealized input registers over $\mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1}$ by registers over $\mathbb{Z}_q \times \mathbb{Z}_q$.

(a) Explain why this choice lets us prepare the initial superposition exactly using Hadamard gates, while the oracle can still be defined using the residues of a and b modulo $p - 1$.

(b) After measuring the output register, describe the resulting periodic superposition on the first two registers and explain why applying $QFT_q \otimes QFT_q$ still yields enough information to recover x by classical post-processing.

Exercise 3 (Graph isomorphism as an HSP). Let G_1 and G_2 be graphs on n vertices. You have to show that Graph Isomorphism can be reduced to a hidden subgroup problem over a non-abelian group. Define appropriate function for reduction and the subgroup it hides. Explain how solving the resulting HSP would let us decide whether $G_1 \cong G_2$.

Exercise 4. Suppose a function over D_N hides the subgroup

$$H = \langle (y, 1) \rangle = \{(0, 0), (y, 1)\}$$

Explain why simon's algorithm won't work here ?

References

[Bab15] László Babai. Graph isomorphism in quasipolynomial time, 2015.

[Chi11] Andrew M. Childs. Lecture 4: The abelian hsp and decomposition of abelian groups, 2011. Quantum Algorithms course notes, University of Waterloo.

[Chi13] Andrew M. Childs. Lecture 3: The abelian hidden subgroup problem, 2013. Quantum Algorithms course notes, University of Waterloo.

[Chi25] Andrew M. Childs. Lecture notes on quantum algorithms, 2025. Available at the University of Maryland course webpage.

[dW23] Ronald de Wolf. Quantum computing: Lecture notes, 2023.

[EH99] Mark Ettinger and Peter Høyer. A quantum observable for the graph isomorphism problem, 1999.

[Kup05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005.

[Reg04a] Oded Regev. Quantum computation and lattice problems. *SIAM Journal on Computing*, 33(3):738–760, 2004.

[Reg04b] Oded Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space, 2004.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, pages 84–93, 2005.