CPSC 436Q Project

Marcus Lai, Vincent Ling, Rain Zimin Yang
University of British Columbia

Abstract

This paper surveys the current progress on the Hidden Subgroup Problem (HSP), a fundamental question in quantum computing. The work begins by exploring the Quantum Fourier Transform (QFT) and its role in solving the HSP. It then discusses both classical and quantum approaches for solving the HSP, with a focus on the Abelian HSP. Several problems, such as Simon's Problem, the Period Finding Problem, the Discrete Logarithm Problem (DLP), and its elliptic curve variant (ECDLP), are shown to reduce to the HSP. These reductions provide the foundation for quantum algorithms, such as Shor's algorithm, to break encryption protocols like RSA, Diffie-Hellman, and Elliptic Curve Diffie-Hellman. Finally, the paper briefly discusses solving specific cases of the HSP on the Dihedral Group.

1 Introduction

The Hidden Subgroup Problem (HSP) is a fundamental problem in quantum computing, underpinning many significant quantum algorithms. Its importance lies in its ability to model computational problems that are hard to solve with conventional classical computer but quantumly efficient. The HSP generalizes problems such as Simon's Problem and the Period Finding Problem. These reductions have profound implications for cryptography, as they enable quantum attacks on some protocols like RSA, Diffie-Hellman, and elliptic curve cryptography.

In Section 2, we introduce the Quantum Fourier Transform (QFT) and explore its key applications, particularly its pivotal role in quantum algorithms. In Section 3, we review fundamental definitions from group theory and present the well-known classical and quantum approaches for solving the Abelian Hidden Subgroup Problem (HSP). In Section 4, we discuss several common problems that can be reduced to the Abelian HSP, such as Simon's Problem, Period Finding, the Discrete Logarithm Problem (DLP), and the Elliptic Curve Discrete Logarithm Problem (ECDLP), highlighting their implications for real-world applications, particularly in cryptographic protocols. Finally, in Section 5, we briefly examine approaches for solving the HSP in the non-Abelian setting, with an emphasis on the dihedral group.

2 The Quantum Fourier Transform

The Fourier Transform, classically used to analyze the frequency components of signals, is a cornerstone of signal processing and data analysis. It turns out that the Discrete Fourier Transform (DFT) can be implemented efficiently with a quantum circuit using the Quantum Fourier Transform (QFT), which exponentially speeds up certain computational tasks. Furthermore, it serves as the foundation for many applications in quantum computing, such as Shor's algorithm for integer factorization and phase estimation in quantum algorithms.

Definition 2.1 (the quantum Fourier Transform)**.** *Given an orthonormal basis* $|0\rangle, |1\rangle, ..., |N-1\rangle$ *for some* $N \in \mathbb{N}$*, the quantum Fourier Transform (QFT) is the unitary matrix* Q *such that* $\forall j \in [N]$*,*

$$Q|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle . \tag{1}$$

Below, we present one application of the QFT.

2.1 Phase Estimation

Phase Estimation is an elementary but important application of the quantum Fourier Transform. Given a unitary matrix U with eigenvector $|u\rangle$, the algorithm seeks to find the eigenvalue $e^{2\pi i\psi}$ of $|u\rangle$. Note that it is well known that eigenvalues of unitary matrices have norm 1, thus any eigenvalue $v \in \mathbb{C}$ of U is expressible as $e^{2\pi i\psi_v}$ for some $\psi_v \in [0, 1)$.

The idea of the algorithm is to exploit the properties of eigenvalues to construct the post Fourier transformation of a particular basis vector, then perform the inverse Fourier transform (which we know exist since the QFT is invertible as an unitary operation).

Consider first applying a Hadamard gate to $|0\rangle$ then performing a controlled-U operation from this bit to $|u\rangle$. Then

$$|0\rangle|u\rangle \xrightarrow{H} \frac{(|0\rangle + |1\rangle)}{\sqrt{2}}|u\rangle = \frac{(|0\rangle|u\rangle + |1\rangle|u\rangle)}{\sqrt{2}}$$

$$\xrightarrow{controlled-U} \frac{(|0\rangle|u\rangle + |1\rangle U|u\rangle)}{\sqrt{2}} = \frac{(|0\rangle|u\rangle + e^{2\pi i\psi}|1\rangle|u\rangle)}{\sqrt{2}} = \frac{(|0\rangle + e^{2\pi i\psi}|1\rangle)}{\sqrt{2}}|u\rangle .$$

In particular, we note that applying controlled-U does not transform $|u\rangle$, allowing us to reuse $|u\rangle$ throughout our procedure. By reproducing this sequence with U^{2^j} for $j \in [n]$, we get

$$\frac{(|0\rangle + e^{2\pi ij\psi}|1\rangle)}{\sqrt{2}}|u\rangle .$$

By aggregating the result of this sequence for all $j \in [n]$, we obtain (excluding the $|u\rangle$ register which stays constant)

$$(\frac{(|0\rangle + e^{2\pi i2^{n-1}\psi}|1\rangle)}{\sqrt{2}})(\frac{(|0\rangle + e^{2\pi i2^{n-2}\psi}|1\rangle)}{\sqrt{2}})...(\frac{(|0\rangle + e^{2\pi i2^{0}\psi}|1\rangle)}{\sqrt{2}}) \tag{2}$$

$$= \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi ik\psi}|k\rangle$$

To see the last equality, notice when we distribute the products in Eq. (2), the jth term multiplied yields $|0\rangle$ or $e^{2\pi j2^j\psi}$. We notice this is precisely the quantum Fourier Transform of $|j\rangle$ such that $j/2^n = \psi$. So we can simply invert the transformation, measure to obtain j, and infer ψ. Notice however since $j \in [n]$, there only exist a finite number of ψ that we can infer. Namely, given a fixed n, we do not have arbitrary precision on ψ.

