# Introduction to Quantum Computing

**George Voutsadakis**[1]

[1]Mathematics and Computer Science
Lake Superior State University

LSSU Math 500

1. Shor's Algorithm
   - Classical Reduction of Factoring to Period-Finding
   - Shor's Factoring Algorithm
   - Example Illustrating Shor's Algorithm
   - The Efficiency of Shor's Algorithm
   - Omitting the Internal Measurement
   - Generalizations

Subsection 1

## Classical Reduction of Factoring to Period-Finding

# Order of an Integer

- The **order** of an integer a modulo $M$ is the smallest integer $r > 0$ such that
$$a^r = 1 \mod M.$$
- If no such integer exists, the order is said to be infinite.
- Two integers are **relatively prime** if they share no prime factors.
- As long as $a$ and $M$ are relatively prime, the order of $a$ is finite.

# Order and Period

- For $a$ relatively prime to $M$, consider the function

$$f(k) = a^k \mod M.$$

- Note that, for $a$ relatively prime to $M$,

$$a^k = a^{k+r} \mod M \quad \text{if and only if} \quad a^r = 1 \mod M.$$

- So the order $r$ of $a$ modulo $M$ is the period of $f$.

# Reducing Factoring to Finding the Period

- Suppose $a^r = 1 \mod M$ and $r$ is even.
- Then we can write

$$(a^{r/2}+1)(a^{r/2}-1) = 0 \mod M.$$

- Suppose, further, that neither $a^{r/2}+1$ nor $a^{r/2}-1$ is a multiple of $M$.
- Then both

$$a^{r/2}+1 \quad \text{and} \quad a^{r/2}-1$$

  have nontrivial common factors with $M$.
- Thus, if $r$ is even, $a^{r/2}+1$ and $a^{r/2}-1$ are likely to have a nontrivial common factor with $M$.

# Factoring Strategy

- This property suggests a strategy for factoring $M$:
  - Randomly choose an integer $a$;
  - Determine the period $r$ of $f(k) = a^k \mod M$;
  - If $r$ is even, use the Euclidean algorithm to compute efficiently the greatest common divisor of $a^{r/2} + 1$ and $M$;
  - Repeat if necessary.
- In this way, factoring $M$ has been converted to the problem of computing the period of the function

$$f(k) = a^k \mod M.$$

- Shor's quantum algorithm attacks the problem of efficiently finding the period of a function.

Subsection 2

# Shor's Factoring Algorithm

# Shor's Factoring Algorithm: An Overview

- Quantum computation is required only for parts 2 and 3.
  1. Randomly choose an integer a such that $0 < a < M$.
     Use the Euclidean algorithm to determine whether $a$ and $M$ are relatively prime.
     - If not, we have found a factor of $M$.
     - Otherwise, apply the rest of the algorithm.
  2. Use quantum parallelism to compute $f(x) = a^x \mod M$ on the superposition of inputs.
     Then apply a quantum Fourier transform to the result.
     We will see that it suffices to consider input values $x \in \{0, \ldots, 2^n - 1\}$, where $n$ is such that $M^2 \leq 2^n < 2M^2$.
  3. Measure. With high probability, a value $v$ close to a multiple of $\frac{2^n}{r}$ will be obtained.
  4. Use classical methods to obtain from $v$ a conjectured period $q$.
  5. When $q$ is even, use the Euclidean algorithm to check efficiently whether $a^{q/2} + 1$ (or $a^{q/2} - 1$) has a nontrivial common factor with $M$.
  6. Repeat all steps if necessary.

# The Quantum Core

- We use quantum parallelism to create the superposition

$$\sum_x |x, f(x)\rangle.$$

- Then Shor's algorithm applies the quantum Fourier transform.
- The values $f(x) = a^x \mod M$ can be computed efficiently classically.
- By previous results, the transformation

$$U_f : |x\rangle|0\rangle \to |x\rangle|f(x)\rangle$$

  has an efficient implementation.
- We use quantum parallelism with $U_f$ to obtain the superposition

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|f(x)\rangle.$$

- The analysis simplifies slightly if we now measure the second register.
- We will see how the measurement can be omitted without affecting the efficiency or the result of the algorithm.

