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 2:00-4:00 pm, DUANG2B90 (physics help room)
- Steven Kordonowy: Th 11am-12pm, ECAE 124.
- Matteo Wilczak: Wednesday, 1-2 pm, 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 correctness
- Reduce Factoring to Period Finding
- Intro to Cryptography
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 $\frac{N}{r}$. 

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

- **Definition:**
  A non-trivial square root of $1 \mod N$ is any integer $x \neq \pm 1 \mod N$, such that $x^2 = 1 \mod N$.

Claim: If we can find a non-trivial square root of $1 \mod N$, then it is easy to decompose $N$ into a product of two nontrivial factors and repeat.
Factoring

Lemma: If $x$ is a non-trivial square root of 1 mod $N$, then $\gcd(x+1,N)$ is a non-trivial factor of $N$. 
Factoring

Lemma: If $x$ is a non-trivial square root of 1 mod $N$, then $\gcd(x+1,N)$ is a non-trivial factor of $N$

Example: find a non-trivial root of 1 for $N=15$. Find the factors of $N$ using above lemma
Factoring

• Definition:
The order of $x \mod N$ is the smallest positive integer $r$ such that $x^r = 1 \mod N$.

Claim:
Let $N$ be an odd composite, with at least two distinct prime factors. Let $x$ be chosen uniformly at random between 0 and $N - 1$. If $\gcd(x, N) = 1$, then with probability at least $\frac{1}{2}$, the order $r$ of $x \mod N$ is even, and moreover, $x^{r/2}$ is a nontrivial square root of 1 mod $N$. 
Factoring

- The claim implies that if we could compute the order $r$ of a randomly chosen element $x$ mod $N$, then there’s a good chance that this order is even and that $\gcd(x^{r/2} + 1, N)$ is a factor of $N$. 
Some number theory.

What is 575 (mod 7)

A) 1  

B) 3

C) 5  

D) 0
One of the earliest known cryptographic ciphers was used by Julius Caesar. His strategy was to shift each letter of the alphabet forward 3 places, wrapping around when you get to the end.

In this scheme, for example: $A \mapsto D$, $K \mapsto N$, $Y \mapsto B$

This is often called a **Caesar Cipher** or a **Shift Cipher**.

- Mathematically, we can accomplish this by assigning to each letter a number between 0 and 25. For example:
  
  $A \mapsto 0$, $K \mapsto 10$, $Y \mapsto 24$

- The encoding can be done by passing the value through a **shift function modulo 26**:
  
  $f(p) = (p+3) \mod 26$
In general, for a shift \( k \) we can use the function

\[
f(p) = (p+k) \mod 26
\]

We can encode a message by:
1. Convert letters to numbers between 0 and 25
2. Pass each value through \( f(p) \)

- **Example:** Encode HELLO WORLD using a shift=5 cipher
In general, for a shift $k$ we can use the function

$$f(p) = (p+k) \mod 26$$

We can encode a message by:
1. Convert letters to numbers between 0 and 25
2. Pass each value through $f(p)$

**Example:** Encode HELLO WORLD using a shift=5 cipher

1. Convert to numbers: HELLO WORLD $\rightarrow$ 7 4 11 11 14 22 14 17 11 3 1. Shift: 12 9 16 16 19 1 19 22 16 8

The encoded message is: MJQQT BTWQI
How do we decode a message like MJQQT BTWQI?

If we know the shift, then it’s easy - just run the message through the inverse:

\[ f^{-1}(p) = (p - k) \mod 26 \]

Why is this not a very secure cipher?
The Affine Cipher

- Instead of only shifting, **multiply** and then **shift**
  \[ f(p) = (ap + b) \mod 26 \]
  where \( a \) and \( b \) are integers with \( \gcd(a, 26) = 1 \)

- S’pose we know \( a \) and \( b \) (i.e., we have the **key**) - how could we decode a message?
Some crypto

S’pose we know $a$ and $b$ (i.e., we have the key) - how could we decode a character $c$ (say $p$ is the original character that got encoded to $c$)?

$$f(p) = (ap + b) \mod 26$$

A) $p = a^{-1}(c - b) \mod 26$
B) $p = a^{-1}b \mod 26$
C) $p = a^{-1}c \mod 26$
D) $p = (c - b) \mod 26$
S’pose we know $a$ and $b$ (i.e., we have the **key**) - how could we decode a message?

