Breaking RSA encryption - Shor's Algorithm

PHYS/CSCI 3090

Prof. Alexandra Kolla
Alexandra.Kolla@Colorado.edu
ECES 122

Prof. Graeme Smith
Graeme.Smith@Colorado.edu
JILA S326

https://home.cs.colorado.edu/~alko5368/indexCSCI3090.html
Come see us!

- Alexandra Kolla/ Graeme Smith: Friday 3:00-4:00 pm, JILA X317.
- Ariel Shlosberg: Tu/Th 2:00-4:00pm, DUANG2B90 (physics help room)
- Steven Kordonowoy: Th 11am-12pm, ECAE 124.
- Matteo Wilczak: Wednesday, 1-2pm, DUANG2B90 (physics help room)
Exam coming up!

- Midterm 2 March 18! (next week, on Wednesday)
- Focused on Chapter 6 and 3.
Last Class

- Finishing period Finding
- In class exercise without the offset
Today

- Finish Period finding in class
- Finish correctness argument
- Start on connections with Factoring
Let $\omega = e^{2\pi i/N}$ a primitive $N$-th root of unity

$f = (f(0), f(1), \ldots, f(N-1))$ a function

The Discrete Fourier Transform of $f$ is defined as

$$F = (F(0), F(1), \ldots, F(N-1))$$

Where $F(k) = \sum_n \omega^{kn} f(n)$
QFT

- Say we have n qubits. (So $2^n$ possible basis vectors).
  \[ \omega = e^{\frac{2\pi i}{2^n}} \]
  \[ U_{FT} |x\rangle_n = \frac{1}{n} \sum_y \omega^{xy} |y\rangle_n \]
QFT

- \( f = (f(0), f(1), \ldots, f(N - 1)) \), \( N = 2^n \)
- \( f = \sum_x f(x) |x\rangle_n \)
- \( U_{FT}(\sum_x f(x) |x\rangle_n) = \sum_y F(y) |y\rangle_n \)
- \( F(y) = \frac{1}{n} \sum_x \omega^{xy} f(x) \)
- So \( U_{FT}(\sum_x f(x) |x\rangle_n) = \frac{1}{n} \sum_y \sum_x \omega^{xy} f(x) |y\rangle_n \)
The Algorithm

- Assume, for now, that $r$ divides $N$.
- The superposition $\left( \sum_k |x_0 + kr\rangle \right)$ consists of $N/r$ different multiples of $r$.
- So my state after collapse is

$$
\frac{1}{\sqrt{\frac{N}{r}}} \left( \sum_{k=0 \text{ to } \frac{N}{r}} |x_0 + kr\rangle \right)
$$
The Algorithm

- Theorem:
  Suppose the input to QFT is periodic with period \( r \), for some \( r \) that divides \( N \). Then the output will be a multiple of \( N/r \), and it is equally likely to be any of the \( r \) multiples of \( N/r \).

- Now we repeat the experiment a few times and take GCD of all the indices returned, we get \( N/r \) (and thus \( r \)) w.h.p.
The Algorithm

- 4) Apply QFT to \( \frac{1}{\sqrt{N}} \sum_j |x_0 + jr\rangle \). First assume \( x_0 = 0 \).
- Claim (ex. in class):

  If \( |a\rangle = \frac{1}{\sqrt{N}} \left( \sum_j |jr\rangle \right) \) then

\[
U_{FT}(|a\rangle) = |\beta\rangle = \frac{1}{\sqrt{r}} \left( \sum_{j=0 \text{ to } r-1} \frac{|jN\rangle}{r} \right)
\]
Sums of roots of unity.

