- Discrete Log Problem
- Shor's Quantum Algorithm
  1. Simple case
  2. Hard case

1 Discrete Log Problem

We will now consider the problem of computing the discrete log, which can be posed as follows:

GIVEN: A prime p, a generator $g \in \mathbb{Z}_p^*$, and any $x \in \mathbb{Z}_p^*$

FIND: $r \bmod p-1$ such that $x \equiv g^r (\bmod p)$

Like the factoring problem which we considered in the previous lecture, the most famous application of the discrete log problem is in cryptography, namely, in the implementation of the ElGamal encryption scheme. In a practical system, we construct the function $f(r) = g^r \bmod p$ by choosing a random prime p such that $p-1$ has a known prime factorization:

$$p - 1 \equiv \prod_i {q_i}^{e_i} \tag{1}$$

Then, it is easy to find a generator by choosing a random element g and checking that

$$g^{\frac{p-1}{q_i}} \not\equiv 1 \pmod p \tag{2}$$

Here, we are using the heuristic that there are many generators in $\mathbb{Z}_p^*$ so that we will choose a generator therein with high probability.

Our function $f(r)$ is known as a *one-way function* because it is believed that its inverse (i.e., the discrete log) is hard to compute.

2 Shor's Quantum Algorithm

Before diving into the details of Shor's algorithm, we'll begin with a few comments. First, a preliminary observation:

Fact 1 *Suppose that we know the order k of the generator $g \in \mathbb{Z}_p^*$. Then, if $g^r \equiv x \pmod p$, the following equation*

$$g^a x^{-b} \equiv (\text{constant}) \pmod p \tag{3}$$

is satisfied for all integers $b \geq 0$ such that $a = k + br$.

We now describe Shor's quantum algorithm for computing the discrete log. We will be making extensive use of the Fourier transform, as in previous algorithms. Thus far, we have implicitly assumed that we can perform the Fourier transform in polynomial time. In his original paper [1], Shor addresses this assumption by showing that a number a such that $0 \le a < q$ can be transformed into a number b in $0 \le b < q$ with amplitude $1/\sqrt{q}\exp(2\pi iab/q)$ in polynomial time provided that: (1) q can be represented with a polynomial number of bits, and (2) that q must be *smooth*, i.e., must have "small" prime factors.

2.1 The easy case

Beginning in the state $|0\rangle|0\rangle$, we perform a Fourier transform $\bmod (p-1)$ to place us in the uniform superposition state:

$$
\begin{aligned}
|0\rangle|0\rangle \quad \stackrel{\text{FT mod }(p-1)}{\longrightarrow} \quad & \frac{1}{\sqrt{p-1}}\sum_{a=0}^{p-2}|a\rangle \cdot \frac{1}{\sqrt{p-1}}\sum_{b=0}^{p-2}|b\rangle \\
&= \frac{1}{p-1}\sum_{a,b}|a\rangle|b\rangle
\end{aligned}
\tag{4}
$$

over all a and b (these correspond to those shown in the preliminary fact). In effect, we are in the state

$$
\frac{1}{p-1}\sum_{a,b}|a\rangle|b\rangle\underbrace{|g^a x^{-b} \bmod p\rangle}_{\text{measure}} \tag{5}
$$

We now perform the usual notional measurement on the quantity shown. Since there is a value a corresponding to each value of b in accordance with the preliminary fact, we are reduced to the superposition state

$$
\frac{1}{\sqrt{p-1}}\sum_{b=0}^{p-2}|br + k \bmod (p-1)\rangle|b\rangle \tag{6}
$$

Note that the term $|br + k \bmod (p-1)\rangle|b\rangle$ is a coset of the subgroup generated by $|br \bmod (p-1)\rangle|b\rangle$; thus, by group invariant arguments we are free to consider our subsequent operations on this subgroup instead:

$$
\frac{1}{\sqrt{p-1}}\sum_{b=0}^{p-2}|br \bmod (p-1)\rangle|b\rangle \tag{7}
$$

Our next step is to carry out a Fourier transform over $\mathbb{Z}_{p-1}^2$ on equ. (7), i.e., we will find $\alpha_{c,d}$ such that

$$
\frac{1}{\sqrt{p-1}}\sum_{b=0}^{p-2}|br \bmod (p-1)\rangle|b\rangle \quad \stackrel{\text{FT mod }(p-1)}{\longrightarrow} \quad \sum_{c,d}\alpha_{c,d}|c,d\rangle \tag{8}
$$

To do so, we will need to assume that $p-1$ is smooth. This is the main assumption of "the easy case."[1] Then, let ω be the $p-1$ root of unity, i.e., $\omega = e^{2\pi i/(p-1)}$. Explicitly, the calculation is

$$
\begin{aligned}
\xrightarrow{\text{FT mod } (p-1)} \quad & \frac{1}{\sqrt{p-1}} \sum_{b=0}^{p-2} \frac{1}{\sqrt{p-1}} \sum_{c=0}^{p-2} \omega^{brc} |c\rangle \frac{1}{\sqrt{p-1}} \sum_{d=0}^{p-2} \omega^{bd} |d\rangle \\
= \quad & \frac{1}{(p-1)^{3/2}} \sum_{c,d} \underbrace{\sum_{b=0}^{p-2} \left(\omega^{rc+d}\right)^{b} |c\rangle |d\rangle}_{\alpha_{c,d}}
\end{aligned}
\tag{9}
$$

Notice that if $rc + d \not\equiv 0 \pmod{p-1}$, then $\alpha_{c,d} = 0$, since we would essentially be summing all of the $(p-1)^{\text{st}}$ roots of unity. Thus, our probability will be non-zero if and only if $rc + d \equiv 0 \pmod{p-1}$, i.e.,

$$
r \equiv -\frac{d}{c} \pmod{p-1} \tag{10}
$$

This shows us that if c and $p-1$ are relatively prime, then we can find r by division—and the probability that c and $p-1$ are relatively prime will be non-negligible since we will be choosing from among all c's with equal probability (i.e., since we are in a uniform superposition over $|c, d\rangle$).

