Lecture 10

In which we present a polynomial time quantum algorithm for the discrete logarithm problem.

1 The Discrete Log Problem

If p is a prime and g is a generator of the multiplicative group $\mathbb{Z}_p^*$, then the modular exponentiation function

$$x \to g^x \bmod p$$

is a bijection of $\mathbb{Z}_p^*$ to $\mathbb{Z}_p^*$. The discrete log problem is the problem of inverting this mapping, that is, given a prime p, a generator g of $\mathbb{Z}_p^*$ and an element $z \in \mathbb{Z}_p^*$, find the unique r, $0 \leq r \leq p-2$, such that $g^r \equiv z \pmod{p}$.

An efficient algorithm for the discrete log problem can be used to break several public-key cryptosystems whose design is based on having $(p, g, g^{x \cdot y} \bmod p)$ as a private key, where p is a properly chosen prime, g is a generator of $\mathbb{Z}_p^*$, and x, y are randomly chosen, while the public key is $(p, g, g^x \bmod p, g^y \bmod p)$.

The discrete log problem can be formulated for every group G. Once the group is fixed, or a description is given, an input to the problem are two elements $a, z \in G$, and the goal is to find an integer r such that $a^r = z$, where a^r means $a \times a \times \cdots a$ r times and $\times$ is the group operation. An algorithm for this more general problem breaks public-key cryptosystems based on elliptic curves.

In this lecture we describe a polynomial time quantum algorithm for the discrete logarithm problem in $\mathbb{Z}^*p$, but the algorithm can be adapted to work in any Abelian group. (The groups arising in elliptic curves cryptographic constructions are Abelian.)

2 A Fourier Transform for Bivariate Functions

We briefly describe a theory of discrete Fourier transforms for functions with two inputs. If M is a positive integer, the functions

$$f : \{0, \dots, M-1\} \times \{0, \dots, M-1\} \to \mathbb{C}$$

form an M^2-dimensional vector space. In the univariate case of functions $f : \{0, \dots, M-1\} \to \mathbb{C}$, we derived the Fourier transform by defining an orthonormal basis and writing f as a linear combination of basis functions. Similarly, we will now define M^2 orthonormal functions and write each bivariate function as a linear combination of basis functions.

A general principle is that if $v_1, \dots, v_k$ are an orthonormal vectors, then the collection of all the tensor products $v_i \otimes v_j$ is a set of k^2 orthonormal vectors. Consider now the functions $\chi_s(x) = \frac{1}{\sqrt{M}} e^{-2\pi i \frac{1}{M} \cdot s \cdot x}$; we prove that they are orthonormal, and so the collection of their tensor products

$$\chi_{s_1,s_2}(x,y) := \frac{1}{M} e^{-2\pi i \frac{1}{M} \cdot (s_1 x + s_2 y)}$$

is a collection of M^2 orthornomal functions, and thus it is an orthonormal basis for the set of functions

$$f : \{0, \dots, M-1\} \times \{0, \dots, M-1\} \to \mathbb{C}$$

Each such function can be written as a linear combination

$$f(x,y) = \sum_{s_1,s_2} \hat{f}(s_1,s_2) \cdot \chi_{s_1,s_2}(x,y)$$

where the coefficients of the linear combination (the Fourier coefficients of f) can be computed as

$$\hat{f}(s_1,s_2) = \langle f, \chi_{s_1,s_2} \rangle = \frac{1}{M} \sum_{x,y} f(x,y) e^{2\pi i \frac{1}{M} \cdot (s_1 x + s_2 y)}$$

The transformation from the values $f(x,y)$ to the coefficients $\hat{f}(s_1,s_2)$ is a change of orthonormal basis, and so it is unitary linear transformation, and so if

$$\sum_{x,y} f(x,y) |x,y\rangle$$

is a quantum state, then

$$\sum_{s_1,s_2} \hat{f}(s_1,s_2) |s_1,s_2\rangle$$

is also a quantum state, and the transformation from the former to the latter is the quantum (bivariate) Fourier transform; it is easy to see that this can be done in quantum polynomial time if $M = 2^m$ is a power of 2 by first applying the standard quantum Fourier transform to x and then to y.

3 A Generalized Period Finding Algorithm

Given a prime p, a generator g of $\mathbb{Z}_p^*$ and an element $a = g^r \bmod p$ of $\mathbb{Z}_p^*$, define the function

$$F(x, y) := a^x \cdot (g^{-1})^y \bmod p = g^{xr-y} \bmod p$$

Then $F(\cdot, \cdot)$ has a period in the sense that for every (x, y) we have

$$F(x, y) = F(x - 1, y + r)$$

We will perform a bivariate version of the period-finding algorithm that we used to solve the factorization problem, and then we will see that, after a constant number of executions of the algorithm, we can recover r.

