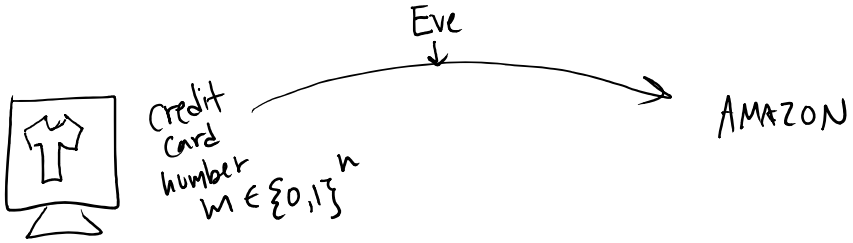
S. KIMMEL

Using Modular Arithmetic for Public Key

↓ mod K



I encript m using function f

$m' = f(m)$

↳ Evesdropper can't figure out m given $f(m)$

Amazon decrypts m' using function f'

$f'(m') = m$

Old Strategy: Shift Cipher

Meet up with AMAZON, choose $a, b \in \{0, \cdots, 25\}$.

To send a message

A → 0
B → 1
⋮
Z → 25

$$f(p) = ap + b \pmod{26}$$

↑
p a letter

S. KIMMEL

ex: $a = 7$, $b = 3$:

$f(p) = 7p + 3 \mod 26$

Q: What does D (4) get encrypted as?

A: 4 B: 5 C: 6 D: 8

$7 \cdot 4 + 3 = 31 = 26 \cdot 1 + 5$

$31 \equiv 5 \mod (26)$

To decrypt:

$15(p' - 3) \mod 26$

If $p' = 5$ $15(2) = 30 \equiv 4 \mod 26$

Why $15((7p + 3) - 3) \equiv 15 \cdot 7p \pmod{26}$

but $15 \cdot 7 = 105 \equiv 4 \cdot 26 + 1 \equiv 1 \pmod{26}$

$\equiv p \pmod{26}$

S. KIMMEL

- With regular arithmetic, $7 \cdot x = 1 \Rightarrow x = \frac{1}{7}$
- " mod 26 " $7 \cdot x \equiv 1 \pmod{26} \Rightarrow x = 15$

Shift Cypher is not secure. Quickly test all combos of

$$f(p) = ap + b \pmod{26}$$

26 choices, 26 choices

Better: Random key

Amazon and I would get together and choose a random string $\{0,1\}^n = k$ for key. Convert m to binary.

Me: $m = 101$ $k = 110$ $m'_i \equiv m_i + k_i \pmod 2$

$m' = 011$

Amazon: $m_i = m'_i + k_i$

$m' \oplus k = 101$

Q: Why can't Eve figure out m from m'?

S.KIMMEL

Problem: Can't meet up with AMAZON :(, Can't reuse key

Solution: Public Key

Amazon 2 Keys, public (sends to world) K_{pub}
private (keeps secret) K_{pri}

To send m to amazon

1) Create $m' = f(m, K_{pub})$ (lock up)
2) Send to Amazon
3) Amazon does $f'(m', K_{priv})$ to get m (unlock)

RSA Public Key

To create K_{pub}, K_{pri}

1. Choose p, q large primes ex: 43, 59
   (really 200 digits)
2. Choose e relatively prime to $(p-1)(q-1)$
   ↓
   no common prime factors

S.KIMMEL

ex: 12, 35 are relatively prime

$12 = 2 \cdot 2 \cdot 3$ ← nothing in common

$35 = 5 \cdot 7$

$k_{pub} : (n = p \cdot q, e)$ $k_{pri} = p, q$

ex: $(n = 43 \cdot 59 = 2537, \ e = 13)$

Encryption (using n, e)

A → 00

⋮

Z → 25

- Break message into chunks so each chunk $\leq n$
- Each chunk m_i

$$m_i' = m_i^e \pmod n$$

- m' = chunks put back together.

S.KIMMEL

ex: STOP = 1819 | 1415

$2537 = 1415^{13} \pmod{2537}$

$2081 = 1819^{13} \pmod{2537}$

① Fast alg for modular exponentiation

$m' = 20812537$

Decryption

multiplicative inverse of e

Find d: $de \equiv 1 \bmod ((p-1)(q-1))$

② How to find d if know p, q, e

$m = (m_i')^d \bmod n$

Why? ③ Chinese Remainder Theorem

④ $a^{p-1} \equiv 1 \pmod{p}$

Punchline

$k_{pub} = (n = p \cdot q, e)$

$k_{pri} = (p, q)$

- Holder of p, q can calculate inverse of e mod $(p-1)(q-1)$ using Euclidean Alg.
- Why can't you get p, q if know n?
  → No fast algorithm to factor

BUT: Quantum Computers can factor quickly
⇒ Will break most public key crypto
⇒ Can read all past messages