What is the sum $1 + \omega^{jr} + \omega^{2jr} + \omega^{3jr} + \cdots + \omega^{(N/r-1)jr}$, where $\omega = e^{2\pi i/N}$

$$1 + \omega^{jr} + \omega^{2jr} + \omega^{3jr} + \cdots + \omega^{(N/r-1)jr} = \frac{N}{r} \text{ if } jr = 0 \mod N$$

$$1 + \omega^{jr} + \omega^{2jr} + \omega^{3jr} + \cdots + \omega^{(N/r-1)jr} = 0 \text{ otherwise}$$
\[ |a\rangle = \sum_{j=0}^{r-1} \sqrt{\frac{n}{N}} |j\rangle = \sum_{j=0}^{r-1} \sqrt{\frac{N}{n}} |j\rangle + \sum_{x} 0.1 |x\rangle \]

\[ f_r(x) = \begin{cases} 0 & x \neq 0 \mod r \\ \frac{r}{N} \text{ otherwise} \end{cases} \]

\[ \text{claim: } |a\rangle = \sum f_r(x) |x\rangle = \sum f_r(x) |x\rangle + \sum_{x \equiv 0 \mod r} \frac{r}{N} |x\rangle \]

\[ \text{FFT} \left( \sum_{x \equiv 0 \mod r} f_r(x) |x\rangle \right) = \sum \delta(x) |y\rangle \]

\[ F_r(y) = \sum_{j=0}^{r-1} \omega^{xy} f(j) \]
\[
\begin{align*}
\mathbf{f}_r(y) &= \sum_{\ell=0}^{r-1} \mathbf{f}_r(\ell) \omega^\ell y \\
\mathbf{f}_r(y) &= \frac{1}{N} \sum_{\ell=0}^{r-1} \omega^{\ell+r} y \\
\sum_{\mathbf{f}_r(y)} |y \rangle &= \frac{1}{\mathbf{f}_r} \sum_{i=0}^{N-1} |e^{i \omega^i/r} \rangle \\
\mathbf{f}(\ell) &= \begin{cases} \\
\frac{\mathbf{f}_r}{N} & \text{if } \ell = 0 \\
\frac{\mathbf{f}_r}{N} & \text{if } \ell \neq 0 \mod N \\
\frac{\mathbf{f}_r}{N} & \text{if } \ell = 0 \mod N \\
\frac{\mathbf{f}_r}{N} & \text{for } y = i \cdot N/r \\
0 & \text{otherwise}
\end{cases}
\end{align*}
\]
Some probability.

If I take a sample from the uniform distribution on \(\{0, 1, \ldots, r-1\}\), what is the probability (at most) that the number I get is a multiple of an integer \(j\)?

A) 0  
B) \(1/j\)  
C) 1  
D) I don't know
The Algorithm

Lemma:
Suppose I take $s$ independent samples drawn uniformly from $0, \frac{N}{r}, \ldots, \frac{(r-1)N}{r}$. Then with probability at least $1 - \frac{r}{2^s}$, the GCD of these samples is $N/r$.

\[ \begin{align*}
\text{e.g.: } & \quad \frac{N}{r} = 3 \\
\text{r: } & \quad 100
\end{align*} \]

4. 0% 8. N/r (bad)
Some more probability.

Suppose I flip $s$ coins, each of which has probability of heads *less* than $p$. What is an upper bound on the probability that all of them come out heads?

(the smallest)

A) $p$

B) $p^s$

C) $1 - p$

D) $1 - p^s$
Some more more probability.

Suppose I have a set of $r$ bad events, $\{E_j\}_{j=0}^{r-1}$, and each of the $E_j$ happens with probability at most $p$. What is and upper bound on the probability that none of the bad events happen?

A) $p$

B) $rp$

C) $1 - p$

D) $p^r$

$\Pr(\bigcup E_j) = ? \leq \sum \Pr[E_j]$
The Algorithm

**Lemma:**
Suppose I take $s$ independent samples drawn uniformly from $0, \frac{N}{r}, \ldots, \frac{(r-1)N}{r}$. Then with probability at least $1 - \frac{r}{2^s}$, the GCD of these samples is $\frac{N}{r}$. 