3 The Hidden Subgroup Problem

The hidden subgroup problem generalizes many computational problems, such as factoring, discrete logarithm, and graph isomorphism, into a unified framework in the context of group theory. Before stating the statement of the problem, we need some notation and definitions.

3.1 Preliminaries

Definition 3.1 (Abstract Group)**.** *A group is an ordered pair $(G, *)$ where G is a set and $* : G \times G \to G$ is a binary operation on G that satisfies the following:*

1. *Associativity: $(a * b) * c = a * (b * c)$ for all $a, b, c \in G$,*
2. *Identity: there exists an element $e \in G$ such that $g * e = e * g = g$ for all $g \in G$,*
3. *Inverse: for every $g \in G$, there exists is an element g^{-1} such that $g * g^{-1} = g^{-1} * g = e$.*

We say $(H, *)$ is a subgroup of $(G, *)$ if $H \subseteq G$, and $(H, *)$ is a group.

Note that an abstract group doesn't have to be commutative, that is, for $a, b \in G$, we don't necessarily have $a * b = b * a$. When $a * b = b * a$ for all $a, b \in G$, we say $(G, *)$ is *abelian*.

Definition 3.2 (Coset)**.** *Let H be a subgroup of G, we define the **left cosets** of G with respect of H as the sets $gH = \{gh, h \in H\}$, note that we could define **right cosets** in similar fashion*

Definition 3.3 (Subgroup Generated by Set)**.** *Let G be a group, and let $S \subseteq G$ be a subset of G. The **subgroup generated by** S, denoted $\langle S \rangle$, is the smallest subgroup of G that contains all elements of S.*

*If $S = \{g\}$ contains only a single element, $\langle g \rangle$ is called the **cyclic subgroup generated by** g.*

Definition 3.4 (Group Homomorphism)**.** *Let $(G, *_G)$, $(H, *_H)$ be groups, a map $\varphi : G \to H$ is called a **group homomorphism** if $\varphi(g_1 *_G g_2) = \varphi(g_1) *_H \varphi(g_2)$ for every $g_1, g_2 \in G$. When φ is a bijection, we say φ is an **isomorphism**, and write $G \cong H$.*

For more information about groups, we refer to the textbook [DF03].

We will mainly focus on finite groups, that is, when $|G|$ is finite.

Theorem 3.5 (Classification of Finite Abelian Group)**.** *Every finite abelian group is a direct product of cyclic groups. That is,*

$$G \cong \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_k},$$

where each $\mathbb{Z}_{n_i}$ is a cyclic group of order $n_i \in \mathbb{N}$.

We now present the general definition of the Hidden Subgroup Problem.

Definition 3.6 (The Hidden Subgroup Problem)**.** *A function $f : G \to X$ from a group G to a finite set X is said to hide a subgroup H of G if $f(x) = f(y) \Leftrightarrow x$ and y are in the same coset of H, that is, $x - y \in H$. Given this function f, find a subset of H such that every element of H is a finite product of elements from this set, i.e., find a generating subset of H.*

The finite abelian hidden subgroup problem has been completely solved.

We conclude this section with a result that guarantees the correctness of the quantum algorithm.

Theorem 3.7. *Let G be a finite group, and $t \geq 0$ be an integer. Then the probability that $t + \lceil \log_2 |G| \rceil$ uniformly random chosen elements from G generates the whole group with probability at least $1 - \frac{1}{2^t}$.*

For the proof of this theorem, we refer to [Pak].

3.2 Classical Approach

The best known randomized approach achieves an expected query complexity of $O(\sqrt{n/m})$ where $n = |G|$ is the order of the group and $m = |H|$ is the size of the hidden subgroup, with a worst case complexity of $O(\sqrt{n})$ [YL22][Nay22].

In this subsection, we present the nonconstructive proof for a classical HSP algorithm by Nayak [Nay22]. This approach relies on the idea that any finite abelian group G could be expressed as the product of two subsets S_1, S_2 of G, or a *generating pair*, such that $|S_1|, |S_2| \in O(\sqrt{n})$. Indeed, given such a decomposition, we have $\forall h \in H \subseteq G, h = s_1 s_2$ such that $s_1 \in S_1$, $s_2 \in S_2$, so we can simply query f at $s_1^{-1}, \forall s_1 \in S_1$ and all $s_2 \in S_2$. Then $f(s_1^{-1}) = f(s_2) \implies h = s_1 s_2 \in H$. We further notice that any $h \in H \subseteq G$ has form $s_1 s_2$ so we can construct H with the algorithm. Since we are querying two sets of size $O(\sqrt{n})$, the algorithm yields a deterministic query complexity of $O(\sqrt{n})$, matching the worst case query complexity of the randomized case.

