

# RSA and Fermat.

RSA:

**Key Generation:** (Alice)

Primes:  $p, q$ .  $N = pq$ .

Encryption Key:  $e$ , where  $\gcd(e, (p-1)(q-1)) = 1$

Decryption Key:  $d = e^{-1} \pmod{(p-1)(q-1)}$

Message:  $m$

**Encryption** (Bob):  $y = E(m) = m^e \pmod{N}$ .

**Decryption** (Alice):  $D(y) = y^d \pmod{N}$ .

Result:  $m^{ed} \pmod{N}$

Want  $D(E(x)) = x$

**Thm:**  $x^{ed} = x \pmod{N}$

# RSA and Fermat: mathematical connection

**Thm:**  $m^{ed} = m \pmod{pq}$  if  $ed = 1 \pmod{(p-1)(q-1)}$

Seems like magic!

**Fermat's Little Theorem:** For prime  $p$ , and  $a \not\equiv 0 \pmod{p}$ ,

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

$$3^6 \pmod{7} ? 1.$$

$$3^7 \pmod{7} ? 3.$$

$$3^{19} \pmod{7} ? 3^{3*6+1} \pmod{7} ? (3^{3*6} * 3) \pmod{7} ? 3.$$

**Corollary:**  $a^{k(p-1)+1} = a \pmod{p}$

Get  $a$  back when exponent is  $1 \pmod{p-1}$ .

A little like RSA:

$$a^{ed} \pmod{(p-1)(q-1)} \text{ is } a$$

when exponent is  $1 \pmod{(p-1)(q-1)}$ .

**Proof of Corollary.** If  $a = 0$ ,  $a^{k(p-1)+1} = 0 \pmod{m}$ .

Otherwise  $a^{1+k(p-1)} \equiv a^1 * (a^{p-1})^k \equiv a * (1)^k \equiv a \pmod{p}$



Idea: Fermat removes the  $k(p-1)$  from the exponent!

## Correctness of RSA...

**Lemma 1:** For any prime  $p$  and any  $a, b$ ,

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

**Lemma 2:** For any two different primes  $p, q$  and any  $x, k$ ,

$$x^{1+k(p-1)(q-1)} \equiv x \pmod{pq}$$

Let  $a = x$ ,  $b = k(p-1)$  and apply Lemma 1 with modulus  $q$ .

$$x^{1+k(p-1)(q-1)} \equiv x \pmod{q}$$

$$x^{1+k(q-1)(p-1)} - x \equiv 0 \pmod{q} \implies \text{multiple of } q.$$

Let  $a = x$ ,  $b = k(q-1)$  and apply Lemma 1 with modulus  $p$ .

$$x^{1+k(p-1)(q-1)} \equiv x \pmod{p}$$

$$x^{1+k(q-1)(p-1)} - x \equiv 0 \pmod{p} \implies \text{multiple of } p.$$

$$x^{1+k(q-1)(p-1)} - x \text{ is multiple of } p \text{ and } q.$$

$$x^{1+k(q-1)(p-1)} - x \equiv 0 \pmod{pq} \implies x^{1+k(q-1)(p-1)} = x \pmod{pq}.$$



# RSA decodes correctly..

**Lemma 2:** For any two different primes  $p, q$  and any  $x, k$ ,  
 $x^{1+k(p-1)(q-1)} \equiv x \pmod{pq}$

**Theorem:** RSA correctly decodes!

Recall

$$D(E(x)) = (x^e)^d = x^{ed} \equiv x \pmod{pq},$$

where  $ed \equiv 1 \pmod{(p-1)(q-1)} \implies ed = 1 + k(p-1)(q-1)$

$$x^{ed} \equiv x^{k(p-1)(q-1)+1} \equiv x \pmod{pq}.$$



## Key Generation...

1. Find large (100 digit) primes  $p$  and  $q$ ?

**Prime Number Theorem:**  $\pi(N)$  denotes the number of primes less than or equal to  $N$ . For all  $N \geq 17$

$$\pi(N) \geq N/\ln N.$$

Choosing randomly gives approximately  $1/(\ln N)$  chance of number being a prime. (How do you tell if it is prime? ... cs170..Miller-Rabin test.. Primes in  $P$ ).

2. Choose  $e$  with  $\gcd(e, (p-1)(q-1)) = 1$ .  
Use gcd algorithm to test.
3. Find inverse  $d$  of  $e$  modulo  $(p-1)(q-1)$ .  
Use extended gcd algorithm.

All steps are polynomial in  $O(\log N)$ , the number of bits.

# Security of RSA.

## Security?

1. Alice knows  $p$  and  $q$  (and  $d$ , and other numbers).
2. Bob only knows,  $N(= pq)$ , and  $e$ .  
Does not know, for example,  $d$  or factorization of  $N$ .
3. Breaking this scheme  $\implies$  factoring  $N$ .  
Don't know how to factor  $N$  efficiently on regular computers.

## Much more to it in practice!

If Bob sends a message (Credit Card Number) to Alice,  
Eve sees it. (The encrypted CC number.)

**Eve can send same credit card number again!!**  
“Replay attack”

The protocols are built on RSA but more complicated;  
For example, several rounds of challenge/response.

One trick:

Bob encodes credit card number,  $c$ ,  
concatenated with random  $k$ -bit number  $r$ .

Never sends just  $c$ .

Again, more work to do to get entire system.

CS161...