# The Quantum Core (Cont'd)

- Suppose we measure the second register randomly.
- Let $u$ be the value returned for $f(x)$.
- Then the state becomes

$$C\sum_x g(x)|x\rangle|u\rangle,$$

where

$$g(x)=\begin{cases}1, & \text{if } f(x)=u\\ 0, & \text{otherwise}\end{cases}$$

and $C$ is the appropriate scale factor.
- Here, the value of $u$ is of no interest.
- Moreover, the second register is no longer entangled with the first.
- So the second register can be ignored.

# The Quantum Core (Cont'd)

- The function
$$f(x) = a^x \mod M$$
has the property that $f(x) = f(y)$ if and only if $x$ and $y$ differ by a multiple of the period.
- So the values of $x$ that remain in the sum, i.e., those with $g(x) \neq 0$, differ from each other by multiples of the period.
- Thus, the function $g$ has the same period as the function $f$.

# The Quantum Core (Cont'd)

- If we could somehow obtain the value of two successive terms in the sum, we would have the period.
- Unfortunately, the laws of quantum physics permit only one measurement from which we can obtain only one random value of $x$.
- Repeating the process does not help because we would be unlikely to measure the same value $u$ of $f(x)$.
- So the two values of $x$ obtained from two runs would have no relation to each other.

# The Quantum Core (Cont'd)

- Apply the quantum Fourier transform to the first register of this state produces

$$U_F\left(C\sum_x g(x)|x\rangle\right) = C'\sum_c G(c)|c\rangle,$$

where

$$G(c) = \sum_x g(x)\exp\left(\frac{2\pi \boldsymbol{i} cx}{2^n}\right).$$

# The Quantum Core (Cont'd)

- A previous analysis tells us that when the period $r$ of the function $g(x)$ is a power of two, $G(c) = 0$ except when $c$ is a multiple of $\frac{2^n}{r}$.
- When the period $r$ does not divide $2^n$, the transform approximates the exact case.
- So most of the amplitude is attached to integers close to multiples of $\frac{2^n}{r}$.
- For this reason, measurement yields, with high probability, a value $v$ close to a multiple of $\frac{2^n}{r}$.
- The quantum core of the algorithm has now been completed.
- The next section examines the classical use of $v$ to obtain a good guess for the period.

# Classical Extraction of the Period

- We sketch a purely classical algorithm for extracting the period from the measured value $v$ obtained from the quantum core of Shor's algorithm.
- When the period $r$ happens to be a power of 2, the quantum Fourier transform gives exact multiples of $\frac{2^n}{r}$.
- This makes the period easy to extract.
- In this case, the measured value $v$ is equal to $j\frac{2^n}{r}$ for some $j$.
- Most of the time $j$ and $r$ will be relatively prime.
- We reduce the fraction $\frac{v}{2^n}$ to its lowest terms.
- This yields a fraction $\frac{j}{r}$ whose denominator is the period $r$.
- The rest of this section explains how to obtain a good guess for $r$ when it is not a power of 2.

# Classical Extraction of the Period (Cont'd)

- In general the quantum Fourier transform gives only approximate multiples of the scaled frequency.
- This complicates the extraction of the period from the measurement.
- When the period is not a power of 2, a good guess for the period can be obtained from the continued fraction expansion of $\frac{v}{2^n}$.
- Shor shows that, with high probability, $v$ is within $\frac{1}{2}$ of some multiple of $\frac{2^n}{r}$, say $j\frac{2^n}{r}$.
- Recall that $n$ was chosen to satisfy $M^2 \leq 2^n < 2M^2$.
- Consider the high-probability case in which $|v - j\frac{2^n}{r}| < \frac{1}{2}$, for some $j$.
- The left inequality $M^2 \leq 2^n$ implies that

$$\left|\frac{v}{2^n} - \frac{j}{r}\right| < \frac{1}{2 \cdot 2^n} \leq \frac{1}{2M^2}.$$

# Classical Extraction of the Period (Cont'd)

- In general, the difference between two distinct fractions $\frac{p}{q}$ and $\frac{p'}{q'}$ with denominators less than $M$ is bounded,

$$\left|\frac{p}{q} - \frac{p'}{q'}\right| = \left|\frac{pq' - p'q}{qq'}\right| > \frac{1}{M^2}.$$

- Thus, there is at most one fraction $\frac{p}{q}$ with denominator $q < M$ such that $|\frac{v}{2^n} - \frac{p}{q}| < \frac{1}{M^2}$.
- So, when $v$ is within $\frac{1}{2}$ of $j\frac{2^n}{r}$, this fraction will be $\frac{j}{r}$.
- The fraction $\frac{p}{q}$ can be computed using a continued fraction expansion.
- We take the denominator $q$ of the obtained fraction as our guess for the period.
- This guess will be correct whenever $j$ and $r$ are relatively prime.