- Input: prime p, generator g of $\mathbb{Z}_p^*$, element $z \in \mathbb{Z}_p^*$
- Step 1: Fix an $M = 2^m$ such that $p \leq M \leq 2p - 1$ and construct the state

  $$\frac{1}{M} \sum_{x,y} |x\rangle|y\rangle|a^x \cdot (g^{-1})^y \bmod p\rangle \tag{1}$$

  where each of the three parts of the state is an integer in $\{0, \ldots, M - 1\}$, represented as an m-qubit string.

  The function $x, y \rightarrow a^x \cdot g^{-y} \bmod p$ is computable in polynomial time, and so there is a quantum circuit C_{modexp} of polynomial size that, given $|x\rangle|y\rangle|0\rangle$, outputs $|x\rangle|y\rangle|a^x g^y \bmod p\rangle$. Starting from the state $|0\rangle|0\rangle|0\rangle$, we first apply Hadamard gates to the first $2m$ bits, which results in the state

  $$\frac{1}{M} \sum_{x,y} |x\rangle|y\rangle|0\rangle$$

  and then we apply C_{modexp} to the above state.

- Step 2: Measure the third integer in the quantum state.

  If the outcome of the measurement is $g^k \bmod p$, then the residual state is

  $$\frac{1}{\sqrt{S_k}} \sum_{x,y \in S_k} |x\rangle|y\rangle|g^k \bmod p\rangle$$

  Where

  $$S_k := \{x, y : 0 \leq x < M \wedge 0 \leq y < M \wedge rx - y \equiv k \pmod{p - 1}\}$$

  From now we will disregard the third integer of the state, which has been fixed by the measurement.

- Step 3: Apply the bivariate quantum Fourier transform.

  This gives the state

$$\frac{1}{M}\frac{1}{\sqrt{S_k}}\sum_{s_1=0}^{M-1}\sum_{s_2=0}^{M-1}\sum_{x,y\in S_k}\omega^{s_1x+s_2y}|s_1\rangle|s_2\rangle$$

  where $\omega = e^{2\pi i\frac{1}{M}\cdot(s_1x+s_2y)}$

- Step 4: Measure the state

It remains to show that after seeing a constant number of executions of the algorithm we can reconstruct r from the outcomes at Step 4.

4 Analysis of the Last Step

To understand the distribution of outcomes that we get at Step 4, let us first do a non-rigorous calculation assuming that $M = p-1$ and that $p-1$ is prime. In such a case, S_k is just the set of $p-1$ pairs $(x,y)\in\mathbb{Z}_{p-1}\times\mathbb{Z}_{p-1}$ such that $y = rx-k \bmod p-1$. The amplitude of state $|s_1,s_2\rangle$ at Step 4 is

$$\frac{1}{(p-1)^{1.5}}\sum_x \omega^{s_1x+s_2\cdot(rx-k \bmod p-1)} = \frac{1}{(p-1)^{1.5}}\sum_x \omega^{s_1x+s_2rx-s_2k}$$

because operations in the exponent of ω are performed mod M, which is the same as mod $p-1$.

The probability of the outcome $|s_1,s_2\rangle$ is

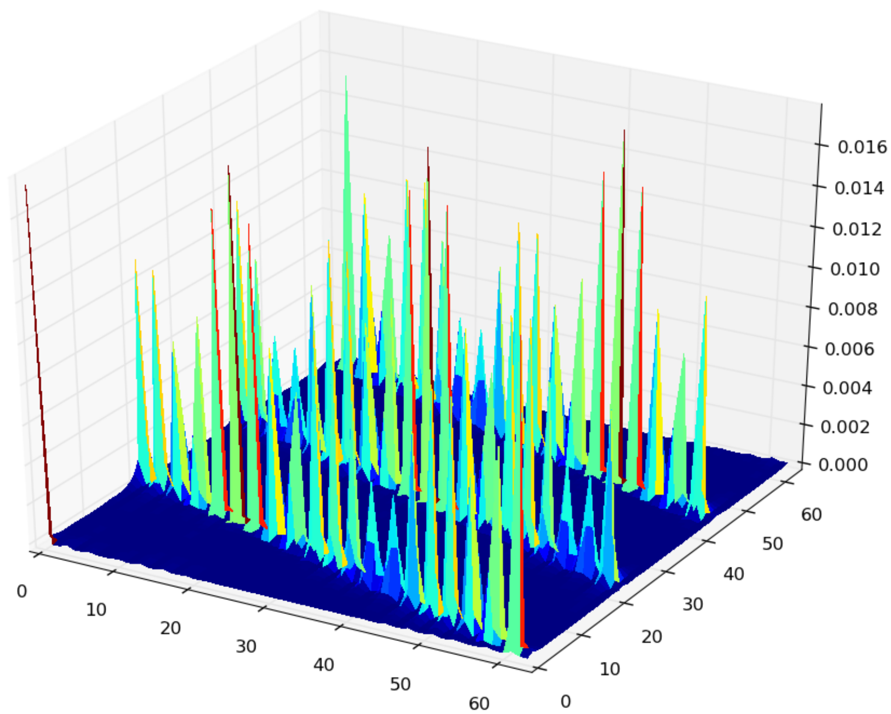
$$\frac{1}{(p-1)^3}\left|\sum_x \omega^{s_1x+s_2rx-s_2k}\right|^2 = \frac{1}{(p-1)^3}\left|\omega^{-s_2k}\right|^2\left|\sum_x \omega^{s_1x+s_2rx}\right|^2 = \frac{1}{(p-1)^3}\left|\sum_x (\omega^{s_1+s_2r})^x\right|^2$$

