NUMBER THEORY

So far :

MINICRYPT

$$\begin{array}{l} \text{OWF} \Rightarrow \text{PRG} \Rightarrow \text{PRF} \Rightarrow \text{SKE} \quad \begin{array}{l}\text{CPA}\\ \text{CCA}\end{array} \\ \Downarrow \qquad\qquad\qquad\qquad \Rightarrow \text{PRP} \\ \text{DIGITAL SIGNATURES} \end{array}$$

We now move CRYPTOMANIA.

CRYPTOMANIA

PKE

(DIGITAL SIG)

CRIT

We will cover the known constructions used in practice: RSA, ElGamal, ~~FDH~~ signatures, Fiat-Shamir signatures ...

Hardness:

- Number theory (FACTORING, DISCRETE LOG, ELLIPTIC CURVES, ...).
- Lattices (LWE, SIS).

Modular arithmetic over $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$

Look: $(\mathbb{Z}_n, +)$ is a group. Importantly, $\exists$ an inverse: $\forall a \in \mathbb{Z}_n, \exists b \in \mathbb{Z}_n$ s.t.

$$a + b = 0 \bmod n.$$

Look at "$\cdot$" instead: $(\mathbb{Z}_n, \cdot)$, it's not a group for every n.

LEMMA If $\gcd(a, n) > 1$, then a is not invertible mod n.

Proof. Assume not: a is invertible, so $\exists b \in \mathbb{Z}_n$ s.t. $a \cdot b \equiv 1 \bmod n$. But then:

$$ab = 1 + qn \quad \text{for } q > 0$$

Then $\gcd(a, n)$ divides $ab - qn$, and thus it divides 1, so $\gcd(a, b) = 1 \Rightarrow \leftarrow$

$$\mathbb{Z}_n^* = \{ a \in \mathbb{Z}_n : a \text{ invertible mod } n \}.$$
$$(\gcd(a, n) = 1.) \quad ?$$

$$\# \mathbb{Z}_n^* = \varphi(n) \quad ; \quad \text{e.g.} \quad n = p \cdot q$$
$$\varphi(n) = (p-1) \cdot (q-1).$$

$$\mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{0\} = \{1, \dots, p-1\}.$$
$(\mathbb{Z}_p^*, \cdot)$ is a group.

We want efficient algorithms for computing say over $(\mathbb{Z}_n^*, +, \cdot)$: $|n| = 2048$ bits.

- Addition and multiplication are trivial.
- Inverse? Exponentiation $a^b \bmod n$?

Inverse. Extended Euclidean Algorithm.

LEMMA Let a, b s.t. $a \geq b > 0$. Then $\gcd(a, b) = \gcd(b, a \bmod b)$.

Proof. Because the common divisors between (a, b) are the same as $(b, a \bmod b)$, since

$$a = qb + a \bmod b \quad ; \quad q = \lfloor a/b \rfloor$$

$\blacksquare$

THM. Given $a \geq b > 0$, we can compute $\gcd(a, b)$ in poly-time. Also, we can find integers u, v, s.t. $au + bv = \gcd(a, b)$

Cor We can compute the inverse: If $\gcd(a,b)=1$

$\Rightarrow \quad au + bv = 1$

$\Rightarrow \quad au \equiv 1 \mod b$

Proof. Use the lemma recursively:

$$a = bq_1 + r_1 \quad ; \quad 0 \leq r_1 < b.$$

By the lemma: $\gcd(a,b) = \gcd(b, r_1)$.

Keep going: $\quad b = r_1 q_2 + r_2 \quad ; \quad 0 \leq r_2 < r_1$

$$\gcd(b, r_1) = \gcd(r_1, r_2)$$

$\ldots \quad r_{t+1} = 0 \Rightarrow \gcd(a,b) = r_t$

It's polynomial time because $r_{i+2} \le r_i/2$ $\forall\, i = 0, 1, \dots, t-2$.

Clearly, $r_{i+1} < r_i$. If $r_{i+1} \le r_i/2$ we are done. So assume $r_i > r_{i+1} > r_i/2$.

But then:

$$
\begin{aligned}
r_{i+2} = r_i \bmod r_{i+1} &= r_i - q_{i+2}\, r_{i+1} \\
&< r_i - r_{i+1} \\
&< r_i - r_i/2 = r_i/2.
\end{aligned}
$$

$\#$ of steps: $\approx 2 \cdot \lambda$ where $\lambda = |b|$

Example: $a = 14$; $b = 10$. Then:

$$14 = 10 \cdot 1 + 4 \quad ; \quad 10 = 2 \cdot 4 + 2 \Rightarrow r_t = 2$$

$$; \quad 4 = 2 \cdot 2 + 0 \Rightarrow r_{t+1} = 0$$

$$\Rightarrow \gcd(14, 10) = 2$$

To get u, v:

$$2 = 10 - 2 \cdot 4 = 10 - 2 \cdot (14 - 10)$$

$$= 3 \cdot 10 + (-2) \cdot 14$$

$$u = -2 \; ; \; v = 3$$

- Exponentiation: Square-and-multiply-

let $b = (b_l, b_{l-1}, \ldots, b_0)$

$$e^b \equiv e^{\sum b_i \cdot 2^i} \mod n$$

$$\equiv \prod e^{b_i \cdot 2^i} \mod n$$

$$\equiv \prod_{i:\, b_i = 1} e^{(2^i)} \mod n$$

$$\equiv e^{b_0} \cdot (e^2)^{b_1} \cdot (e^4)^{b_2} \cdot \ldots \cdot (e^{2^l})^{b_l}$$

(Spoiler: RSA encryption will be smth like

$c \equiv m^e \mod n$.)

Decryption : $c^d \bmod n$.)

Few more things : Prime numbers. How do we generate large primes ?.

Thm There are infinitely many primes and

$$\pi(x) = \text{"}\#\text{ primes} \leq x\text{"} \geq \frac{x}{3\log_2 x} \approx x/\log x.$$

Idea : Pick a random p and test if p is prime.

Thm We can test if p is prime in poly-time.

⇒ We can sample large primes. Sample, Test and if not prime sample again!

↳ λ-bit integer

$$\Pr\left[x \text{ prime} : x \in [2^\lambda - 1] \right] \geq \frac{\dfrac{2^\lambda - 1}{3 \log(2^\lambda - 1)} \quad \text{(\# of primes)}}{2^\lambda - 1 \quad \text{(\# of integers)}} \approx 1/3\lambda$$

$\Pr[$ fail after t steps $] \leq \left(1 - \frac{1}{3\lambda}\right)^t$