Primality

Input size = $\log n = N$

Running time = $T_n = 2^{\frac{\log n}{2}} = 2^{\frac{N}{2}}$.

Fermat's little theorem: ~ gcd(a,p) = 1 and a < p.

If p is a prime and (a is not divisible by p) then,

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

Proof: By induction on a. Base case of $a = 0$ is trivial.

Note that $(x+y)^p \equiv x^p + y^p \bmod p \ \forall\ x, y$ not div. by p.

$$\Rightarrow (k+1)^p \equiv k^p + 1^p \bmod p$$

$$\equiv k^p + 1 \bmod p$$

$$\equiv k + 1 \bmod p \quad // \ k^p \equiv k \bmod p \text{ by induction hypothesis.}$$

$$(x+y)^p = \sum_{i=0}^{p} \binom{p}{i} \cdot x^i \cdot y^{p-i}$$

$\binom{p}{i} \equiv 0 \bmod p \ \forall\ i \in [1, p-1]$

Thus $(k+1)^{p-1} \equiv 1 \bmod p$.

For all $a \in \{1, 2, \ldots, n-1\}$ $a^{n-1} \equiv 1 \bmod n$ if n is prime.

Remark: If $a^{n-1} \equiv 1 \bmod n$ but n is not prime then n is called "pseudoprime to base a". Ex: $n = 341, a = 2$.

If $\exists\ b$ s.t $b^{n-1} \not\equiv 1 \bmod n$ then n is composite.

$2^{340} \equiv 1 \bmod 341$

Carmichael numbers:

A Carmichael number is a composite number n s.t $\forall$ a co-prime with n and $a < n$,

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

Ex: 561, 1729.

If $a^{n-1} \not\equiv 1 \bmod n$ then a is called Fermat's

Fermat's primality test: FermatTest(n)

witness.

Fermat's lie: if n is composite $a^{n-1} \equiv 1 \pmod n$.

1. Pick a uniformly at random from $\{1, 2, \ldots, n-1\}$.
2. If $a^{n-1} \not\equiv 1 \bmod n$ return composite.
3. Else, return "~~probably~~ may be a prime".

Lemma: If n is a composite number s.t there exists an element a s.t n does not divide a, and $a^{n-1} \not\equiv 1 \pmod n$ Then there is a set $S \subseteq \{1, 2, \ldots, n-1\}$ s.t

- $b^{n-1} \not\equiv 1 \pmod n$ for all $b \in S$, and.
- $|S| \geq \frac{n-1}{2}$.

$\{a_1, \ldots, a_{\frac{n-1}{2}}\}$ s.t $a_i^{n-1} \not\equiv 1 \bmod n$.

Assuming the lemma, we get that FermatTest(n) succeeds with a prob of at least $\frac{1}{2}$.

Miller-Rabin primality test: ↗ Solovay-Strassen primality test

1. Compute s, d s.t $n-1 = 2^s d$, and d is odd.
2. Pick a from $\{1, 2, \ldots, n-1\}$ uniformly at random.
3. Compute

   $(a^d \bmod n)$, $(a^{2d} \bmod n)$, $(a^{2^2 d} \bmod n)$, …, $(a^{2^{s-1} d} \bmod n)$, $(a^{2^i d} \bmod n)$.
4. If any of these numbers is $n-1 \bmod n$ then conclude that n is prime. Else not a prime.

$(x+a)^n = x^n + a^n \pmod n$

Thm: If n is a prime then algo always concludes that n is prime. If n is composite, algo outputs that n is prime with prob at most $\frac{1}{4}$.

Sketch:

$a^{n-1} = 1 \mod n$

↳ $a^{2^s d} \equiv 1 \mod n$.

1. If n is an odd prime, only square roots of 1 are $+1$ and $-1 \mod n$.

$n = pq$

$\begin{cases} n \mid x+1 \Rightarrow x \equiv -1 \mod n \\ n \mid x-1 \quad\quad x \equiv 1 \mod n \end{cases}$

2. $\exists$ a term in the seq $a^{2^{s-1}d}, \dots, a^d$ s.t $a^{s^t d} \equiv -1 \mod n$ if n is an odd prime.

$x^2 - 1 \equiv 0 \mod n \Leftarrow x^2 \equiv 1 \mod n$.

$(x+1)(x-1) \equiv 0 \mod n$

$n \mid x^2 - 1 \Rightarrow n \mid (x+1)(x-1)$

RSA encryption

Find 3 large integers e, d and n s.t for all integers $m \leq n$,

$(m^e)^d \equiv m \pmod n$ and $(m^d)^e \equiv m \pmod n$.