For the non-abelian case, a similar algorithm is possible but is limited by how $|S_1|, |S_2|$ can be only shown to be $\in O(\sqrt{n \log n})$. We demonstrate the proof here.

Proposition 3.8 (Generating Pair of Non-Abelian Groups [Nay22])**.** *For any group G with order $n > 1$, there is a generating pair S_1, S_2 for G such that $|S_1|, |S_2| \leq \lceil \sqrt{n \log n} \rceil$.*

Proof. Let $t := \sqrt{n \log n}$ and fix any $S_1 = \{g_1, g_2, ..., g_t\} \subseteq G$. Let R be a set of t distinct group elements chosen uniformly and randomly from the collection of subsets of G with size t. Consider their product

$$S_1 R = \{xy : x \in S_1, y \in R\}.$$

Now, fix an element $g \in G$. Then the probability that $g \notin S_1 R$ is

$$\begin{aligned}
\Pr[g \notin S_1 R] &= \Pr[\forall i \in [t], g_i^{-1} g \notin R] \\
&= \frac{\text{for our fixed } g, \#\text{ of ways to make } R \text{ without any of } g_i^{-1} g \text{ (t such elements)}}{\text{all possible } R} \\
&= \frac{\binom{n-t}{t}}{\binom{n}{t}} = \frac{\frac{(n-t)!}{(n-2t)!t!}}{\frac{n!}{(n-t)!t!}} = \frac{(n-t)(n-t-1)...(n-2t+1)}{n(n-1)...(n-t+1)} = \prod_{k=0}^{t} \frac{n-t-k}{n-k} \\
&= \prod_{k=0}^{t} \left(1 - \frac{t}{n-k}\right)
\end{aligned}$$

since $t > 1$,

$$\Pr[g \notin S_1 R] < \left(1 - \frac{t}{n}\right)^t \leq e^{-\frac{t^2}{n}} \leq \frac{1}{n}$$

with the second to last inequality due to $1 - x \leq e^{-x}$, and the last equality exploiting $t = \sqrt{n \log n}$. Thus,

$$\Pr[G \neq S_1 R] = \Pr[G \not\subseteq S_1 R] = \Pr[\exists g \in G, g \notin S_1 R] \leq \sum_{i=1}^{n} \Pr[g \notin S_1 R] < 1$$

by the union bound. Since this probability is strictly less than 1, there must exist some S_2 with $t = \sqrt{n \log n}$ elements such that $S_1 S_2 = G$ and we are done.

□

Thus, the same deterministic algorithm would give us query complexity $O(\sqrt{n \log n})$.

3.3 Quantum Approach

In this subsection, we present the quantum algorithm for solving the abelian hidden subgroup problem. The following is discussion based on [CVD10], [Chi], and [Lom04].

Throughout this section, let G denote a finite abelian group, and H a subgroup of G hidden by a function $f : G \to X$. By the structure theorem of finite abelian group, G can be expressed as $G \cong \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_k}$ for some $k \in \mathbb{N}$, where $n_i \geq 2$, $n_i \in \mathbb{N}$. Elements of G can be represented as a k-tuple $g = (g_1, g_2, \ldots, g_k) \in G$, where each $g_i \in \mathbb{Z}_{n_i}$.

If we examine the algorithm for finding hidden subgroup $\mathbb{Z}_r$ in $\mathbb{Z}_n$, the quantum procedure samples elements from the quotient group $\mathbb{Z}_n/\mathbb{Z}_r$, and by computing the greatest common divisor of these coset representatives, we can find the unknown r with high probability.

The approach generalized to finite abelian groups. By generating sufficiently many group elements, the hidden subgroup can be reconstructed from this data. In the cyclic case, the elements generated are multiples of the subgroup generator (which corresponds to a subgroup of $\mathbb{Z}_n$ isomorphic to $\mathbb{Z}_n/\mathbb{Z}_r$), and this can be generalized to a subgroup of G isomorphic to G/H.

To generalize the Quantum Fourier Transform to all finite abelian groups, we need group representation theory.

Definition 3.9 (Character)**.** *For any $g, h \in G$, define χ_g to be a map from G to $\mathbb{C}^*$ (the set of nonzero complex numbers) via*

$$\chi_g(h) = \prod_{i=1}^{k} e^{\frac{2\pi i}{n_i} g_i h_i}.$$

One can check that the map χ_g is a group homomorphism.

For more details on the theory of characters, we refer to the textbook [Lan02].

Definition 3.10 (QFT over a Finite Abelian Group)**.** *The Quantum Fourier Transform over G is the unitary matrix*

$$\mathrm{QFT}_G := \bigotimes_{i=1}^{k} \mathrm{QFT}_{n_i},$$

where QFT_{n_i} is the cyclic Quantum Fourier Transform.

