CS 124 Lecture 14 — March 25, 2020

RSA and Crypto!

message x

Encoding $e(x)$

Decoding $d(e(x))$

x — Alice $\xrightarrow{e(x)}$ Bob — $d(e(x)) = x$

$\uparrow$ Eve

Information-Theoretic Approach - One Time Pad

$\oplus := \text{XOR}$

$$\begin{array}{r} 110101 \\ \oplus\ 011001 \\ \hline 101100 \end{array}$$

message x

random string r ← private

$|r| = |x|$

encoding $e(x) = x \oplus r$

decoding $d(e(x)) = e(x) \oplus r$

$= (x \oplus r) \oplus r = x$

Eve sees $e(x)$. Does she gain any information about x?

$\Pr(\text{message is } x \mid e(x))$

$= \Pr(\text{message is } x)$ ← original guess

* using one-time pad multiple times gives info about r!

$e(x)$

$e(y)$

$e(x) \oplus e(y)$

$= x \oplus r \oplus y \oplus r$

$= x \oplus y$

RSA - Public Key Cryptography

Based on computational hardness

→ If Eve could break RSA scheme, then she'd know how to solve a very hard class of problems

Needs:

- Generate big prime numbers (primality test)
- Fast Exponentiation (repeated squaring)
- Euclid's Algo + Extended Euclid's Algo

Euclid's Algo

Greatest Common Divisor

gcd (a, b) = largest int d, d divides a and b

= $d \mid a$, $d \mid b$

poly logarithmic
↳ time proportional to # of digits

gcd - Euclid (a, b)
if b == 0 return a
return gcd - Euclid (b, a mod b)

360, 84
84, 24
24, 12
12, 0

gcd (b, a mod b) = gcd (a, b)

O(log a) rounds
↓
must at least half every round

$b \leq a/2$: then in one round cut by 1/2

$b > a/2$: a mod b = $a - b < a/2$
then in two rounds decreases by 1/2

at most $2 \times \log_2 a$ rounds

most number of rounds when a is a Fib #

Extended Euclid's

$ee(a,b)$

returns $d = \gcd(a,b)$

and integers x, y $ax+by=d$

method to find multiplicative inverses

$\gcd(a,b)=1$

$\gcd(a,p)=1$

when $p \times a$

$ax \equiv 1 \mod p$

$ax+py=1$

$ax \equiv 1 \mod p$

RSA Protocol

Bob — public key

[private info]

512 bit

Bob chooses p, q primes

(of roughly equal length)

Bob computes $n = p \times q$

and finds random int e s.t.

[$e=3$]

$\gcd((p-1),(q-1),e)=1$

* (n,e) is Bob's public key

Bob's private info is

$d = e^{-1} \mod (p-1)(q-1)$

by extended Euclid's Algo

Alice takes

message is a number mod n

$e(x) = x^e \mod n$ ← by fast exponentiation

To decode, Bob takes

$d(e(x)) = (e(x))^d \mod n$

Claim. $d(e(x)) = x \mod n$

Pf. $d(e(x)) = x^{ed} \mod n$

e and d are multaplicative inverses mod $(p-1)(q-1)$

$d(e(x)) = x^{1+k(p-1)(q-1)} \bmod n$

Show: $x^{1+k(p-1)(q-1)} = x \bmod p = x \bmod q$

$x^{p-1} = 1 \bmod p$ by Fermat's Little Thm

if $x \mathrel{!=} 0 \bmod p$

$x^{k(p-1)(q-1)} = 1 \bmod p$

$x^{1+k(p-1)(q-1)} = x \bmod p$

same arg for q

How would Eve decode?

Factor n into $p \times q$ and compute d