RECAP

- Congruent structures: $\mathbb{Z}_n$
  operations on $\mathbb{Z}_n$: $+, \cdot$
- $(\mathbb{Z}_n, +)$ is an additive group
- $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n : \gcd(a,n) = 1\}$
  $(\mathbb{Z}_n^*, \cdot)$ is a multiplicative group
- Computation of a^{-1}, $a \in \mathbb{Z}_n^*$
  via $EE(a,n)$
- Euler's function $\varphi(n) = |\mathbb{Z}_n^*|$
- Euler's theorem $\forall a : \gcd(a,n) = 1$

$$a^{\varphi(n)} \bmod n = 1$$

- Fermat's little theorem

$\forall p$ prime $\forall a \in \mathbb{Z}_p^* \; (= \mathbb{Z}_p^+)$

$$a^{p-1} \bmod n = 1$$

THE CHINESE REMINDER THEOREM

(Sun-Tzu, 300 AD)

(important result: establishes a bijection between congruent structures.

Can be seen as a conversion between different representations of certain congruent structures

APPLICATIONS: Proof of correctness of RSA, fast computations mod n (large), solution of systems of congruences ...

THEOREM Let $n = n_1 \cdot n_2 \cdot \ldots \cdot n_k$, with $k > 1$ and $\gcd(n_i, n_j) = 1$, $1 \leq i \neq j \leq k$.

Let $f: \mathbb{Z}_n \to \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \ldots \times \mathbb{Z}_{n_k}$ defined as $f(a) = (a \bmod n_1, a \bmod n_2, \ldots, a \bmod n_k)$

Then:

1. f is bijective
2. $f((a+b) \bmod n) = ((a+b) \bmod n_1, \ldots, (a+b) \bmod n_k)$
3. $f((ab) \bmod n) ((ab) \bmod n_1, \ldots, (ab) \bmod n_k)$

Function f "converts" reminders mod n into k-tles of reminders $(\bmod n_1, \ldots, \bmod n_k)$ so that arithmetic mod n can be executed

equivalently by doing arithmetic w.r.t. smaller modules!

(COMPUTATIONAL ADVANTAGE! computing mod n is quadratic in $|\langle n \rangle|$!)

Eg: $n = p \cdot q$, with $p, q \approx \sqrt{n}$

$\Rightarrow |\langle p \rangle|, |\langle q \rangle| \approx |\langle n \rangle|/2$

$|\langle p \rangle|^2, |\langle q \rangle|^2 \approx |\langle n \rangle|^2/4$

$\Rightarrow$ Computing $\{x \bmod p, x \bmod q\}$ costs about a half as computing $x \bmod n$

PROOF

Let us first prove:

2. $f((a+b) \bmod n) = ((a+b) \bmod n_1, \ldots, (a+b) \bmod n_k)$
3. $f((ab) \bmod n)((ab) \bmod n_1, \ldots, (ab) \bmod n_k)$

These immediately follow from property m4. For $m, n > 0$:

$$m \mid n \Rightarrow (x \bmod n) \bmod m = x \bmod m$$

since $n_i \mid n$, $1 \le i \le k$, thus

$$[(a+b) \bmod n] \bmod n_i = (a+b) \bmod n_i$$
$$[(ab) \bmod n] \bmod n_i = (ab) \bmod n_i$$

Therefore

$$f((a+b) \bmod n) = ([(a+b) \bmod n] \bmod n_1, \ldots, [(a+b) \bmod n] \bmod n_k)$$
$$= ((a+b) \bmod n_1, \ldots, (a+b) \bmod n_k)$$

Let us prove 1. Since

$|\mathbb{Z}_n| = n = n_1 \cdot n_2 \cdot \ldots \cdot n_k = |\mathbb{Z}_{n_1}| \times |\mathbb{Z}_{n_2}| \times \ldots \times |\mathbb{Z}_{n_k}|$

to prove bijectivity it is sufficient to prove surjectivity (or injectivity).

Given $(a_1, \ldots, a_k) \in \mathbb{Z}_{n_1} \times \ldots \times \mathbb{Z}_{n_k}$ we will determine $x \in \mathbb{Z}_n$ such that

$f(x) = (x \bmod n_1, \ldots, x \bmod n_k) = (a_1, \ldots, a_k)$

REMARK We need to show that the following system of congruences:

$$\begin{cases} x \equiv a_1 \bmod n_1 \\ x \equiv a_2 \bmod n_2 \\ \vdots \\ x \equiv a_k \bmod n_k \end{cases}$$

always admits a (unique) solution $x \in \mathbb{Z}_n$

The proof is constructive: it gives a procedure to compute x!

