CSCI 3100

Recap

- HW due
- Last graded one back Monday
- Sample final
- Review Monday
- Final Friday: 8am
- EC due Monday

Next week:

I'll be in Wed + Thursday.

(Post hours on webpage or email to set up)

Basic Assumptions

1. Multiplication is easy.
2. Factoring is hard.

   ↳ meaning current alg are slow.

3. Generating prime numbers is easy.

   (Not obvious — but they are common)

   (more later)

(4) Modular exponentiation is easy:

Given n, m, e, can compute

$$c = m^e \bmod n$$

(5) Given prime factors, can do modular root extraction:

Given n, e, c & $n = pq$, can recover m given $m^e \bmod n$.

(6) Conjecture Without p & q, (5) is hard.

So: RSA (finally!)

<u>Bob:</u> Selects 2 large primes p & q

- Let $n = pq$
  - $\hookrightarrow \Phi(n) = (p-1)(q-1)$
- Select e & d s.t.
  - e and $\Phi(n)$ are relatively prime
  - $ed \equiv 1 \mod \Phi(n)$
    - $\hookrightarrow$ Extended Euc Alg

<u>Now</u>:
- (e, n) is public key
- d is private key (also p & q)

__Encryting__: Alice gets (e, n).

She takes a message M, with $0 < M < n$.

(chops into pieces)

Then:

$$C \leftarrow M^e \bmod n$$

(Remember public part: (e, n) was key)

Alice sends C to Bob

Decrypting: Bob gets C

$$C = M^e \mod n$$

Bob calculate:

$$C^d \mod n$$

Claim: $\overset{?}{=} M$

Why?

$$C^d \mod n = (M^e)^d \mod n$$

$$= M^{ed} \mod n$$

Know $ed \equiv 1 \mod \Phi(n)$

$$M^{(ed)} = M^{(k\Phi(n)+1)} \mod n$$

$$= \underbrace{\left(M^{\Phi(n)}\right)^k}_{\text{Euler thm}} \cdot M^1 \mod n$$

$$= 1^k \cdot M \mod n = M$$

Example:

Key Pair			**Key Pair Generation**			
Public key: $n = 55, e = 3$ *Private key: $n = 55, d = 7$*			*Primes: $p = 5, q = 11$* *Modulus: $n = pq = 55$* *Public exponent: $e = 3$* *Private exponent: $d = 3^{-1} \bmod 20 = 7$*			
Message	***Encryption*** $c = m^3 \bmod n$		***Decryption*** $m = c^7 \bmod n$			
m	$m^2 \bmod n$	$m^3 \bmod n$	$c^2 \bmod n$	$c^3 \bmod n$	$c^6 \bmod n$	$c^7 \bmod n$
0	0	0	0	0	0	0
1	1	1	1	1	1	1
2	4	8	9	17	14	2
3	9	27	14	48	49	3
4	16	9	26	14	31	4
5	25	15	5	20	15	5
6	36	51	16	46	26	6
7	49	13	4	52	9	7
8	9	17	14	18	49	8
9	26	14	31	49	36	9

So: Why secure?

Bob can decrypt!

He knows (secret) d.

Attacker Eve's goal:

Figure out d!

How? • Bob needed $\Phi(n)$, since

d is e's inverse mod n.

• Attacker knows n

(but not $\Phi(n)$).

How to find $\Phi(n)$?

So:

Whole thing is secure, as long as Eve can't get $\Phi(n)$, or p + q.

$(p-1)(q-1)$ $\hookrightarrow$ these would give d

Bad news: Factoring is NOT NP-Hard.

Best algorithm:

Number field sieve:

$$O\left(e^{\left(\frac{64}{9}\log n\right)^{1/3}(\log\log n)^{2/3}}\right)$$

Some practical notes

- RSA can be used to encrypt entire message (but usually isn't)
- Slow (compared to XOR-ing)
- Easier to break than AES or other symmetric protocols

- Also: I was assuming $(M, n) = 1$!

  Here, saved since $n = pq$, & M <u>will</u> be relatively prime to p or q.

- Can also be used for <u>digital</u> <u>signing</u>.

Continuing Work

- Actual RSA is a bit more complex
  (some n's, e's, d's, etc. are better than others!)
- Still in an "arms race" to break this
  - quantum computing
    - ↳ new ways?

Related problem:
Primality Testing

Fact:

In $\mathbb{Z}_p$, there is no value x (other than 1 + $-1 = (p-1)$) with $x^2 \equiv 1 \pmod{p}$. $^{(=p-1)}$

Fact 2:

p prime $\Rightarrow$ $p-1$ even, so

$p-1 = 2^s \cdot d$ for some $s, d > 0$

Remember this s!

Thm: If p is prime

For every $a \in \mathbb{Z}_p$, either:

(a) $a^d \equiv 1 \mod p$

(b) $a^{2^r \cdot d} \equiv -1 \mod p$

for some $0 \le r \le s-1$

Why?

We saw:

$$a^{p-1} \equiv 1 \mod p$$

(since $\Phi(p) = p-1$)

So take square root of a^{p-1}:

must get $= 1$ or -1

If -1: (b) holds

If never get -1, remove powers of 2 $\Rightarrow$ (a) holds

So: Contrapositive:

If $\exists$ a such that

(a) $a^d \not\equiv 1 \bmod p$

and

(b) $a^{2^r d} \not\equiv -1 \bmod p$

for all $0 \leq r \leq s-1$,

then p is not prime.

Such an a is a witness (Miller-Rabin primality testing)

How to find?

Guess an a, + check.