2.2 The hard case

Recall that what made the easy case "easy" was the assumption that $p-1$ was smooth. For the more general case, we remove this assumption by choosing a smooth q such that $p \leq q \leq 2p$, and perform our transformation of equ. (7) in $\mathbb{Z}_q$.[2]

Our algorithm in the hard case is identical to the easy case up to equ. (7). Now, instead of transforming over $\mathbb{Z}_{p-1}^2$, we transform with respect to $\mathbb{Z}_q^2$:

$$
\frac{1}{\sqrt{p-1}} \sum_{b=0}^{p-2} |br \bmod (p-1)\rangle |b\rangle \quad \xrightarrow{\text{FT mod } q}
$$

$$
\frac{1}{p-1} \frac{1}{q} \sum_{c=0}^{q} \sum_{d=0}^{q} \alpha_{c,d} |c, d\rangle \tag{11}
$$

where

$$
\alpha_{c,d} = \sum_{b=0}^{p-2} \omega_q^{br \bmod (p-1) c + bd} \tag{12}
$$

[1] As Shor [1] notes, there already exist classical polynomial time algorithms for the smooth case.

[2] However, we claim in our summation over b in equ. (7), the upper limit remains $p-2$, instead of $q-1$. This is because we can "throw out" values greater than $p-2$ as not useful. In practice, we might do this by adding a bit which we can then measure and test for such unwanted values, without disrupting the state of b.

Note that we are now working with the q^{th} root of unity $\omega_q = \exp(2\pi i/q)$; we will henceforth supress the subscript q.

Recall the definition of mod:

$$br \bmod (p-1) = br - \left\lfloor \frac{br}{p-1} \right\rfloor (p-1) \tag{13}$$

Then, using a little algebraic manipulation, rewrite

$$\begin{aligned} \omega^{br \bmod (p-1)} &= \omega^{brc - \frac{br}{p-1}[c(p-1) \bmod q]} && (14) \\ & \times \omega^{\left[\frac{br}{p-1} - \left\lfloor \frac{br}{p-1} \right\rfloor (p-1)\right][c(p-1) \bmod q]} && (15) \end{aligned}$$

For now, let's conceal the second factor as an "error factor"; so,

$$\begin{aligned} \alpha_{c,d} &= \sum_{b=0}^{p-2} \left[\omega^{rc+d-\frac{r}{p-1}(c(p-1) \bmod q)}\right]^b \\ & \times (\text{errorfactor}) && (16) \end{aligned}$$

Now we want to show that certan values of c and d occur with large probability. First, think of choosing c such that

$$c(p-1) \bmod q \leq \frac{q}{20} \tag{17}$$

The factor of 1/20, though somewhat arbitrarily, is selected so that the error factor is near 1.[3] Next, note that the values of c, d which occur with relatively high probability satisify the bound

$$\left| rc + d - \frac{r}{p-1}\left[c(p-1) \bmod q\right] \right| \leq \frac{1}{2} \tag{18}$$

In particular, each such pair of c, d satisfying equs. (17–18) occur with an approximate probability amplitude of

$$\begin{aligned} \text{amplitude} &\approx \frac{1}{\sqrt{p-1}} \frac{1}{q} \times (p-1) \times (\text{constant}) \\ &\approx \frac{1}{\sqrt{q}} \times (\text{constant}) && (19) \end{aligned}$$

where we have used the fact that $q \approx p$. Therefore, the probability of such a pair is roughly $\frac{1}{p} \times (\text{constant})$.

Next, we must show that if we find such a pair c, d that we can then find r. Begin by rewriting condition (18) as

[3] Shor [1] notes that the selection of this factor implies that the error factor will not deviate from 1 by more than the small factor, $\exp(\pi i/10)$.

$$\left| d + \frac{r[c(p-1) - c(p-1) \bmod q]}{p-1} \right| \leq \frac{1}{2} \tag{20}$$

Notice that r is the only unknown. Also observe that $q|[c(p-1) - c(p-1) \bmod q]$. Then, dividing both sides by q,

$$\left| \frac{d}{q} - \frac{r \times l}{p-1} \right| \leq \frac{1}{2q} \tag{21}$$

This is our old friend, the continued fraction error bound. To find r, it is easiest to round $\frac{d}{q}$ to the closest multiple of $p-1$ (call it $\frac{e}{p-1}$), and then compute r from

$$\frac{e}{p-1} = \frac{rl}{p-1} \Longrightarrow r = \frac{e}{l} \tag{22}$$

3 Summary

We have explored in detail the methodology of Shor's quantum discrete log algorithm. In particular, we showed that in the general case, we could begin in a random superposition state of two integers and then perform a series of Fourier transformations to set-up a new superposition state. In this new state, we were able to obtain, with relatively high probability, two new integers c, d satisfying

$$\left| rc + d - \frac{r}{p-1} \left[c(p-1) \bmod q \right] \right| \leq \frac{1}{2} \tag{23}$$

Any c, d that satisfies this bound will occur with relatively high, constant probability. From this, we were easily able to calculate the value r.

References

[1] P. Shor. "Algorithms for quantum computation: Discrete log and factoring." [This paper is available from the *Quantum Computation Archive* web site, `http://feynman.stanford.edu/qcomp/`.]