This Quantum Fourier Transform is essentially taking the Fourier Transform over each cyclic component of the abelian group. Assuming that the matrix is indexed by the group elements, with $g = (g_1, g_2, \ldots, g_k)$, one can verify the following identity:

$$\mathrm{QFT}_G = \frac{1}{\sqrt{|G|}} \sum_{g,h \in G} \chi_g(h) |g\rangle\langle h|.$$

Definition 3.11 (Orthogonal Subgroup)**.** *For any subgroup H of G, define the orthogonal subgroup*

$$H^\perp = \{g \in G : \chi_g(h) = 1 \ \forall h \in H\}.$$

One can check that this is a subgroup of G isomorphic to G/H, and the correspondence $H \to H^\perp$ is one-to-one, with inverse given by $(H^\perp)^\perp = H$.

For each subset S of G, denote

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

The following theorem shows the relation between the Quantum Fourier Transform and the orthognoal subgroup.

Theorem 3.12.

$$\mathrm{QFT}_G \left|H\right\rangle = \left|H^{\perp}\right\rangle .$$

Definition 3.13. *For $t \in G$, the translation τ_t and phase-change operators are defined by*

$$\tau_t = \sum_{g \in G} |t+g\rangle\langle g|, \;\; \phi_t = \sum_{g \in G} \chi_t(g)|g\rangle\langle g|.$$

The following lemma shows why the Quantum Fourier Transform is useful for our purpose, and one can verify it by a direct computation.

Lemma 3.14. *For all $t \in G$,*

$$\mathrm{QFT}_G \, \tau_t = \phi_t \, \mathrm{QFT}_G .$$

We now present the algorithm. First, consider the follow quantum procedure that randomly samples an element of $H^{\perp}$.

1. Create the uniform superposition state over all the group elements,

$$\left|G\right\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} \left|g\right\rangle \left|0\right\rangle .$$

2. Using one quantum query to create

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} \left|g\right\rangle \left|f(g)\right\rangle .$$

3. Measuring and then discarding the second register gives the state

$$\left|g+H\right\rangle = \frac{1}{\sqrt{H}} \sum_{h \in H} \left|g+h\right\rangle = \tau_g \left|H\right\rangle ,$$

   for some $g \in G$ chosen uniformly at random.

4. Apply QFT_G to the previous state, and by the previous theorem, we get

$$\mathrm{QFT}_G \, \tau_g \left|H\right\rangle = \phi_t \, \mathrm{QFT}_G \left|H\right\rangle = \phi_t \left|H^{\perp}\right\rangle .$$

5. Since the phase-change operator has no effect on the norm, measuring the state in the computational basis produces an element of $H^{\perp}$ chosen uniformly at random.

As a consequence of theorem 3.7, by running the above procedure $\lceil \log_2 |G| \rceil + t$ times, we can find a generating set of $H^{\perp}$ with probability at least $1 - \frac{1}{2^t}$. The final step of the algorithm is to efficiently find a generating set of H from this generating set of $H^{\perp}$, and this can be done classically.

Let $N = \lceil \log_2 |G| \rceil + t$, and $S = \{g^{(1)}, g^{(2)}, \ldots, g^{(N)}\}$ be a set of random elements of $H^{\perp}$ chosen using the above procedure, and let $d = \mathrm{lcm}(n_1, n_2, \ldots, n_k)$, so we can write $m_j = d/n_j$. For each element $h \in H$, we have

$$\chi_{g^{(l)}}(h) = \prod_{j=1}^{k} \exp\left(\frac{2\pi i}{n_j} g_j^{(l)} h_j\right) = \prod_{j=1}^{k} \exp\left(\frac{2\pi i m_j}{d} g_j^{(l)} h_j\right) = \exp\left(\frac{2\pi i}{d} \sum_{j=1}^{k} m_j g_j^{(l)} h_j\right) = 1,$$

which is equivalent to

$$\sum_{j=1}^{k} m_j g_j^{(l)} h_j \equiv 0 \pmod{d}.$$

By the correspondence between H and $H^\perp$, we have the following characterization for H:

$$\forall g \in G,\ g \in H \Leftrightarrow \forall s \in H^\perp,\ \chi_s(g) = 1.$$

If $\langle S \rangle = H^\perp$, then every element of $H^\perp$ is a finite sum of element from S, and $\chi_s(g) = \chi_g(s)$ is a group homomorphism, so we can rewrite the characterization as

$$\forall g \in G,\ g \in H \Leftrightarrow \forall s \in S,\ \chi_s(g) = 1.$$

The above discussion shows that an element $(x_1 \pmod{n_1}, x_2 \pmod{n_2}, \ldots, x_k \pmod{n_k})$ of G is in H if and only if it's a solution to the following system of linear equations, and each solution corresponds to exactly one element of H:

$$\begin{aligned}
m_1 g_1^{(1)} x_1 + m_2 g_2^{(1)} x_2 + \cdots + m_k g_k^{(1)} x_k &\equiv 0 \pmod{d} \\
m_1 g_1^{(2)} x_1 + m_2 g_2^{(2)} x_2 + \cdots + m_k g_k^{(2)} x_k &\equiv 0 \pmod{d} \\
&\vdots \\
m_1 g_1^{(N)} x_1 + m_2 g_2^{(N)} x_2 + \cdots + m_k g_k^{(N)} x_k &\equiv 0 \pmod{d}.
\end{aligned}$$

It suffices now to randomly generate solutions to the above system to randomly sample elements from H. This process can be done efficiently using Smith normal form. For the theory on Smith normal form of matrices with entries in a principal ideal domain ($\mathbb{Z}_d$ in this case), we refer to chapter 3 of the textbook [Jac09].

