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 Kordonowy: Th 11am-12pm, ECAE 124.
- Matteo Wilczak: Wednesday, 1-2pm, DUANG2B90 (physics help room)

Last Last Last Class

- Simon's problem, dealing with periodic functions under bitwise addition mod 2

Today

- Shor's problem

- Find the period r of a function f on the integers that is periodic under ordinary addition.

The problem

- One is told that f is periodic under ordinary addition, $f(y) = f(x)$, if $y = kr + x$, for any integer k.
- i.e x and y differ by an integral multiple or period r.
- The problem is to find the period r.

The problem

- Finding the period is not always easy.
- The function can be virtually random within the period r
- Best known classical algorithms take time exponential ($O(2^{n^{\frac{1}{3}}})$), where n is the number of bits of r.
- Shor's algorithm takes time a little less than n^3.
- Any computer that can efficiently find periods, breaks RSA
- Would be enormous threat to the security of military and commercial communications.

Primes

I am given a number N and I want to determine if N is a prime.
Consider the following algorithm:
For all integers i from $1\ to\ N/2$, check if i divides N.
If I find an i that divides N, output "NO"
Otherwise, output "YES".

The running time of this algorithm is

A) Linear in the input size.

B)Exponential in the input size

C) Quadratic in the input size

D) Polynomial in the input size.

Periodic?

Assume we have a function $f(x) = b^x \ (mod\ N)$, for some integers b and N
Is f periodic?

A) Yes

B) No

C) Maybe

D) It depends on b.

Periodic?

Assume we have a function $f(x) = b^x \ (mod\ N)$, for some integers b and N With gcd (b,N)=1. Is f periodic?

A) Yes

B)No

C) Maybe

D) It depends on b.

Some number theory

- Assume we have a function $f(x) = b^x \ (mod\ N)$,

for some integers b and N that are coprime.

- Fact: there is an integer r such that $b^r \equiv 1 \ (mod\ N)$
- So $f(x + r) = b^{x+r} = b^x b^r = b^x = f(x) \ (mod\ N)$

- Also, $f(x + kr) = f(x)$ for any multiple k of r. (ex)

Congruences

Which of the following congruences is true?

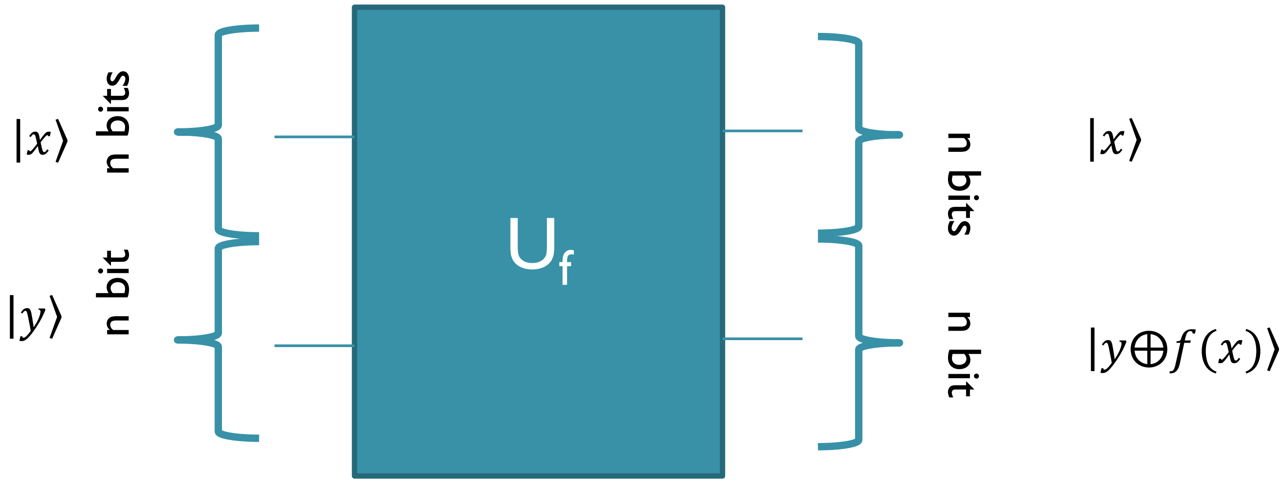
A) $6 \equiv 1 \ (mod\ 4)$

B) $5 \ \equiv 0 \ (mod\ 4)$

C) $15 \ \equiv 3 \ (mod\ 4)$

D) $12 \ \equiv 1 \ (mod\ 4)$

The setup



- We have a function $f(x) = b^x \ (mod\ N)$, which is periodic with some period r.
- We want to find r fast.
- Classically?
- Try to find two different values x, y that f(x)=f(y).
- Will learn something about the period this way (x,y differ by multiple of the period)
- Really inefficient even classically!!