# Review of the Continued Fraction Expansion

- The unique fraction with denominator less than $M$ that is within $\frac{1}{M^2}$ of $\frac{v}{2^n}$ can be obtained efficiently from the continued fraction expansion of $\frac{v}{2^n}$.
- Let $[x]$ be the greatest integer less than $x$.
- Define the following sequences:

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0;$$
$$a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i.$$

- Moreover,

$$p_0 = a_0, \quad p_1 = a_1 a_0 + 1, \quad p_i = a_i p_{i-1} + p_{i-2};$$
$$q_0 = 1, \quad q_1 = a_1, \quad q_i = a_i q_{i-1} + q_{i-2}.$$

- The recurrences compute the first fraction $\frac{p_i}{q_i}$, with $q_i < M \leq q_{i+1}$.

# Example

- Assume $M = 21$, $n = 9$, $v = 427$.

  We work with

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0;$$
$$a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i;$$
$$p_0 = a_0, \quad p_1 = a_1 a_0 + 1, \quad p_i = a_i p_{i-1} + p_{i-2};$$
$$q_0 = 1, \quad q_1 = a_1, \quad q_i = a_i q_{i-1} + q_{i-2}.$$

  We obtain:

| $i$ | $a_i$ | $p_i$ | $q_i$ | $\epsilon_i$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | $\frac{427}{512}$ |
| 1 | 1 | 1 | 1 | $\frac{85}{427}$ |
| 2 | 5 | 5 | 6 | $\frac{2}{85}$ |
| 3 | 42 | 211 | 253 | $\frac{1}{2}$ |

Subsection 3

# Example Illustrating Shor's Algorithm

# The Problem

- We illustrate the operation of Shor's algorithm as it attempts to factor the integer $M = 21$.
- We compute
$$M^2 = 441 \leq 2^9 < 882 = 2M^2.$$
- So we take $n = 9$.
- As $\lceil \log M \rceil = m = 5$, the second register requires five qubits.
- Thus, the state
$$\frac{1}{\sqrt{2^9}} \sum_{x=0}^{2^9-1} |x\rangle |f(x)\rangle$$
is a 14-qubit state, with:
  - Nine qubits in the first register;
  - Five qubits in the second register.

# Measurement

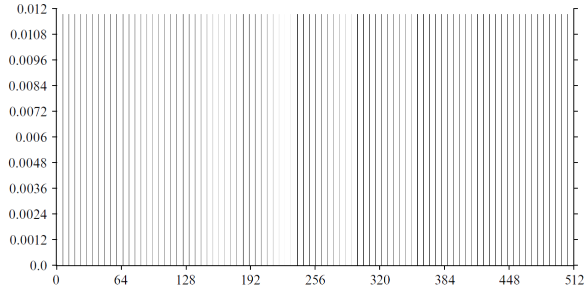
- Suppose the randomly chosen integer is $a = 11 < 21 = M$.
- $a$ and $M$ are relatively prime.
- We measure the second register of the superposition of equation

$$\frac{1}{\sqrt{2^9}} \sum_{x=0}^{2^9-1} |x\rangle|f(x)\rangle.$$

- Suppose the measurement produces $u = 8$.
- The state of the first register after this measurement is shown in the figure on the next slide.

# State of First Register After Measurement

- The state of the first register after the measurement $u = 8$.

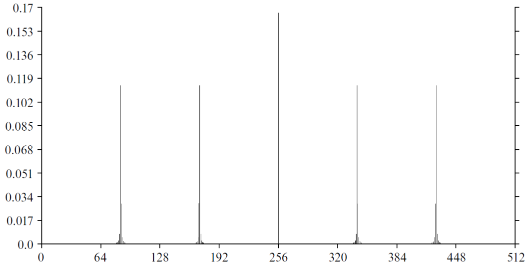



Probabilities for measuring $x$ when measuring $C\sum_{x\in X}|x,8\rangle$, where $X = \{x|211^x \mod 21 = 8\}$.

