CS 124 Lecture 15.

RSA = public key cryptography

Bob has 2 keys

k_e = public

k_d = private

k_e to encode, k_d to decode

Scenario:

- post public key
- others use it to encode and send messages
- private key is used to decode

RSA Tools

- primality testing
- exponentiation
- Euclid's algorithm

Greatest common divisor

Defn. Integers $a, b \geq 0$, then the gcd of a, b is the largest integer $d \geq 0$ that divides both.

notation: $d | a$, $d | b$
d divides a

ex:
$\gcd(360, 84) = 12$

gcd - Factoring

→ nobody knows a poly-time alg. for factoring

gcd - without factoring

Assume $a \geq b \geq 0$

```
Euclid (a, b)
  if b = 0 return a
  return (Euclid ( b, a mod b))
```

ex:

360, 84

↳ 84, 360 - 336 = 24

↳ 24, 84 - 72 = 12

↳ 12, 24 - 24 = 0

↳ return 12

correctness

$$\gcd(a, b) = \gcd(b, \underbrace{a \bmod b}_{a - kb})$$

mod ops and other arithmetic are poly-time in # of digits

$$d | a, \ d | b \Rightarrow d | b, \ d | a - kb$$

$$\Leftarrow$$

runtime

take two steps
1) a, b
2) $b, a \bmod b$
3) $a \bmod b, \ldots$

terminate after $2 \cdot \log_2 a$ steps

$a \bmod b \leq \frac{a}{2}$

① if $b \leq \frac{a}{2}$, done, the remainder is $< b$, thus $\leq \frac{a}{2}$

② if $b > \frac{a}{2}$, $a \bmod b = a - b < \frac{a}{2}$

Extended Euclid's Alg.

[redacted]

in addition to
$d = \gcd(a, b)$, get
integers x, y, s.t.
$ax + by = d$

$EE(a, b)$
if $b = 0$, return $(a, 1, 0)$
compute k such that $a = bk + (a \bmod b)$
$(d, x, y) = EE(b, a \bmod b)$
return $(d, \underbrace{y}_{x}, \underbrace{x - ky}_{y})$
new

[redacted]

correctness by induction

$b = 0$ (base case), $x = 1$, $y = 0$ $\quad d = a \cdot 1 = a$ ✓

$b \neq 0$ (inductive step)

IH: assume $(d, x, y) = EE(\overset{a'}{b}, \overbrace{a \bmod b}^{b'})$ is correct

then $\overset{a'x + b'y = d}{bx + (a \bmod b)y = d}$

$a \bmod b = a - kb$

then $bx + (a - kb)y = d$

$bx + ay - kby = d$

$\underbrace{a}_{}y + b\underbrace{(x - ky)}_{} = d$

new x, y

for
d, x, y [redacted] a', b'

↓

for
$d, y, x - ky$ [redacted] a, b

↑

last return call provides x and y for input a, b

EE used to find multiplicative inverses

ex: use EE to find multiplicative inverses

given p

what is $1000^{-1} \bmod p$

$EE(1000, p)$

$\gcd(1000, p) = 1$

$1000x + py = 1$

$\downarrow \bmod p$

$1000x + py = 1 \bmod p$

$1000x = 1 \bmod p$

multiplicative inverse [redacted]

$x = 1000^{-1} \bmod p$

RSA Assumes Factoring is Hard

Bob picks 2 large random primes p, q

Bob computes $n = p \cdot q$

Bob picks e (randomly, $e=3$), s.t.

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

Public key: (n, e)

n is published, but p, q remain private due to hardness of factoring

Private key: (p, q, d)

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

x = message

$1 \leq x \leq n$

$e(x) = x^e \bmod n$

repeated squaring

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

Prove $d(e(x)) = x$

$x^{ed} \stackrel{?}{=} x \mod n$ $\quad$ $d = e^{-1} \mod (p-1)(q-1)$

by def. of multiplicative inverses

$x^{1+k(p-1)(q-1)} \stackrel{?}{=} x \mod n$

fact a

primes (or co-primes)

$x = y \mod p$

$x = y \mod q$ $\Rightarrow$ $x = y \mod pq$

ex

2 mod 7 $\quad$ 2, 9, 16, 23, 30, 37

2 mod 5 $\quad$ 2, 7, 12, 17, 22, 27, 32, 37

$\Rightarrow$ 2 mod 35 $\quad$ 2, 37 ——

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

case $x = 0 \mod p$ ✓

case $x \neq 0 \mod p$

divide both sides by x (mod. arithmetic)

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

by mod. arithmetic

$x^{(p-1)} = 1 \mod p$ by FLT

(2) same proof for $x^{1+k(p-1)(q-1)} = x \mod q$

$\Rightarrow x^{1+k(p-1)(q-1)} = x \mod n$ ✓

by (1), (2), fact a