CIS 3362 10/28/26

1. Factoring
   - (a) Fermat Factoring
   - (b) ~~R~~ Pollard-Rho
2. Diffie-Hellman Key Exchange

~~Fern~~ Fermat Factoring

Usually goal is to factor $n = p \times q$,
$p, q \in$ Prime $p, q \in$ Odd.

$13 \times 17 = (15-2) \times (15+2)$ $\quad (x-y)(x+y)$ $\quad x^2-y^2$

15 is the midpoint; 13 and 17 are the gap, even

$n = (x-y)(x+y)$

$n = x^2 - y^2$

$$\boxed{y^2 = x^2 - n}$$

x	x^2-n
15	$4 = 2^2$ ✓

$2^2 = 15^2 - 221$

$221 = (15-2)(15+2)$

13×17

$n = 221$

1. Find smallest int $> \sqrt{n}$. (ie $\lceil \sqrt{n} \rceil$) since n isn't a perfect sq.
2. try each possible value of x, starting at this number until ~~x^2~~ is a perfect sq (x^2-n)

$n = 4559 \qquad \lceil \sqrt{n} \rceil = 68$

x	$x^2 - n$	
68	$68^2 - 4559 = 65$	$\times$
69	$69^2 - 4559 = 202$	$\times$ $\quad 2\times 69+1 = 139$
70	$70^2 - 4559 = 341$	$\times$ $\quad 2\times 70+1 = 141$
71	$71^2 - 4559 = 482$	$\times$ $\quad 2\times 71+1 = 143$
72	$72^2 - 4559 = 625 = 25^2$	

$$25^2 = 72^2 - 4559$$
$$4559 = 72^2 - 25^2$$
$$= (72-25)(72+25)$$
$$= 47 \times 97$$

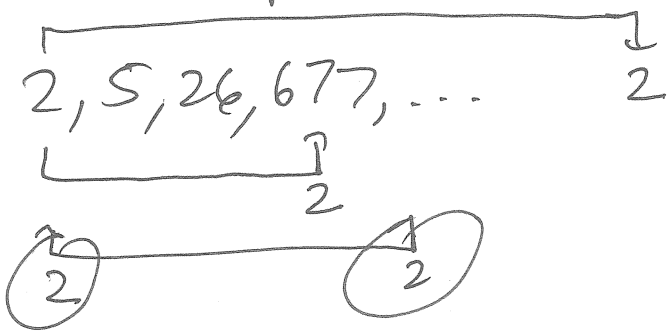
$(x+1)^2 - x^2 =$
$x^2 + 2x + 1 - x^2 =$
$2x+1$

Always works but worst case runs in $O(\sqrt{n})$ time.

Faster if #s close together.

Pollard-Rho

Generate a sequence under mod n
we're guaranteed it will eventually repeat!
If $n = p \times q$, then we can also look at cycles of sequence mod p and mod q } must cycle in at most p or q times p is much smaller than n!



$2 - 2 \equiv 0 \mod q$ $\quad q \mid (a_x - a_1)$

$\gcd(n, a_x - a_1) = q$ $\quad n \nmid (a_x - a_1)$

Instead it computes

$a_2 - a_1$ cycle = 1
$a_4 - a_2$ cycle = 2
$a_6 - a_3$ cycle = 3
$a_8 - a_4$ cycle = 4 etc.
...
$a_1 = 2$

~~$a_{n+1} = (a_n * a_n + 1)$~~

$a_{k+1} = (a_k * a_k + 1) \mod n$

Pollard-Rho

```
a = 2     // a1, a2, a3 ...
b = 2     // a2, a4, a6, ...

while (true) {
    a = (a * a + 1) % n
    b = (b * b + 1) % n
    b = (b * b + 1) % n
    d = gcd(n, abs(a - b))
    if (d > 1 && d < n)
        return d;
    if (d == n)
        return -1; // failed
}
```

Key Distribution

To use DES/AES any symmetric cryptosystem, you need exchange a secret key with the entity you want to communicate with.

Seems IMPOSSIBLE! How can Alice + Bob have a conversation that Eve listens to (entire thing) yet at the end of it, Alice + Bob have exchanged a secret key that Eve cannot figure out.

Whitfield Diffie - did MIT undergrad 1965... freelanced, really interested in cryptography + specifically key exchange.

1974 → visits IBM
most people think impossible!
Alan Kobeim told Diffie to talk to Martin Hellman (prof Stanford)

Diffie meets Hellman on a cold call
Hang out all night talking about crypto
⟹ Diffie enrolls as a grad student @ Stanford ~ Sept 1974

Ralph Merkle also joined
Hellman working by himself at night
Sketched out whole idea in ~1 hr.

Public key Key Exchange

Both Public keys (things everyone knows)
and Private keys (both Alice + Bob will have their own private key)

Alice picks large prime p and a generator g. (To see if $g\#$ is a generator take $p-1$, find all of its ~~its prime divisors~~: $p-1 = 2^k \cdot x$ Then make sure that $g^{2^k} \not\equiv 1 \bmod p$, $g^x \not\equiv 1 \bmod p$, $g^{\frac{p-1}{2}} \not\equiv 1 \bmod p$)

These ARE PUBLIC

Alice picks private key a.

Alice computes $g^a \bmod p$ → Bob

Bob picks private key b

Alice ← Bob computes $g^b \bmod p$

Alice

$(g^b)^a \bmod p$

$\equiv g^{ab} \bmod p$

Bob

$(g^a)^b \bmod p$

$\equiv g^{ab} \bmod p$

Eve knows g^a, g^b but can't figure out ~~a, b quickly~~!

$g^a \times g^b \equiv g^{a+b}$

$(g^a)^{g^b} \not\equiv g^{ab}$

ONE WAY FUNCTION TO KEEP a, b secret

$$g^{ab} \equiv g^{ab \bmod p-1} \bmod p$$