- S’pose we have an encrypted character $c$ that we know must satisfy
  \[ c \equiv ap + b \pmod{26} \]
- Then we need to solve this congruence for $p$. So subtract $b$ from both sides
  \[ c - b \equiv ap \pmod{26} \]
- Now we need the inverse of $a$ (modulo 26), which we know exists because $\gcd(a, 26) = 1$. Call the inverse $\overline{a}$, and we have
  \[ p \equiv \overline{a} (c - b) \pmod{26} \]
Some crypto.

Use an affine cipher with $a=7$ and $b=13$ to encrypt the letter $K$.

A) A  
B) W  
C) H  
D) F
Example: Use an affine cipher with $a=7$ and $b=13$ to encrypt the letter $K$.

Example: Find a decryption formula for this affine cipher and use it to decrypt the character $F$. 
Example: Use an affine cipher with $a=7$ and $b=13$ to encrypt the letter $K$.

Solution: The numerical value for $K$ is 10, so we have

$K \mapsto a \cdot 10 + b = 7 \cdot 10 + 13 = 83 \equiv 5 \pmod{26} \mapsto F$

Example: Find a decryption formula for this affine cipher and use it to decrypt the character $F$.

Solution: Recall from earlier that we had the formula: $p \equiv \overline{a} (c - b) \pmod{26}$

So we need the inverse of $a = 7$ (modulo 26)
Cryptography warm-up

Example: Find a decryption formula for this affine cipher and use it to decrypt the character $F$. 
**Example:** Find a decryption formula for this affine cipher and use it to decrypt the character F.

So we need the inverse of $a = 7 \pmod{26}$:

$$
26 = 3 \cdot 7 + 5 \\
7 = 1 \cdot 5 + 2 \\
5 = 2 \cdot 2 + 1
$$

... and in reverse ...

$$
1 = 5 - 2 \cdot 2 \\
= 5 - 2 \cdot (7 - 1 \cdot 5) = 3 \cdot 5 - 2 \cdot 7 \\
= 3 \cdot (26 - 3 \cdot 7) - 2 \cdot 7 = 3 \cdot 26 - 11 \cdot 7
$$

So the inverse of 7 (modulo 26) is -11
Example: Find a decryption formula for this affine cipher and use it to decrypt the character $F$.

So the inverse of 7 (modulo 26) is -11

\[
\bar{a} \equiv (c - b) \pmod{26}
\]

\[
\equiv -11 \cdot (5 - 13) \pmod{26}
\]

\[
\equiv 88 \pmod{26}
\]

\[
\equiv 10 \pmod{26} \rightarrow K
\]
Example: Find a decryption formula for this affine cipher and use it to decrypt the character $F$.

So the inverse of 7 (modulo 26) is -11

Plugging into the decryption formula we have: \( p \equiv \frac{c - b}{a} \pmod{26} \)
\( \equiv -11 \cdot (5 - 13) \pmod{26} \)
\( \equiv 88 \pmod{26} \)
\( \equiv 10 \pmod{26} \) \( \mapsto K \)

FYOG: Encrypt HELLO WORLD with an affine cipher with $a=5$ and $b=17$. Derive the decryption formula and check that your encrypted message decrypts back properly.
Systems of congruences and Cryptography-Lite

We need:

- **Back substitution** and **Chinese Remainder Theorem** provide two avenues to solve systems of congruences
- **Fermat’s Little Theorem** offers a nice way to calculate giant numbers quickly
- **Affine** and **shift cyphers** -- two relatively simple examples for encoding/decoding messages; based on shifting and multiplying your intended message

Next time:

- **Cryptography-Heavy** and bringing it all this number theory together: **RSA**