Modular arithmetic:

for integers x, N where $N \geq 1$,

$x \bmod N =$ remainder r when $\frac{x}{N}$

$= r$ where $x = qN + r$
$0 \leq r < N$.

Example: mod 3, there are 3 equivalence classes:

$\ldots, -9, -6, -3, 0, 3, 6, 9, \ldots$

$\ldots, -8, -5, -2, 1, 4, 7, 10, \ldots$

$\ldots, -7, -4, -1, 2, 5, 8, 11, \ldots$

Denote as: $-6 \equiv 9 \bmod 3$

Basic facts: if $x \equiv x' \bmod N$ & $y \equiv y' \bmod N$

then $x + y \equiv x' + y' \bmod N$ & $xy \equiv x'y' \bmod N$.

Example: Since $32 \equiv 1 \bmod 31$,

$2^{345} \equiv (2^5)^{69} \equiv 32^{69} \equiv 1 \bmod 31$

How do we compute $x \bmod N$?

compute $\frac{x}{N}$ & take the remainder

Working with huge numbers,

n-bit numbers x, y, N

what's time as a function of n?

(think of $n \approx 1000$ or $n \approx 2000$)

Adding n-bit numbers x & y takes $O(n)$ time.

Multiplying takes $O(n^2)$ time $\left(\text{can do faster}\right)$

Dividing takes $O(n^2)$ time.

What about modular exponentiation?

computing $x^y \bmod N$

Brute-force: $\underbrace{(x)(x)\cdots(x)}_{y \text{ multiplications}} \bmod N$

Since $y \leq 2^n$ this takes $O(n^2 2^n)$ time.

Better: repeated squaring

Example: for $7^{33} \bmod 19$

use that $33 = (100001)_2$

$7^{33} \equiv 7^{32} \times 7^1 \bmod 19$

compute 7^{2^i} for $i = 0 \to 5$

$\boxed{O(n^3) \text{ time}}$

$7 \equiv 7^{2^0} \bmod 19$

$7^2 \equiv 49 \equiv 11 \bmod 19$

$7^4 = 11^2 \equiv 7 \bmod 19$

$7^8 \equiv 11 \bmod 19$, $7^{16} \equiv 7 \bmod 19$

$7^{32} \equiv 11 \bmod 19$, $7^{33} \equiv 77 \equiv 1 \bmod 19$

Multiplicative inverses?

In real arithmetic, $a \times \frac{1}{a} = 1$

Is there $\frac{1}{a} \bmod N$?

Say x is the multiplicative inverse of $a \bmod N$

if $ax \equiv 1 \bmod N$

Can be at most 1 such x (mod N)

Denote as $x \equiv a^{-1} \bmod N$

Example: $N = 14$

$a = 1$, $a^{-1} \equiv 1 \bmod N$

$a = 2$, $a^{-1} \bmod N$ does not exist

$a = 3$, $a^{-1} \equiv 5 \bmod N$

$a = 4$, $a^{-1} \bmod N$ does not exist

$a = 5$, $a^{-1} \equiv 3 \bmod N$

$a = 9$, $9^{-1} \equiv 11 \bmod N$

$a = 13$, $13^{-1} \equiv 13 \bmod N$

$6^{-1}, 7^{-1}, 8^{-1}, 10^{-1}, 12^{-1} \bmod 14$ does not exist.

When does the inverse exist?

Theorem: $a^{-1} \bmod N$ exists iff $\underbrace{\gcd(a,N)=1}_{\text{Say } a \,\&\, N \text{ are relatively prime}}$

How to compute $a^{-1} \bmod N$ when $\gcd(a,N)=1$?

First, how to check if $\gcd(a,N)=1$?

Euclid's algorithm computes $\gcd(a,N)$

using the fact: $\gcd(a,N) = \gcd(N, a \bmod N)$

for $a \geq N$

takes time $O(n^3)$.

Extended-Euclid computes integers α, β, d where

$d = \gcd(a,N)$ & $a\alpha + N\beta = d$

thus if $d=1$ then $a\alpha \equiv d \equiv 1 \bmod N$

So $\alpha \equiv a^{-1} \bmod N$.

takes time $O(n^3)$.

Fermat's little theorem:

if p is prime, then for every a where $1 \le a < p$,

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

$\Rightarrow$ This will be the basis for testing if a number is prime & for the RSA cryptosystem.

Proof: Fix p & a.

Let $S = \{1, 2, 3, \ldots, p-1\}$

Look at $S' = aS \bmod p =$
$= \{a \bmod p, 2a \bmod p, \ldots, (p-1)a \bmod p\}$

Example: $p = 7$, $a = 3$,
$S = \{1, 2, 3, 4, 5, 6\}$, $S' = \{3, 6, 2, 5, 1, 4\}$
Note: $S = S'$ (just dif't. order)

Claim: in general, for prime p, $S = S'$

Proof: we'll prove that the elements of S' are distinct and non-zero.
Hence S' has $p-1$ elements so they match S.
Suppose they are not distinct: $ai \equiv aj \bmod p$ for $i \ne j$
p is prime so $\gcd(a,p) = 1$ & $a^{-1} \bmod p$ exists.
Then, $aia^{-1} \equiv aja^{-1} \bmod p$
$i \equiv j \bmod p$ $\Rightarrow\!\Leftarrow$

Similarly, if $ai \equiv 0 \bmod p$ then $i \equiv 0 \bmod p$.

