Lecture 12

Thursday, 13 October 2022 10:45

RSA signatures

Keygen : the same as in RSA encryption (PKE)

pick primes $p \neq q$ of $\frac{\ell}{2}$ bits $\Rightarrow$ n has ℓ bits

$n = p \cdot q$

$\varphi(n)$

pick e with low Hamming Weight ↗ number of 1s (ones) in binary representation of e

compute $d \equiv e^{-1} \mod \varphi(n)$

Sign: $s \equiv (h(m))^d \mod n$

Verify: $s^e \equiv h(m) \mod n$

for hash function h; h maps to ℓ bits (with fixed padding). This removes the homomorphic property.

General principle: keep only things you need; here, keygen outputs (n, e) & (n, d); forgets $p, q, \varphi(n)$

But GPG keeps p, q and some u, because we want faster private operations.

We cannot choose a special d for security, but we can save some effort.

CRT: $(h(m))^{d_p} \equiv s_p \mod p$

$(h(m))^{d_q} \equiv s_q \mod q$ ← both have $\frac{\ell}{2}$ bits

then s is CRT of s_p & s_q

$s \equiv s_p + p \cdot u (s_q - s_p) \mod pq$, ($p \cdot q = n$)

where $u \equiv p^{-1} \mod q$

This is the u in GPG

We gain a factor of 3-4 (3 to 4) from working on size $\frac{\ell}{2}$ operands, but need 2 of these.

integers mod p have order $p-1$, hence $d_p \equiv d \mod p-1$ ← note the -1!

$d_q \equiv d \mod q-1$

d has bitlength ℓ, d_p & d_q have bitlength $\frac{\ell}{2}$ $\Rightarrow$ each square-and-multiply takes half the time (is 'the same' as double-and-add, but in multiplicative group)

Overall saving is factor of 3-4 (3 to 4). CRT is fast

How to pick p and q?

Pick a candidate p, test for primality, repeat on failure

Primality test: returns "composite" or "probably prime"

Primality proof: returns "prime" or "probably composite" more expensive

know probability, iterate to improve

Fermat's primality test

Fermat's little theorem says: for any a with $\gcd(a,p) = 1$ we have $a^{p-1} \equiv 1 \mod p$ if p is prime

Test:

1. pick $1 < a < p-1$
2. if $\gcd(a,p) \neq 1$ → output "composite"
3. if $a^{p-1} \equiv 1 \mod p$ → output "probably prime"
4. else, output "composite"

Next round, pick different a.

Some exceptions; Carmichael numbers are caught only at gcd test.

For all other composites, it fails quickly.

Smallest Carmichael number: $n = 561 = 3 \cdot 11 \cdot 17$

Miller-Rabin test does not have exceptions

if p is prime, then $x^2 \equiv 1 \mod p$ has two solutions

$x_{1,2} = \pm 1$

for $n = p \cdot q$, we have

$x \equiv \pm 1 \mod p$
$x \equiv \pm 1 \mod q$

These 4 choices of x (signs taken independently) all satisfy $x^2 \equiv 1 \mod p \cdot q$

The ones with opposite signs give new solutions $x \notin \{\pm 1\}$. For k factors, there are 2^k solutions.

If $x^2 \equiv 1 \mod p$ always returns ± 1, then p is prime - but we cannot solve equations modulo composites

→ If we could, we'd get a factorization algorithm

Get answers of $x^2 \equiv 1 \mod n$, if we get $x \neq \pm$, then $\gcd(x+1, n)$ is a factor of n in $(1, n)$ ($1 < \gcd(x+1, n) < n$)

This gcd is the product of the primes where $x \equiv -1 \mod p_i$

$\gcd(x+1, n) = \prod p_i$ ← p_i with $x \equiv -1 \mod p_i$

Can pick random a, compute $a^2 \mod n \equiv b$. Then ask for solution to $x^2 \equiv b \mod n$ (pick a large enough so that b is not an integer square)

50% chance that answer $x \not\equiv a \mod n$

Computing square roots mod n is as hard as factoring n.

Miller-Rabin:

1. pick $1 < a < p-1$
2. write $p-1 = 2^s \cdot t$ with t odd
3. compute $b \equiv a^t \mod p$
4. if $b \equiv \pm 1$ → output "probably prime"

   else a) for $i = 1$ to $s-1$

   $b \leftarrow b^2 \mod p$

   if $b = -1$ → output "probably prime"

   if $b = 1$ → output "composite" → we got here by squaring a number $b \not\equiv \pm 1$, so p is not prime

   b) output "composite" → we have computed $a^{\frac{p-1}{2}}$ so p fails Fermat or has a root unequal to ± 1.

Miller-Rabin finds composites in exponentiations reaching +1 or by $a^{p-1} \not\equiv 1 \mod p$ (Fermat's little theorem)

→ at least 50% chance to detect compositeness. Repeat to improve chances.

Look up Pocklington primality proof (last slide in rsa-3.pdf)

Factorization methods

congruence of squares (above) factors numbers.

Dixon's method
quadratic sieve
number field sieve

} → build up quadratic equations

These all need lots of factorizations of auxiliary numbers which are easier to factor

(trial division, Pollard's rho for factoring, $p-1$ method & generalizations $p+1$ method & ECM (elliptic curve method))

p-1 method

pick an s with many small factors, $s = \text{lcm}(2, 3, \ldots, B)$

Repeat

pick $1 < a < n-1$

compute $b \equiv a^s \mod n$ ← we know n, this ensures b & all intermediate results are $< n$

compute $\gcd(b-1, n) = d$

output d if not 1 or n

We know if $(p-1) | s$ then for all $1 \leq a \leq p-1$, we have

$a^s \equiv 1 \mod p$,

so p divides d for p a divisor of n.

But, we do not need $(p-1) | s$, only need that order of $a \mod p$ divides s.

Need some other prime q with $a^s \not\equiv 1 \mod q$ to split p & q.