- The probabilities for measuring $x$ when measuring the state is $C\sum_{x\in X}|x,8\rangle$, where $X = \{x|11^x \mod 21 = 8\}$.
- The figure shows the periodicity of $f$.

# Fourier Transform

- The next figure shows the result of applying the quantum Fourier transform to this state.




Probability distribution of the quantum state after Fourier transformation.

- It is the graph of the FFT of the function of the preceding figure.
- In this particular example, the period of $f$ does not divide $2^n$.
- So the probability distribution has some spread around multiples of $\frac{2^n}{r}$ instead of having a single spike at each of these values.

# The Output

- Suppose that measurement of the state returns $v = 427$.
- Since $v$ and $2^n$ are relative prime, we use the continued fraction expansion to obtain a guess $q$ for the period.
- The following table shows a trace of the continued fraction algorithm:

| $i$ | $a_i$ | $p_i$ | $q_i$ | $\epsilon_i$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0.8339844 |
| 1 | 1 | 1 | 1 | 0.1990632 |
| 2 | 5 | 5 | 6 | 0.02352941 |
| 3 | 42 | 211 | 253 | 0.5 |

- The algorithm terminates with $6 = q_2 < M \leq q_3$.

# The Period and the Factors

- We came up with $q = 6$ as our guess for the period of $f$.
- Now 6 is even.
- So the following are likely to have a common factor with $M$:

$$\begin{aligned} a^{6/2} - 1 &= 11^3 - 1 = 1330; \\ a^{6/2} + 1 &= 11^3 + 1 = 1332. \end{aligned}$$

- In this particular example,

$$\gcd(21, 1330) = 7 \quad \text{and} \quad \gcd(21, 1332) = 3.$$

Subsection 4

# The Efficiency of Shor's Algorithm

# Number of Steps

- We consider the efficiency of Shor's algorithm, examining:
  - The efficiency of each part in terms of the number of gates or classical steps needed to implement the part;
  - The expected number of times the algorithm would need to be repeated.
- The Euclidean algorithm on integers $x > y$ needs at most $O(\log x)$ steps.

  So both Parts 1 and 5 require $O(\log M) = O(m)$ steps.
- The continued fraction algorithm used in Part 4 is related to the Euclidean algorithm and also requires $O(m)$ steps.
- Part 3 is a measurement of $m$ qubits.

  In addition, as we will see, it can be omitted altogether.

# Number of Steps (Cont'd)

- Part 2 consists of the computation of $U_f$ and the computation of the quantum Fourier transform.
- We showed that the quantum Fourier transform on $m$ qubits requires $O(m)$ steps.
- The algorithm for modular exponentiation requires $O(n^3)$ steps could be used to implement $U_f$.
- The transformation $U_f$ can be implemented more efficiently using an algorithm for modular exponentiation, described by Shor, that is based on the most efficient classical method known, and runs in $O(n^2 \log n \log \log n)$ time and $O(n \log n \log \log n)$ space.
- These results show that the overall runtime of a single iteration of Shor's algorithm is dominated by the computation of $U_f$.
- Moreover, the overall time complexity for a single iteration of the algorithm is $O(n^2 \log n \log \log n)$.

# Number of Repetitions

- To show that Shor's algorithm is efficient, we also need to show that the parts do not need to be repeated too many times.
- Four things can go wrong:
  - The period of $f(x) = a^x \mod M$ could be odd.
  - Part 4 could yield $M$ as $M$'s factor.
  - The value $v$ obtained in Part 3 might not be close enough to a multiple of $\frac{2^n}{r}$.
  - A multiple $j\frac{2^n}{r}$ of $\frac{2^n}{r}$ is obtained from $v$, but $j$ and $r$ could have a common factor, in which case the denominator $q$ is actually a factor of the period, not the period itself.

# Number of Repetitions (Cont'd)

- The first two problems appear in the classical reduction.
- Standard classical arguments bound the probabilities as at most $\frac{1}{2}$.
- For the case in which the period $r$ divides $2^n$, problem 3 does not arise.
- Shor shows that, in the general case, $v$ is within $\frac{1}{2}$ of a multiple of $\frac{2^n}{r}$ with high probability.
- As for Problem 4, when $r$ divides $2^n$, it is not hard to see that every outcome $v = j\frac{2^n}{r}$ is equally likely.