Let A be the coefficient matrix of the above system, then A has Smith normal form $A = UDV$ where U, D, V are matrices with entries in $\mathbb{Z}_d$, with U, V being invertible $N \times N, k \times k$ matrices, and D a diagonal $N \times k$ matrix. The matrices U, D, V can be computed efficiently from A, and we refer to the article [Sto96].

A random solution to the equation $Dy = 0$ for $y = (y_1, y_2, \ldots, y_k)$ can be generated by solving congruences for each entry. Let $x = V^{-1}y$, then we have $Ax = UDVx = UDVV^{-1}y = 0$, so x is a solution to the above system chosen uniformly at random. Now repeating the process $N = \lceil \log(|G|) \rceil + t$ times would produce N elements of H chosen uniformly at random. Therefore the two procedures together produces a generating set for H with probability at least $\left(1 - \frac{1}{2^t}\right)\left(1 - \frac{1}{2^t}\right)$, which can be made arbitrarily small by choosing t to be a large constant.

Overall the algorithm runs polynomial time in $\log(|G|)$, with $O(\log(|G|))$ oracle calls.

4 Application of Hidden Subgroup Problem

4.1 Simon's Problem

Definition 4.1 (Simon's Problem)**.** *Let $f : \mathbb{F}_2^n \to X$ be a function, where X is an arbitrary set, such that:*

$$f(x) = f(y) \iff x \oplus y = s,$$

where $s \in \mathbb{F}_2^n$ is a fixed, unknown bitstring with $s \neq 0^n$.

The goal of Simon's Problem is to determine the hidden bitstring s using the smallest number of queries to f.

Theorem 4.2. *Simon's problem is a special case of the Hidden Subgroup Problem with $G = \{\mathbb{F}_2^n, \oplus\}$, i.e., G is a group of binary strings under the XOR operation. We need to find the hidden subgroup $H = \langle s \rangle$*

4.2 Period Finding Problem

Definition 4.3 (Period Finding Problem)**.** *Let $f : \mathbb{Z} \to X$ be a function, where X is a set, and $f(x)$ is periodic. That is, there exists a smallest positive integer r (called the **period** of f) such that:*

$$f(x) = f(x + r) \quad \textit{for all } x \in \mathbb{Z}.$$

The goal of Period Finding Problem is to find the smallest integer r that satisfies above condition

Theorem 4.4. *The Period Finding Problem is a special case of the Hidden Subgroup Problem with $G = \{\mathbb{Z}, +\}$. The goal is to find the smallest r that generates the hidden subgroup $H = \langle r \rangle$, where r corresponds to the period of the function $f(x)$.*

Definition 4.5 (Order Finding Problem)**.** *Let a and N be positive integers such that $\gcd(a, N) = 1$. The goal of the Order Finding Problem is to find the smallest positive integer r such that $a^r \equiv 1 \mod N$.*

Theorem 4.6. *Order Finding Problem is a special case of Period Finding Problem where*

$$f(x) = a^x \mod N$$

Proof. By the definition of the order, r is the smallest positive integer such that:

$$a^r \equiv 1 \mod N.$$

we have:

$$f(x + r) = a^{x+r} \mod N.$$

Using the properties of exponents, we write:

$$a^{x+r} = a^x \cdot a^r.$$

Since $a^r \equiv 1 \mod N$, it follows that:

$$a^{x+r} \equiv a^x \cdot 1 \equiv a^x \mod N.$$

Therefore:

$$f(x + r) = f(x).$$

Therefore, since r is the smallest positive integer such that $a^r \equiv 1 \mod N$, it is the smallest period of $f(x)$. Thus, the Order Finding Problem, which involves finding r, is a special case of the Period Finding Problem. □

Order Finding problem is an important element in Shor's Factoring Algorithm

4.3 Discrete Logarithm Problem

Definition 4.7. *(Discrete Logarithm Problem) Let $G = \langle g \rangle$ be a cyclic group generated by g Given an element $x \in G$, the discrete logarithm of x in G with respect to g, denoted as $\log_g x$, is the smallest non-negative integer α such that $g^\alpha = x$. The discrete logarithm problem is the problem of calculating $\log_g x$. [Chi]*

