Shor's Algorithm & Quantum Cryptography

P. Beggs J. Hill

Department of Mathematics and Computer Science
Hendrix College

December 3, 2024

Recall that:

Recall that:

Discrete Logarithm Problem (DLP): the problem of finding x given $g^x \mod p$.

Recall that:

- Discrete Logarithm Problem (DLP): the problem of finding x given $g^x \mod p$.
- Time complexity: DLP = $\mathcal{O}(2^n)$.

Introduction
Background

Recall that:

- Discrete Logarithm Problem (DLP): the problem of finding x given $g^x \mod p$.
- Time complexity: DLP = $\mathcal{O}(2^n)$.
- Order r: smallest integer $r \geq 1$ such that $x^r \equiv 1 \mod n$

Classical bits: 0 or 1

[We could talk about how “steps” are calculated for big $\mathcal{O}$ time complexity.]

Quantum bits (qubits): Simultaneous values between 0 and 1

[Add some more here about state changes and superpositioning, maybe? This overview should be broad. We can expand upon this information in the Quantum Computers section.]

Classical Computers

Time Complexity

The fastest known algorithm (number field sieve) has time complexity $L_{4096}[\frac{1}{3}, c]$ (where $c < 1.923$) to decrypt a 4096-bit key-size DLP. For total steps, solve:

$$L_p\left[\frac{1}{3}, c\right] = e^{\left(c(\ln p)^{1/3}(\ln \ln p)^{2/3}\right)}.$$ [1]

[1] Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2018). *Handbook of applied cryptography.* CRC press.

Classical Computers

Time Complexity

The fastest known algorithm (number field sieve) has time complexity $L_{4096}[\frac{1}{3}, c]$ (where $c < 1.923$) to decrypt a 4096-bit key-size DLP. For total steps, solve:

$$L_p\left[\frac{1}{3}, c\right] = e^{\left(c(\ln p)^{1/3}(\ln \ln p)^{2/3}\right)}.[1]$$

For $c = (64/9)^{1/3} \approx 1.923$ and $p = 4096$, the total steps would be 10^{155}.

[1] Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2018). *Handbook of applied cryptography.* CRC press.

Using Shor's Algorithm, the DLP can be solved in polynomial time using a quantum computer. For key size 4096, it would take

Quantum Computers

Time Complexity

Using Shor's Algorithm, the DLP can be solved in polynomial time using a quantum computer. For key size 4096, it would take

$$\mathcal{O}\left((\log p)^3\right) = \mathcal{O}\left((\log 4096)^3\right) = 6.8 \cdot 10^{10} \text{ steps.}$$

Two-state representation: $|0\rangle$ and $|1\rangle$

Pure states: $\alpha\,|0\rangle + \beta\,|1\rangle$

Constraint: $|\alpha|^2 + |\beta|^2 = 1$

Two-state representation: $|0\rangle$ and $|1\rangle$

Pure states: $\alpha\,|0\rangle + \beta\,|1\rangle$

Constraint: $|\alpha|^2 + |\beta|^2 = 1$

n-Component System

$$\sum_{i=0}^{2^n-1} \alpha_i\,|s_i\rangle\,, \quad \text{where} \sum |\alpha_i|^2 = 1$$

Shor's Algorithm

Overview

Purpose: Find non-trivial factors p and q of N

Applications:

- Integer factorization
- Discrete logarithm in $\mathbb{F}_p^*$
- Elliptic curve discrete logarithm

Runs in polynomial time (quantum)

Shor's Algorithm

Algorithmic Steps

1. Change problem of factoring into finding the order r.

Shor's Algorithm

Algorithmic Steps

1. Change problem of factoring into finding the order r.
2. Use the Quantum Fourier Transform to extract periodicity of $f(x) = a^x \mod N$.

Shor's Algorithm

Algorithmic Steps

1. Change problem of factoring into finding the order r.
2. Use the Quantum Fourier Transform to extract periodicity of $f(x) = a^x \mod N$.
3. Once r is found (and it is even), use it in the computation of $\gcd(a^{r/2} - 1, N)$.

Key Component

For $0 < a < q$:

$$\frac{1}{q^{1/2}} \sum_{c=0}^{q-1} |c\rangle \exp(2\pi iac/q)$$

Choose q: power of 2 between n^2 and $2n^2$

Probability of state $|c\rangle$ is high when:

$$\left| c - \frac{d}{r} \right| < \frac{1}{2q}$$

Shor's Algorithm

Algorithmic Steps in Detail

1. Choose random integer $a < N$ (to ensure a and N are co-prime).

Shor's Algorithm

Algorithmic Steps in Detail

1. Choose random integer $a < N$ (to ensure a and N are co-prime).
2. Check if a is already a factor of N. If so, then the problem is solved.

Shor's Algorithm

Algorithmic Steps in Detail

1. Choose random integer $a < N$ (to ensure a and N are co-prime).
2. Check if a is already a factor of N. If so, then the problem is solved.
3. Otherwise, find the order r of $a \mod N$ using quantum super-positioning and interference. (Remember that the order is the smallest integer such that $a^r \equiv 1 \mod N$.)

Shor's Algorithm

Algorithmic Steps in Detail

1. Choose random integer $a < N$ (to ensure a and N are co-prime).
2. Check if a is already a factor of N. If so, then the problem is solved.
3. Otherwise, find the order r of $a \mod N$ using quantum super-positioning and interference. (Remember that the order is the smallest integer such that $a^r \equiv 1 \mod N$.)
4. Once r is found, compute the factors of N using r. If r is even and

   $$a^{r/2} \not\equiv -1 \mod N,$$

   then the factors are

   $$p = \gcd(a^{r/2} - 1, N) \qquad \text{and} \qquad q = \gcd(a^{r/2} + 1, N).$$

Cryptographic Implications

Vulnerable

RSA

Classical Elgamal

Elliptic curve Elgamal

Cryptographic Implications

Vulnerable

- RSA
- Classical Elgamal
- Elliptic curve Elgamal

Still Secure

- Lattice-based cryptosystems
- Shortest vector problems
- Closest vector problems

Challenges & Future

[Right now, this slide consists of filler. I don't really know what any of this stuff does, I just grabbed some crap off the internet and threw it in here.]

- Building functioning quantum computers
- Decoherence control
- Quantum cryptography applications:
  - Heisenberg uncertainty principle
  - Entanglement of quantum states
  - Secure key exchange