Key generation:

1. Choose two distinct primes p and q.
2. Compute $n = pq$.
3. Compute $\lambda(n) = \text{lcm}(p-1, q-1)$.
4. Choose an integer e s.t $\gcd(e, \lambda(n)) = 1$. // Public key (n, e)
5. Compute d s.t $de = 1 \pmod{\lambda(n)}$ // Private key (d).

Alice Bob

Public key
(n, e)

Converts message M into an integer m.

$C \equiv m^e \pmod n$

Recovers the message by computing c^d
$= (m^e)^d \bmod n$?
$\equiv m \pmod n$.

Claim: e, d, n generated above satisfy the eqn

$m^{ed-1} \equiv 0 \bmod p$
$\equiv 0 \bmod q$

$p \mid m^{ed-1}$, $q \mid m^{ed-1}$

$(m^e)^d \equiv m \bmod n$ ✓
$(m^d)^e \equiv m \bmod n$

Correctness:

We want $(m^e)^d \equiv m \pmod n$. $\rightarrow$ $(m^e)^d \equiv m \bmod p$ or $\equiv m \bmod q$

Note that Bob does not know the values of p and q.

Recall that $\lambda(n) = \text{lcm}(p-1, q-1)$ and $ed \equiv 1 \bmod \lambda(n)$.

$\Rightarrow (ed-1) \equiv 0 \bmod \lambda(n)$

$= h(p-1) = k(q-1)$.

Say m is a multiple of p then

$(m^e)^d \equiv 0 \bmod p = m \bmod p$

Else, $(m^e)^d = m^{h(p-1)} \cdot m = (m^{p-1})^h \cdot m \equiv m \pmod p$

Similarly, if m is a multiple of q,

$(m^e)^d \equiv 0 \mod q = m \mod q$.

Else

$$m^{ed} = m^{ed-1} \cdot m = m^{k(q-1)} \cdot m = (m^{q-1})^k \cdot m$$
$$\equiv m \pmod q.$$

Since $m^{ed} \equiv m \mod p$ and $m^{ed} \equiv m \mod q$ we get that $m^{ed} \equiv m \pmod n$. ← p.q

$p \mid m^{ed-1}$

$q \mid m^{ed-1}$

$\Rightarrow pq \mid m^{ed-1}$.

Signing messages:

Let's fix a standard hash function H.

Compute $h = H(m)$.

Bob uses his private key to sign. → $h^{d_{Bob}}$.

Bob appends $h^{d_{Bob}}$ to $c = m^{e_{Alice}} \pmod n$.

$(c, h^{d_{Bob}})$.

Private key d_{Alice} d_{Bob} n

Public keys e_{Alice} e_{Bob}

Alice gets both c and $h^{d_{Bob}}$.

Does $c^{d_{Alice}}$ to obtain $m \pmod n$

Does $(h^{d_{Bob}})^{e_{Bob}}$ to get $h \bmod n$.

Check if $h \stackrel{?}{=} H(m)$.

Fingerprinting

Dropbox has a file x and user has a file y, both of size at most l.

Qn: Are x and y the same files?

1. Convert x and y to bit strings.
2. Dropbox picks uniformly a prime p in $[1, l^2]$.
3. Dropbox sends the user p and $x \bmod p$.
4. Check if $y \equiv x \bmod p$.

Lemma: Probability that the algorithm makes an error is at most $\frac{3 \ln l}{l}$.

Proof: If $x = y$ then $x \equiv y \bmod p$ trivially. Otherwise $x \neq y$. Then the algo makes an error only if $x - y$ is zero mod p. That is, p divides $|x-y|$.

Note that $|x-y| \leq 2 \cdot 2^l$ and if $|x-y|$ has k prime factors $|x-y| \geq 2^k$.

$\Rightarrow \quad k \leq l+1$.

Prime number theorem.

No. of prime factors between 1 and $l^2 \approx \frac{l^2}{2 \ln l}$.

$\Rightarrow$ $x \equiv y \bmod p$ if p is picked such that p divides $|x-y|$.

Prob of picking such a $p \leq \dfrac{\ell+1}{\ell^2/2\ln\ell} \leq \dfrac{3\ln\ell}{\ell}$.

For example: $|x| = |y| \approx 1$ Gigabyte $\Rightarrow$ $\ell \approx \log(8\times 10^9)$

$= 9\log 10 + 3$

$\leq 9 \times 3.32 + 3$

≤ 35.

We exchange at most $2\log p$ bits $\leq 4\log\ell$ bits.

$\leq$ 20 bits.