Theorem 4.8. *Let $f : \mathbb{Z}_N \times \mathbb{Z}_N \to G$ such that*

$$f(a,b) = x^a g^b = g^{a \log_g x} g^b = g^{a \log_g x + b}.$$

We can see that the function f hide

$$H = \langle (1, -\log_g x) \rangle$$

Proof. let $n \in \mathbb{Z}_N$, then

$$f(a,b) = g^{a \log_g x + b} = g^{a \log_g x + b} g^{n \log_g x - n \log_g x} = g^{(n+a) \log_g x + (b - n \log_g x)} = f(a+n, b - n \log_g x)$$

therefore $\forall c \in \mathbb{Z}_N \times \mathbb{Z}_N, \forall r \in H, f(c) = f(c+r)$. Let $s \in (\mathbb{Z}_N \times \mathbb{Z}_N) \setminus H$, then $s = (\alpha, -\beta \log_g x)$ for some $\alpha, \beta \in \mathbb{Z}_N$ and $\alpha \neq \beta$, thus

$$f(c+s) = g^{a \log_g x + b} g^{(\alpha - \beta) \log_g x}$$

Since $\alpha \neq \beta$, therefore $g^{(\alpha-\beta) \log_g x} \neq 1$, it follows that

$$f(c+s) \neq f(c)$$

Thus

$$f(c+r) = f(c) \iff r \in H$$

as required. Finding the generator of S □

Finding the generator of H will give us $-log_g x$ which correspond to the solution of Discrete Logarithm Problem

Definition 4.9. *(Diffie-Hellman Key Exchange) Diffie-Hellman Key Exchange is a cryptographic protocol that allowed two parties, typically referred as Alice and Bob, to share secret keys over an insecure channel. Let G be a cylic group with prime order p and generator g:*

- *Alice has a private key $a \in \mathbb{Z}_p$ and compute a public key $A = g^a$*
- *Bob has a private key $b \in \mathbb{Z}_p$ and compute a public key $B = g^b$*
- *Alice and Bob can exchange their public key A and B*
- *Both can compute shared secret key $s = g^{ab}$. In which Alice calculate A^b and Bob compute B^a*

The Diffie-Hellman key exchange protocol relies on the computational hardness of the Discrete Logarithm Problem, as breaking the protocol requires solving this problem..

4.4 Elliptic Curves

Definition 4.10. *(Elliptic Curves) Let $\mathbb{F}$ is a field and $a, b \in \mathbb{F}$, the elliptic curve $E_{a,b}$ is defined as the set points $(x, y) \in \mathbb{F}^2$ satisfying the equation*

$$y^2 = x^3 + ax + b$$

and

$$4a^3 + 27b^2 \neq 0$$

as well as special an infinity point $\mathcal{O}$. [Chi]

Theorem 4.11. *(Elliptic Curves as abelian group) Let $E_{a,b} = \{(x, y) \in \mathbb{F}^2, y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$ be an Elliptic Curves, then $E_{a,b}$ together with an operation "+" defined below, form an abelian group.*

1. *Identity Element ($\mathcal{O}$):*
   for any point $P \in E_{a,b}$, $\mathcal{O} + P = P + \mathcal{O} = P$

2. *Inverse of a Point:*
   for any element $P = (x, y) \in E_{a,b}$ we define the inverse $-P = (x, -y)$ and $P + (-P) = \mathcal{O}$

3. *Point Addition:*
   for any two distinct points $P = (x_P, y_p), Q = (x_Q, y_Q) \in E_{a,b}$ and $P \neq -Q$, we define the slope of the two point as

   $$\lambda = \frac{y_Q - y_P}{x_Q - x_P}$$

   then we define

   $$P + Q = (\lambda^2 - x_P - x_Q, \lambda(x_P - x_{P+Q}) - y_P)$$

4. *Point Doubling:*
   for any points $P = (x, y) \in \mathbb{F}$ then the slope of λ is the tangent point to the curve at P.

   $$\lambda = \frac{3x^2 + a}{2y}$$

   then we define

   $$2P = (\lambda^2 - 2x, \lambda(x_P - x_{2P}) - y_P)$$

Definition 4.12. *(Elliptic Curve Diffie-Hellman) Suppose we have an agreed upon Elliptic Curve $E_{a,b}$, then we choose an agreed upon point $g \in E_{a,b}$, suppose n is the order of g, i.e. $ng = \mathcal{O}$. Then each party(denote as Alice and Bob) can generate a private key and public key:*

- *Alice generate private key $a \in \mathbb{Z}_N$ and public key $A = ag$*
- *Bob generate private key $b \in \mathbb{Z}_N$ and public key $B = bg$*

Then both Alice and Bob could generate a secret key $S = abg$, Alice calculating aB and Bob by calculating bA

Elliptic Curve Diffie-Hellman assumes that finding the secret keys a and b is computationally hard. This is analogous to solving the Discrete Logarithm Problem. As shown in Theorem 4.8, the Discrete Logarithm Problem can be reduced to the Hidden Subgroup Problem. Therefore, Shor's algorithm can be used to efficiently find a and b.

5 Non-Abelian Case

Unlike the general Abelian Case, HSP on only certain of Non-Abelian HSP have been explored. We present some developments for HSP on the Dihedral Group, which describe all shape preserving symmetries of the n-gon.