and if $s_1+s_2r\equiv 0 \bmod p-1$ then the probability is $1/(p-1)$, and otherwise it is zero. This means that from an outcome (s_1,s_2) of Step 4 we can reconstruct r as

$$r = -s_1\cdot(s_2)^{-1} \bmod (p-1)$$

Unfortunately, we do not have $M = p-1$, and so the modular identities that we get in the exponent of ω are not the same as in the exponent of g, complicating the structure of S_k and adding error terms to the above calculations. We also don't have that $p-1$ is prime, which means that at the end we may have problems inverting modulo $p-1$.

Before discussing how to deal with these problems, let us see what we may expect "in practice." Suppose that we run the algorithm on input $p = 61$, $g = 26$ and $z = 8$. We pick $M = 64$, and the probability of the possible outcomes $|s_1, s_2\rangle$ at Step 4 is plotted below:



and we see that most of the probability is concentrated on outcomes s_1, s_2 such that $s_1 + 3s_2$ is close to a multiple of 64, and indeed 3 is the correct answer.

For every $x \in \{0, \ldots, M-1\}$, the set S_k contains the pairs (x, y) such that $0 \leq y < M$ and $y = rx - k \bmod p - 1$ and there is always either one or two such y. Hence, $M \leq S_k \leq 2M$.

In our analysis it will be convenient to use the following notation: $\{a\}_M$ is the difference between a and the multiple of M that is closest to a. Note that $a \equiv \{a\}_M \pmod{M}$ and that $-M/2 \leq \{a\}_M \leq M/2$.

The probability of an outcome $|s_1, s_2\rangle$ at Step 4 is

$$\frac{1}{M^2} \cdot \frac{1}{S_k} \left| \sum_{x,y \in S_k} \omega^{s_1 x + s_2 y} \right|$$

$$\geq \frac{1}{2M^3} \left| \sum_{x,y \in S_k} \omega^{s_1 x + s_2 y} \right|^2$$

$$= \frac{1}{2M^3} \left| \sum_{x,y \in S_k} \omega^{s_1 x + s_2 (rx - k \bmod p-1)} \right|^2$$

$$= \frac{1}{2M^3} \left| \sum_{x,y \in S_k} \omega^{s_1 x + s_2 rx - s_2 k - (p-1) s_2 \lfloor \frac{rx-k}{p-1} \rfloor} \right|^2$$

$$= \frac{1}{2M^3} \left| \sum_{x,y \in S_k} \omega^{s_1 x + s_2 rx - (p-1) s_2 \lfloor \frac{rx-k}{p-1} \rfloor} \right|^2 \tag{2}$$

where, in the second-to-last equation, we use $a \mod k = a - \lfloor \frac{a}{k} \rfloor$.

Our approach is now to define a notion of "good" pair (s_1, s_2), to show that there are $\Omega(M)$ such pairs, that each of them is generate with probability $\Omega(1/M)$, and that from a good pair it is possible to compute (a large amount of information about) r.

Definition 1 (Good Pairs) *A pair (s_1, s_2) is good if*

1. *$s_1 + s_2 r - \frac{r}{p-1}\{s_2(p-1)\}_M$ differs from a multiple of M by at most $\pm 1/2$.*
2. *$s_2(p-1)$ differs from a multiple of M by at most $\pm M/12$.*

Lemma 2 (Many Good Pairs) *There are at least $M/12$ good pairs.*

Proof: We first prove the following fact.

Claim 3 *Let $k > a > 0$ be positive integers, and $t < k/2$. Then there are at least t distinct values x such that*

$$-t \leq \{ax\}_k \leq t$$

Proof: Consider the mapping

$$x \to ax \bmod k$$

This is a $\gcd(a,k)$-to-1 mapping, that is, there are $k/\gcd(a,k)$ possible outputs, each having $\gcd(a,k)$ preimages, and each possible output is a multiple of $\gcd(a,k)$.