  After the quantum Fourier transform the state is $C' \sum_{c=0}^{2^n-1} G(c)|c\rangle$, where

$$G(c) = \sum_{x \in X_u} \exp\left(2\pi \boldsymbol{i} \frac{cx}{2^n}\right) = \sum_{y=0}^{2^n/r} \exp\left(2\pi \boldsymbol{i} \frac{cry}{2^n}\right)$$

  where $X_u = \{x : f(x) = u\}$.

# Number of Repetitions (Cont'd)

- As we saw, the final sum is 1 when $c$ is a multiple of $\frac{2^n}{r}$, and 0 otherwise.
- Thus, in this case, any $j \in \{0, \dots, r-1\}$ is equally likely.
- From $j$, we obtain the period $r$ exactly when $r$ and $j$ are relatively prime, $\gcd(r, j) = 1$.
- The number of positive integers less than $r$ that are relatively prime to $r$ is given by the famous Euler $\phi$ function, which is known to satisfy $\phi(r) \geq \frac{\delta}{\log \log r}$ for some constant $\delta$.
- Thus we need to repeat the parts only $O(\log \log r)$ times in order to achieve a high probability of success.
- The argument for the general case, in which $r$ does not divide $2^n$, is more involved but yields the same result.

Subsection 5

## Omitting the Internal Measurement

# Intuition

- In Part 3 of Shor's algorithm, to obtain $u$ one measures the second register of the state in

$$\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle|f(x)\rangle.$$

- This step can be skipped entirely.
- We describe the intuition for why this measurement can be omitted.
- Suppose the measurement is omitted.
- Then the state consists of a superposition of several periodic functions.
- Each function corresponds to a value of $f(x)$.
- All of these functions have the same period.

# Intuition (Cont'd)

- Quantum transformations are linear.
- So applying the quantum Fourier transformation leads to a superposition of the Fourier transforms of these functions.
- Each of the functions corresponds to a different value $u$ of the second register.
- So the different functions remain distinct parts of the superposition and do not interfere with each other.
- Measuring the first register gives a value from one of these Fourier transforms.
- As before, this will be close to $j\frac{2^n}{r}$ for some $j$.
- So it can be used to obtain the period in the same way as before.

# Formalism

- Let $X_u = \{x : f(x) = u\}$.
- Let $R$ be the range of $f(x)$.
- Finally, let $g_u$ be the family of functions indexed by $u$, such that

$$g_u(x) = \begin{cases} 1, & \text{if } f(x) = u \\ 0, & \text{otherwise.} \end{cases}$$

- Using this notation, the state can be written as

$$\begin{aligned} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|f(x)\rangle &= \frac{1}{\sqrt{2^n}} \sum_{u \in R} \sum_{x \in X_u} |x\rangle|u\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{u \in R} \left( \sum_{x=0}^{2^n-1} g_u(x)|x\rangle \right) |u\rangle. \end{aligned}$$

- The amplitudes in states with different $u$ in the second register can never interfere (add or cancel) with each other.

# Formalism (Cont'd)

- The result of applying the transform $U_F \otimes I$ to the preceding state can be written

$$
\begin{aligned}
U_F \otimes I \left( \frac{1}{\sqrt{2^n}} \sum_{u \in R} \left( \sum_{x=0}^{2^n-1} g_u(x) |x\rangle \right) |u\rangle \right) &= \frac{1}{\sqrt{2^n}} \sum_{u \in R} (U_F \sum_x g_u(x)|x\rangle)|u\rangle \\
&= C' \sum_{u \in R} (\sum_{c=0}^{2^n-1} G_u(c)|c\rangle)|u\rangle,
\end{aligned}
$$

  where $G_u(c)$ is the discrete Fourier transform of $g_u(x)$.
- This results is a superposition of the possible states of equation $U_F(C \sum_x g(x)|x\rangle) = C' \sum_c G(c)|c\rangle$ over all possible $u$.
- Now the $g_u$ all have the same period.
- So measuring the first part of this state returns a $c$ close to a multiple of $\frac{2^n}{r}$.
- This has the same effect as when the second register was measured.

