

CSci 3100



Recap

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

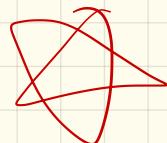
Next week:

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 $p \neq q$,
(5) is hard.

So : RSA (finally!)

Bob: Selects 2 large primes $p \neq q$

• Let $n = pq$

$$\hookrightarrow \varphi(n) = (p-1)(q-1)$$

• Select $e \neq d$ s.t.

- e and $\varphi(n)$ are relatively prime

- $ed \equiv 1 \pmod{\varphi(n)}$

\hookrightarrow Extended Euc Alg

Now:

- (e, n) is public key
- d is private key
(also $p \neq q$)

Encrypting : 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

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

Bob calculate:

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

Claim:

Why?

$$\begin{aligned} C^d \bmod n &= (M^e)^d \bmod n \\ &= M^{ed} \bmod n \end{aligned}$$

$$\text{Know } ed = 1 \bmod \Phi(n)$$

$$M^{ed} = M^{(\frac{\Phi(n)}{2} + 1)} \bmod n$$

$$\begin{aligned} &= (M^{\frac{\Phi(n)}{2}})^k \cdot M^1 \bmod n \\ &= \underbrace{(M^{\frac{\Phi(n)}{2}})^k}_{\text{Euler theorem}} \cdot M^1 \bmod n = M \end{aligned}$$

Example

Key Pair

Public key: $n = 55, e = 3$

Private key: $n = 55, d = 7$

Key Pair Generation

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 $m = c^7 \bmod n$			
	$m^2 \bmod n$	$m^3 \bmod n$	$c^2 \bmod n$	$c^3 \bmod n$	$c^6 \bmod n$	$c^7 \bmod n$
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?

- o Bob needed $\Phi(n)$, since d is e 's inverse mod n .

- o Attacker knows n (but not $\Phi(n)$).

How to find $\Phi(n)$?

So:

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

$(p^i)(q^{-1}) \rightarrow$ these would give d

Bad news: Factoring is NOT NP-Hard.

Best algorithm:

Number field sieve:

$$\mathcal{O}(e^{(\frac{44}{d} \log n)^{1/3}} (\log \log n)^{2/3})$$

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 $n = p q$,
 M will be relatively prime to p or q .
- Can also be used for Digital signing.

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 $x^2 \equiv 1 \pmod{p}$.

Fact 2:

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

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

→ Remember this s !

Then: If p is prime

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

(a) $a^{d-1} \equiv 1 \pmod{p}$

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

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

Why?

We saw:

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

(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) $a^d \not\equiv 1 \pmod{p}$

and

(b) $a^{2^r} \not\equiv 1 \pmod{p}$

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.