Number Theoretic Preliminaries

- We have a function $f(x) = b^x \ (mod\ N)$, which is periodic with some period r.
- We want to find r fast.
- Let N=pq, the product of two primes. Assume it can be represented with n_0 bits (2^{n_0} is smallest power of 2 that exceeds N).
- If N is a 500 digit number, as in popular crypto applications, then n_0 ~1700
- To have an appreciable probability of finding r by random searching, we would need to evaluate f an exponential number of times in n_0
- Quantum parallelism gets us very close to evaluating Uf only once!
- And can solve the problem exactly in polynomial in n_0 time.

The Algortihm

- 1) Prepare: $(H^{\otimes n} \otimes \mathrm{I})\ |0\rangle_n\ \ |0\rangle_{n_0} = \frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\ |0\rangle_n$

- 2) Single application of oracle (can be done efficiently for our f):

$$\mathrm{U_f}\left(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\ |0\rangle_{n_0}\right) = \frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\ |f(x)\rangle_{n_0}$$

- 3)Measure output register: If I get some value of f, say $f(x_0)$, then input register is…

The collapse step

Assume x_0 is the smallest value such that $f(x_0)=f_0$.
What is the input register after we measure the output register and we get a value, say f_0?

A) $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 + r\rangle)$

B) $\frac{1}{\sqrt{m}}(\sum_k |x_0 + kr\rangle)$

C) $|x_0\rangle$

D) $\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n$

The Algorithm

- 1) Prepare: $(H^{\otimes n} \otimes \mathrm{I})\ |0\rangle_n \quad |0\rangle_{n_0} = \frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\ |0\rangle_n$

- 2) Single application of oracle (can be done efficiently for our f):

$\mathrm{U_f}(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\ |0\rangle_{n_0})=$
$\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\ |f(x)\rangle_{n_0}$

- 3)Measure output register: If I get some value of f, say f(x_0), then input register is $\frac{1}{\sqrt{m}}(\sum_k |x_0 + kr\rangle)$

Reminder: The Algortihm for Simon's

- 1) Prepare: $(H^{\otimes n} \otimes \mathrm{I})\ |0\rangle_n\ \ |0\rangle_n = \frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\ |0\rangle_n$

- 2) Orace: $\mathrm{U_f}(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\ |0\rangle_n) = \frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\ |f(x)\rangle_n$

- 3)Measure output register: If I get some value of f, say f(x_0), then input is $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle)$

Simons

3)Measure output register: If I get some value of f, say f(x_0), then input is $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle)$

Simons

- 3) Measure output register: If I get some value of f, say f(x_o), then input is $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle)$
- 4)Apply $H^{\otimes n}$ to input register

$$Recall\text{:}\ H^{\otimes n}\ |x\rangle_n \quad = \ \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} (-1)^{y\cdot x}\ |y\rangle_n$$

$$H^{\otimes n}\ \frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle) =$$

$$\frac{1}{2^{(n+1)/2}} \sum_{y=0}^{2^n-1} ((-1)^{y\cdot x_0} + (-1)^{y\cdot(x_0 \oplus a)})|y\rangle_n$$

Simons

$H^{\otimes n}\ \frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle)$=

$\frac{1}{2^{(n+1)/2}} \sum_{y=0}^{2^n-1}((-1)^{y\cdot x_0} + (-1)^{y\cdot(x_0\oplus a)})|y\rangle_n$

- Since $(-1)^{y\cdot(x_0\oplus a)} = (-1)^{y\cdot x_0}(-1)^{y\cdot a}$, the coefficient of $|y\rangle$ is zero if $y \cdot a = 1$ and $2(-1)^{y\cdot x}$ if $y \cdot a = 0$
- State is: $\frac{1}{2^{(n-1)/2}} \sum_{y\cdot a=0}(-1)^{y\cdot x_0}\,|y\rangle_n$,
- Only the y's such that $a \cdot y = 0$ survive!
- If we measure the input register, we learn with equal probability any of the values of y such that $a \cdot y = 0$.

Number Theoretic Preliminaries

- For Simon's problem, we moved the information of the "period" a to the phase.
- Can we do something similar here?
- The answer is Quantum Fourier Transform! (to be continued…)