Subsection 6

# Generalizations

# The Discrete Logarithm

- Let $\mathbb{Z}_p^*$ be the group of integers $\{1, \ldots, p-1\}$ under multiplication modulo $p$.
- Let $b$ be a generator for this group (any $b$ relatively prime to $p-1$ will do).
- The **discrete logarithm** of $y \in \mathbb{Z}_p^*$ with respect to base $b$ is the element $x \in \mathbb{Z}_p^*$, such that

$$b^x = y \mod p.$$

# The Discrete Logarithm Problem

**Discrete Logarithm Problem**: Given a prime $p$, a base $b \in \mathbb{Z}_p^*$, and an arbitrary element $y \in \mathbb{Z}_p^*$, find an $x \in \mathbb{Z}_p^*$, such that

$$b^x = y \mod p.$$

- For large $p$, this problem is computationally difficult to solve.
- The Discrete Logarithm Problem can be generalized to arbitrary finite cyclic groups $G$.
- However, for some large $G$, it is is not difficult to solve classically.
- The Discrete Logarithm Problem is a special case of the Abelian Hidden Subgroup Problem.

# Hidden Subgroup Problems

**The Hidden Subgroup Problem**: Let $G$ be a group.

Suppose a subgroup $H < G$ is implicitly defined by a function $f$ on $G$ in that $f$ is constant and distinct on every coset of $H$.

Find a set of generators for $H$.

- The aim is to find a polylogarithmic algorithm that computes a set of generators for $H$ in $O((\log|G|)^k)$ steps, for some $k$.
- The difficulty of the problem depends not only on $G$ and $f$ but also on what is meant by given a group $G$.

# Hidden Subgroup Problems (Cont'd)

- Some useful properties may be expensive to compute from certain descriptions of a group and immediate from others.
- For example, computing the size of a group from certain types of descriptions, such as a defining set of generators and relations, is known to be computationally hard.
- Also, we can hope to find a solution in poly-log time only if $f$ itself is computable in poly-log time.
- The general hidden subgroup problem remains unsolved.
- However, a polylogarithmic bounded probability quantum algorithm for the general case of *finite Abelian groups, specified in terms of their cyclic decomposition*, exists.

# Finite Abelian Hidden Subgroup Problem

**Finite Abelian Hidden Subgroup Problem**: Let $G$ be a finite Abelian group, with cyclic decomposition $G = \mathbb{Z}_{n_0} \times \cdots \times \mathbb{Z}_{n_L}$.

Suppose $G$ contains a subgroup $H < G$ that is implicitly defined by a function $f$ on $G$ in that $f$ is constant and distinct on every coset of $H$.

Find a set of generators for $H$.

Example (**Period-finding as a Hidden Subgroup Problem**):

Period-finding can be rephrased as a hidden subgroup problem.

Let $f$ be a periodic function on $\mathbb{Z}_N$, with period $r$ that divides $N$.

The subgroup $H < \mathbb{Z}_N$ generated by $r$ is the hidden subgroup.

Suppose a generator $h$ for $H$ has been found.

Then the period $r$ can be found by taking the greatest common divisor of $h$ and $N$,

$$r = \gcd(h, N).$$

# The Discrete Logarithm as a Hidden Subgroup Problem

- In addition to Period-finding, both Simon's Problem and the Discrete Logarithm Problem are instances of the finite Abelian Hidden Subgroup Problem.

  Example (**The Discrete Logarithm as a Hidden Subgroup Problem**):

  Recall the Discrete Logarithm Problem.

  Given the group $G = \mathbb{Z}_p^*$, where $p$ is prime, a base $b \in G$, and an arbitrary element $y \in G$, find an $x \in G$ such that

  $$b^x = y \mod p.$$

  Consider $f : G \times G \to G$, where

  $$f(g,h) = b^{-g} y^h.$$

# The Discrete Logarithm as a Hidden Subgroup (Cont'd)

- Let $f : G \times G \to G$, where

$$f(g,h) = b^{-g}y^{h}.$$

Let the hidden subgroup $H$ of $G \times G$ be the set of elements satisfying

$$f(g,h) = 1.$$

It consists of tuples of the form $(mx, m)$.

From any generator of $H$, the element $(x, 1)$ can be computed.

Thus, solving this Hidden Subgroup Problem yields $x$.

So we obtain a solution to the Discrete Logarithm Problem.