Multiply the elements of S:

$$(1)(2)(3)\cdots(p-1) \bmod p = (p-1)! \bmod p$$

Multiply the elements of S':

$$(a)(2a)(3a)\cdots((p-1)(a)) \equiv a^{p-1}(p-1)! \bmod p$$

Since $S = S'$, these 2 are the same:

$$(p-1)! \equiv a^{p-1}(p-1)! \bmod p$$

Since p is prime, every $i \in \{1, 2, \ldots, p-1\}$ has $i^{-1} \bmod p$ exists.

So $(p-1)!^{-1}$ exists

Multiply both sides by it and we get:

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

$\boxtimes$

Euler's theorem: for any N, a, if $\gcd(a,N)=1$

then $a^{\phi(N)} \equiv 1 \bmod N$

where $\phi(N) = \#$ integers $b \in \{1, 2, \ldots, N\}$
where $\gcd(b,N)=1$

Suppose $N=p$ for prime p then $\phi(p)=p-1$.

We'll apply the theorem for $N=pq$
where p & q are primes.

In this case $\phi(N)=(p-1)(q-1)$

Hence for $N=pq$, for a relatively prime to N,

$$a^{(p-1)(q-1)} \equiv 1 \bmod N$$

Application: Let $N=p$.

Take b, c where $c \equiv b^{-1} \bmod p-1$
so $bc \equiv 1 \bmod p-1$

then $bc = 1 + k(p-1)$ for integer k.

Therefore, $a^{bc} \equiv (a)(a^{p-1})^k \equiv a \bmod p$.

Take $N = pq$ for primes p, q.

take d, e where $e \equiv d^{-1} \bmod (p-1)(q-1)$

$$\text{So } de \equiv 1 \bmod (p-1)(q-1)$$

$$\text{and } de = 1 + k(p-1)(q-1).$$

for a relatively prime to N,

$$a^{de} \equiv (a)\left(a^{(p-1)(q-1)}\right)^k \equiv a \bmod N$$

Aside: What about for other a?

Still have that $a^{de} \equiv a \bmod N$

Why? Suppose $\gcd(a, N) > 1$

then a is a multiple of p &/or q.

So either $a \equiv 0 \bmod p$ &/or $a \equiv 0 \bmod q$.

Simple version of Chinese remainder theorem:

for primes p & q if $x \equiv y \bmod p$ & $x \equiv y \bmod q$

then $x \equiv y \bmod pq$

Say $a \equiv 0 \bmod p$.

Then $a^{ed} \equiv 0 \bmod p$

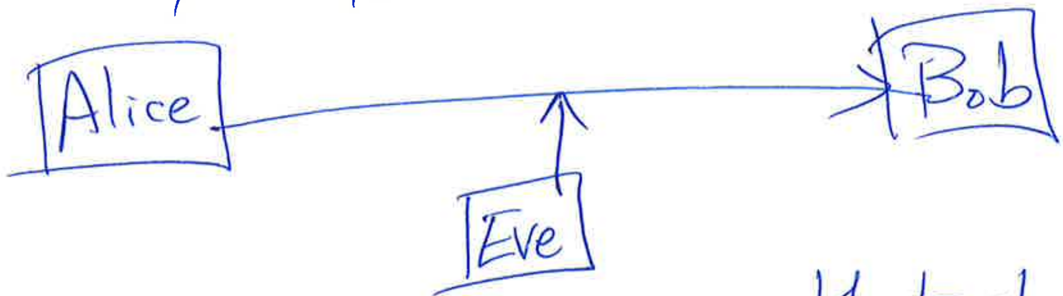
and $a^{ed} \equiv (a)\left(a^{q-1}\right)^{k(p-1)} \equiv a \bmod q$

Thus, $a^{ed} \equiv a \bmod p$ & $a^{ed} \equiv a \bmod q$

So $a^{ed} \equiv a \bmod pq$.

RSA = [Rivest - Shamir - Adelman '77]

Public-key cryptosystem:



Alice has a message m that she wants to send to Bob but Eve can see whatever message is sent.

Bob publishes a public key (N, e)

Alice uses Bob's public key to send an encrypted version of m to Bob

Bob uses his private secret key to decrypt.

<u>Protocol:</u>

Bob: 1) Picks 2 n-bit random primes p & q.

HOW? We'll see next class.

2) Bob chooses an e relatively prime to $(p-1)(q-1)$.

by trying $e = 3, 5, 7, 11, \ldots$

Let $N = pq$.

3) Bob publishes his public key (N, e)

4) Bob computes his private key:

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

<u>Alice</u>: for message m,

1) Looks up Bob's public key (N, e).

2) She computes $y \equiv m^e \bmod N$

3) She sends y.

<u>Bob</u>: 1) He receives y

2) He decrypts using:

$$m \equiv y^d \bmod N.$$

Since $de \equiv 1 \bmod (p-1)(q-1)$.