(This is easy to see using the Chinese remainders theorem.) We are interested in the number of preimages of 0 and of possible outputs in the range $1, \ldots, t$ and in the range $M - t, \ldots, M - 1$; overall there are

$$d + 2d \cdot \left\lfloor \frac{t}{d} \right\rfloor$$

such preimages, where $d := \gcd(a, k)$. If $t \leq d$, then we have at least $d \geq t$ preimages; if $t > d$ we have at least

$$d + 2d \left(\frac{t}{d} - 1 \right) = 2t - d > t$$

preimages. □

From the claim above (applied to $x = s_2$, $a = p - 1$ and $k = M$), we see that there are at least $M/12$ choices of s_2 that satisfy property (2) of being a good pair. For each such s_2, we can find an s_1 for which property (1) holds. □

Lemma 4 (High Probability of Good Pairs) *Each good pair has probability at least $\Omega(1/M)$ of being a possible outcome of Step 4.*

Proof: [Sketch] We write

$$\omega^{s_1 x + s_2 r x - (p-1) s_2 \lfloor \frac{rx-k}{p-1} \rfloor} = \omega^{Ax + B(x)}$$

where

$$A := s_1 + s_2 r - \frac{r}{p-1} \{s_2(p-1)\}_M$$

and

$$B(x) := \{s_2(p-1)\}_M \cdot \left(\frac{rx}{p-1} - \left\lfloor \frac{rx-k}{p-1} \right\rfloor \right)$$

When (s_1, s_2) is a good pair, we have $|A| \leq 1/2$ and $|B| \leq M/12$. The summation

$$\sum_{x, y \in S_k} \omega^{Ax + B(x)}$$

is a summation of complex numbers ω^{Ax}, which are either all of the form $e^{i\theta}$ either with θ between 0 and π or between 0 and $-\pi$, and they are uniformly spaced and some of them may be repeated twice in the sum. Each of them is shifted by $e^{B(x)}$, which is of the form $e^{i\theta}$ with $|\theta| \leq \pi/6$. Such a sum produces a vector of length $\Omega(M)$, and so the overall amplitude of (s_1, s_2) is $\Omega(1/M)$. □

Now suppose that we have a good pair (s_1, s_2); we see that

$$-\frac{1}{2M} \leq \frac{s_1}{M} + r \cdot \left(\frac{s_2(p-1) - \{s_2(p-1)\}_M}{M(p-1)} \right) \leq \frac{1}{2M} \bmod 1$$

where $x \bmod 1$ stand for the difference between the real number x and the closest integer to x.

We also note that $\frac{s_2(p-1)-\{s_2(p-1)\}_M}{M}$ is an integer. This means that by finding the multiple $a/(p-1)$ of $1/(p-1)$ closest to s_1/M we find a number of the form $rc/M \bmod 1$, where $c = \frac{s_2(p-1)-\{s_2(p-1)\}_M}{M}$ is a known quantity. So we have found numbers a, c such that $a/(p-1) \equiv rc/(p-1) \bmod 1$, that is,

$$a \equiv rc \bmod (p-1)$$

Now we can find

$$r = a \cdot c^{-1} \bmod p-1$$

provided that $\gcd(c, p-1) = 0$. What do we do if c and $p-1$ have common factors? We can still get some useful information, because it is definitely true that

$$a \equiv rc \bmod \frac{p-1}{\gcd(p-1, c)}$$

and we can invert c modulo $(p-1)/\gcd(p-1, c)$ and we find $r \bmod (p-1)/\gcd(p-1, c)$.

If we run the algorithm twice, we get good pairs both times, and the two good pairs lead us to values c, c' with no common factors, then from $r \bmod (p-1)/\gcd(p-1, c)$ and $r \bmod (p-1)/\gcd(p-1, c')$ we can reconstruct $r \bmod p-1$ via the Chinese remainders theorem.

The probability of getting good pairs twice in two consecutive runs of the algorithm is $\Omega(1)$. Conditioned on that, what is the probability of ending up with c, c' having no common factor?

This is tricky issue and, indeed, c and c' will always be even. However, it can be argued that c, c' have $\Omega(1)$ probability of having distinct factors except possibly for the first $O(1)$ primes. When we reconstruct r with the Chinese remainder theorem, we will try all possible values of r modulo those primes.

Overall, two executions of the algorithm give us probability $\Omega(1)$ of generating a list of values that include r. After $O(1)$ iterations, we get a list that has a high probability of including r. It is then possible to compute modular exponentiation for each candidate in the list and find the correct r.