Definition 5.1 (Dihedral Group)**.** *The dihedral group of order $2N$ is a semidirect product of two cyclic groups $\mathbb{Z}_N$, $\mathbb{Z}_2$ of order N and 2 respectively. It is isomorphic to the group*

$$D_N = \mathbb{Z}_N \rtimes_\phi \mathbb{Z}_2$$

with multiplication defined by

$$(a_1, b_1)(a_2, b_2) = (a_1 + \phi(b_1)(a_2), b_1 + b_2),$$

where ϕ is a homomorphism defined by

$$\phi : \mathbb{Z}_2 \to Aut(\mathbb{Z}_n) \text{ such that } \phi(0)(a) = a,\ \phi(1)(a) = -a$$

We note that the subgroups of the Dihedral Group could only have form $\mathbb{Z}_K \times \{0\}$, $\{0\} \times \{1\}$, or D_M for $K, M \in \mathbb{N} \cup \{0\}$. One core result of HSP on the Dihedral Group is an algorithm to obtain a generating set of the hidden subgroup with query complexity $\Theta(\log N)$. First, we present a fact.

Fact 5.2. *Let $\gamma : D_N \to R$ be a function with a hidden subgroup H where H is either trivial or $H = \{(0,0), (k_0, 1)\}$ for some $0 \leq k_0 \leq N$. Then there is a quantum algorithm using at most $89\log(N) + 7$ queries that decides which case we are in and outputs k_0 if we are in case 2 with probability at least $1 - \frac{1}{2N}$. [EH99]*

Now, we demonstrate the main theorem of the Dihedral HSP. The technique used in the algorithm exploits the abelian property of the cyclic $\mathbb{Z}^N$ and $\mathbb{Z}^2$ finite groups.

Theorem 5.3 (Dihedral HSP Query Complexity [EH99])**.** *Let $\gamma : D_N \to R$ be a function that hides the subgroup H. Then there exists an algorithm that uses $\Theta(\log N)$ evaluations of γ and outputs a subset $X \subseteq D_N$ such that X is a generating set for H with probability $\geq 1 - \frac{2}{N}$.*

Proof. First, we consider our function γ constrained on the group $\mathbb{Z}_N \times \{0\} \leq D_N$ of order N. We also define $H_1 = H \cap (\mathbb{Z}_N \times \{0\})$. Since $H_1 \subseteq H$, γ, and thus γ_1, is constant on the cosets of H_1. In other words, γ_1 hides the subgroup H_1 so we have an abelian HSP question (indeed $\mathbb{Z}^n \times \{0\}$ is abelian since it is isomorphic to $\mathbb{Z}^n$).

So we apply results from **Section 3.** to obtain a generating set for H_1 in $\Theta(\log N)$ evaluations of γ with probability $1 - \frac{1}{|\mathbb{Z}_N \times \{0\}|} = 1 - \frac{1}{N}$. To generate H_1 efficiently, we note that we can employ the period finding algorithm [Chi] since H_1 is cyclic as a subgroup of the cyclic group $\mathbb{Z}_N \times \{0\}$ and achieve arbitrary precision. Since we are only concerned with query complexity, we can alternatively generate H_1 with the brute force approach as well.

Now, we note that $H_1 = \mathbb{Z}_K \times \{0\}$ for some $K \in \mathbb{N}$ is normal subgroup. To show this, suppose $g \in D_N = (a_1, b_1)$ and let $(\alpha, 0) \in H_1$. Then

$$\begin{aligned}
&(a_1, b_1)(\alpha, 0)((a_1, b_1))^{-1} \\
=&(a_1 + \alpha, b_1)((a_1, b_1))^{-1} \\
=&(\alpha + a_1, b_1)((a_1, b_1))^{-1} \to (\mathbb{Z}_n \text{ abelian}) \\
=&(\alpha, 0)(a_1, b_1)((a_1, b_1))^{-1} \\
=&(\alpha, 0) \in H_1
\end{aligned}$$

Since H_1 is normal, the quotient group D_N/H_1 is well defined. Note also that as a subgroup of $Z^N \times \{0\}$, H_1 must have cyclic form $\{0, M, 2M, ..., kM\} \times \{0\}$ such that $M|N$, for $1 \leq M \leq N$, $k \in \mathbb{N}$. Now, we notice that D_N/H_1 is isomorphic to D_M. To show this, consider the homomorphism $\psi : D_N \to D_M$ defined by

$$(a, b) \to (a \pmod M, b).$$

We check that it is a homomorphism.

$$\begin{aligned}\psi((a_1, b_1) + (a_2, b_2)) =& (a_1 + \phi(b_1)(a_2) \pmod M, b_1 + b_2) \\ =& (a_1 \pmod M + \phi(b_1)(a_2) \pmod M, b_1 + b_2)\end{aligned}$$

Case 1: $b_1 = 0 \to \phi(b_1)(a) = a$

$$\begin{aligned}\psi((a_1, b_1) + (a_2, b_2)) =& (a_1 + \phi(b_1)(a_2) \pmod M, b_1 + b_2) \\ =& (a_1 \pmod M + a_2 \pmod M, b_1 + b_2) \\ =& (a_1 \pmod M + \phi(b_1)(a_2 \pmod M), b_1 + b_2) \\ =& \psi((a_1, b_1)) + \psi((a_2, b_2))\end{aligned}$$

