Shor's Algorithm

Andreas Klappenecker

Factoring Integers

Given an integer n that is not prime, the goal is to find a nontrivial factor of n.

Main Idea (behind most Factoring Algorithms)

Given a positive integer n.

If you can find integers a and b such that

- n divides $a^2 - b^2 = (a+b)(a-b)$
- $a \not\equiv \pm b \bmod n$

then $\gcd(a \pm b, n)$ yields a nontrivial factor of n.

Example 1

Let n = 1271

Given a = 36 and b = 5, we have

- n divides $36^2-5^2 = 1271$
- $36 \not\equiv \pm 5 \bmod 1271$

Thus, we get gcd(36-5, 1271) = 31 and gcd(36+5,1271)=41

In fact, 1271 = 31 * 41.

Problem: How can we find suitable integers a and b?

Example 2

Let $n = 15$.

For $a = 14$ and $b = 1$

Then n divides $(a^2-b^2) = 196 - 1 = 195 = 15*13$

but $14 = a \equiv -b = -1 \bmod n$.

Here we fail to get a nontrivial factor as

$$\gcd(a-b,n)=1 \text{ and } \gcd(a+b,n)=n.$$

Main Idea behind Shor's Algorithm

Given a positive integer n.

If you can find an integer a such that

- n divides $a^2 - 1^2 = (a+1)(a-1)$, equivalently, $a^2 \equiv 1 \bmod n$
- $a \not\equiv \pm 1 \bmod n$

then $\gcd(a \pm 1, n)$ yields a nontrivial factor of n.

How can we find a suitable a?

Order

Let c be a integer such that gcd(c,n)=1.

The smallest positive integer r such that

$$c^r \equiv 1 \bmod n$$

is called the order of c modulo n.

Example

Let n=15.

We determine the order of 2 mod n.

$2, 2^2, 2^3, 2^4 \equiv 1 \bmod 16$

Thus, the order of 2 mod n is 4.

Chinese Remainder Theorem

Chinese Remainder Theorem: Let p and q be coprime integers. Then

$x \equiv a \bmod p$

$x \equiv b \bmod q$

has a unique solution x in the range 0 <= x < pq.

Corollary. There are four different solutions to $x^2 \equiv 1 \bmod pq$, since

$x \equiv \pm 1 \bmod p$

$x \equiv \pm 1 \bmod q$

has four different solutions. Ex: $n=3*5$, $x_1 = 1$, $x_2=14$, $x_3 = 4$, $x_4 = 11$

Reduction

Goal: Factor n.

Choose an integer c such that gcd(c,n)=1. Compute the order r of c.

If r is even and $c^{r/2} \not\equiv -1 \bmod n$, setting $a = c^{r/2}$ and $b = 1$ yields

- n divides $a^2-b^2 = c^r-1$
- $a \not\equiv \pm b \bmod n$, as $c^{r/2} \not\equiv \pm 1 \bmod n$

Therefore, $\gcd(c^{r/2} \pm 1, n)$ yields a factor of n.

Probability to Succeed

Lemma. If $n=\Pi_{i=1}^{k} p_i^{a(i)}$ with p_i odd, then an element c chosen uniformly at random from { c | 0 <= c < n, gcd(c,n)=1 } will have even order r and satisfy $c^{r/2} \not\equiv -1 \bmod n$ with probability $\geqq 1-1/2^{k-1}$.

Indeed, let r(i) denote order of c mod $p_i^{a(i)}$, and let d(i) denote the largest power of 2 dividing r(i).

If r is odd, then d(i)=1 for all i.

If r is even and $c^{r/2} \equiv -1 \bmod n$, then $c^{r/2} \equiv -1 \bmod p_i^{a(i)}$, and we can conclude that r(i) divides r but does not divide r/2. Thus, d(i)>1. Furthermore, all d(i) must all be equal, since r = lcm(r(1),...,r(k)).

In summary, the algorithm fails if and only if d(1)=...=d(k).

The multiplicative group mod $p_i^{a(i)}$ is cyclic for odd p_i. Therefore, the probability that a random element c in this multiplicative group has order divisible by d(i) is <= 1/2. For c chosen uniformly at random all d(i) with 1<i<=k are equal to d(1) with probability <= $1/2^{k-1}$. q.e.d.

Summary

Given an integer n.

If n is even, then return 2

else if n is a power of a prime p, then return p.

Choose c from { c | 1<c<n, gcd(c,n)=1 } uniformly at random.

Calculate order r of c mod n.

If r is even and $c^{r/2} \not\equiv -1 \bmod n$, then return $\gcd(c^{r/2}-1,n)$

otherwise return "fail"