Case 2: $b_1 = 1 \to \phi(b_1)(a) = -a$

$$\begin{aligned}\psi((a_1, b_1) + (a_2, b_2)) =& (a_1 + \phi(b_1)(a_2) \pmod M, b_1 + b_2) \\ =& (a_1 \pmod M - a_2 \pmod M, b_1 + b_2) \\ =& (a_1 \pmod M + \phi(b_1)(a_2 \pmod M), b_1 + b_2) \\ =& \psi((a_1, b_1)) + \psi((a_2, b_2))\end{aligned}$$

Now, the kernel of ψ is precisely the elements $(a, b) \in D_N$ such that $M \mid a$ and $b = 0$, which is exactly the elements of H_1. So we invoke the fundamental theorem of homomorphisms [DF03] to obtain that the quotient group D_N/H_1 is isomorphic to $\text{Image}(\psi) = D_M$.

Recall γ is constant on the cosets of H_1, so we can see γ as a function $\gamma_2 : D_M \to R$ defined via $gH_1 \mapsto \gamma(g)$ composed with the isomorphism from D_M to D_N/H_1. This is again a function that is constant on some hidden subgroup H_2. We claim that there are only two possible choices for H_2, the identity subgroup or a subgroup of order 2. We consider the second component of elements of H. If the second component of H is the identity, then we must have $H \cong H_1$, and in this case, H_1 collapse to the identity in $D_M \cong D_N/H_1$. Otherwise, H has to be isomorphic to a dihedral group with H_1 being the maximal cyclic subgroup, so $|H| = 2|H_1|$, and therefore H is the union of two cosets of H_1 in D_N/H_1, which corresponds to a subgroup of order 2 in D_M.

Thus, if we indeed obtain H_1, we repeat the algorithm in **Fact 5.2** $\lceil \log(2N)/\log(2M) \rceil = \lceil \log_{2M}(2N) \rceil$ times. This allows us to find k_0 with probability at least $1 - \frac{1}{(2M)^t} \geq 1 - \frac{1}{2N}$. Let X_1 be the generating set obtained for H_1. Then suppose we obtain k_0, we output $X = X_1 \cup \{(k_0, 1)\}$. Otherwise, we output $X = X_1$.

Note that X generates H if X_1 generates H_1 *and* if we find h_0. Thus, the probability of success is given by

$$\Pr[\text{success}] \geq (1 - \frac{1}{N})(1 - \frac{1}{2N}) = 1 + \frac{1}{2N} + \frac{1}{2N^2} > 1 - \frac{2}{N}$$

and total number of queries of γ is $\Theta(\log N) + t(89 \log M + 7) = \Theta(\log N)$ as required. □

References

[Chi] Andrew Childs. Lecture notes on Quantum algorithms. URL: https://www.cs.umd.edu/~amchilds/qa/. [pp. 5, 9, 10, 11]

[CVD10] Andrew M. Childs and Wim Van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1):6–8, 1 2010. doi:10.1103/revmodphys.82.1. [p. 5]

[DF03] D.S. Dummit and R.M. Foote. *Abstract Algebra*. Wiley, 2003. URL: https://books.google.ca/books?id=KJDBQgAACAAJ. [pp. 3, 12]

[EH99] Mark Ettinger and Peter Høyer. *On Quantum Algorithms for Noncommutative Hidden Subgroups*, page 478–487. Springer Berlin Heidelberg, 1999. URL: http://dx.doi.org/10.1007/3-540-49116-3_45, doi:10.1007/3-540-49116-3_45. [p. 11]

[Jac09] N. Jacobson. *Basic Algebra I: Second Edition*. Basic Algebra. Dover Publications, 2009. URL: https://books.google.ca/books?id=qAg_AwAAQBAJ. [p. 7]

[Lan02] Serge Lang. *Algebra*. Springer, New York, NY, 2002. [p. 5]

[Lom04] Chris Lomont. The hidden subgroup problem - review and open problems, 2004. URL: https://arxiv.org/abs/quant-ph/0411037, arXiv:quant-ph/0411037. [p. 5]

[Nay22] Ashwin Nayak. Deterministic algorithms for the hidden subgroup problem. *Quantum Information and Computation*, 22(9 10):755–769, July 2022. URL: http://dx.doi.org/10.26421/QIC22.9-10-3, doi:10.26421/qic22.9-10-3. [p. 4]

[Pak] Igor Pak. Probability of Generating a Group. URL: https://www.math.ucla.edu/~pak/courses/pg/l1.pdf. [p. 3]

[Sto96] Arne Storjohann. Near optimal algorithms for computing smith normal forms of integer matrices. In *Proceedings of the 1996 international symposium on Symbolic and algebraic computation*, pages 267–274, 1996. [p. 7]

[YL22] Zekun Ye and Lvzhou Li. Deterministic algorithms for the hidden subgroup problem. *Information and Computation*, 289:104975, 2022. URL: https://www.sciencedirect.com/science/article/pii/S0890540122001304, doi:10.1016/j.ic.2022.104975. [p. 4]