Define $m_i = \prod_{\substack{j=1 \\ j \neq i}}^{k} n_j \qquad 1 \leq i \leq k$

In other words, $m_i = \frac{n}{n_i} = n_1 \cdot \ldots \cdot n_{i-1} \cdot n_{i+1} \cdot \ldots \cdot n_k$

Clearly $\gcd(m_i, n_i) = 1$ (since $\gcd(a,n) = 1$ and $\gcd(b,n) = 1 \Rightarrow \gcd(ab, n) = 1$)

(use Bezout's id to prove this)

Therefore $[m_i]_{n_i} \in \mathbb{Z}_{n_i}^* \Rightarrow \exists [m_i]_{n_i}^{-1} \in \mathbb{Z}_{n_i}^*$:

$(m_i \cdot m_i^{-1}) \bmod n_i = 1$

Set $c_i = m_i \cdot (m_i^{-1} \bmod n_i)$ ($m_i^{-1} \bmod n_i$: principal representative of $([m_i]_{n_i})^{-1}$)

We set $x = \left(\sum_{i=1}^{k} a_i \cdot c_i\right) \bmod n$

and prove that

$f(x) = (a_1, a_2, \dots, a_k)$

Observe that for $1 \le i \ne j \le k$:

$c_i \bmod n_j = 0$

since $m_i (m_i^{-1} \bmod n_i) = K n_j$ (m_i is a multiple of n_j)

Therefore

$$\left[\left(\sum_{i=1}^{k} a_i \cdot c_i\right) \bmod n\right] \bmod n_j =$$

$$\overset{m_4}{=} \left(\sum_{i=1}^{k} a_i \cdot c_i\right) \bmod n_j$$

($c_i \bmod n_j = 0$ for $i \ne j$)

$$\overset{m_1, m_2}{=} \left(\sum_{i=1}^{k} (a_i \bmod n_j)(c_i \bmod n_j)\right) \bmod n_j$$

$$= \left((a_j \bmod n_j)(c_j \bmod n_j)\right) \bmod n_j$$

$$\overset{m_2}{=} (a_j c_j) \bmod n_j$$

$$= \left(a_j n_j (m_j^{-1} \bmod n_j)\right) \bmod n_j$$

$\overset{m_2}{=} \left(a_j \left((M_j M_j^{-1}) \bmod m_j\right)\right) \bmod m_j$

$= a_j \bmod m_j = a_j.$ Q.E.D.

COROLLARY 1 Let $m_1, \ldots, m_k$: $\gcd(m_i, m_j) = 1$ for $1 \le i \ne j \le k$. Then the system of congruences

$$\begin{cases} x \equiv a_1 \bmod m_1, & a_i \in \mathbb{Z}_{m_1} \\ \vdots \\ x \equiv a_k \bmod m_k, & a_k \in \mathbb{Z}_{m_k} \end{cases}$$

has a unique solution in $\mathbb{Z}_m$:

$$x = f^{-1}(a_1, a_2, \ldots, a_k)$$

COROLLARY 2 Let $m_1, \ldots, m_k$: $\gcd(m_i, m_j) = 1$ for $1 \le i \ne j \le k$, $m = \prod_{i=1}^{k} m_i$. Then

$$(x \equiv y) \bmod m \iff (x \equiv y) \bmod m_i, \quad 1 \le i \le k$$

$\Rightarrow$ If $x \bmod m = y \bmod m$ then $(x \bmod m) \bmod m_i = (y \bmod m) \bmod m_i$ that is, from m4: $x \bmod m_i = y \bmod m_i$ for all $1 \le i \le k$

$\Leftarrow$ If $x \equiv y \bmod m_i$, $1 \le i \le k$, then $f(x) = f(y)$ (since $f(a) = (a \bmod m_1, \ldots, a \bmod m_k)$) Therefore $x \equiv y \bmod m$ because f is injective over $\mathbb{Z}_m$

EXAMPLE

Consider the system

$$\begin{cases} x \equiv 2 \mod 5 \\ x \equiv 3 \mod 13 \end{cases}$$

we determine a solution in $\mathbb{Z}_n$, with $n = 5 \cdot 13 = 65$

$n_1 = 5 \quad n_2 = 13 \quad \gcd(5, 13) = 1$ ok

$m_1 = \frac{n}{n_1} = 13 \qquad m_2 = \frac{n}{n_2} = 5$

$m_1^{-1} \mod 5 = 13^{-1} \mod 5 = 2$, since

$13 \cdot 2 \mod 5 = 26 \mod 5 = 1$

$m_2^{-1} \mod 13 = 5^{-1} \mod 13 = 8$, since

$5 \cdot 8 \mod 13 = 40 \mod 13 = 1$

Therefore $c_1 = 13 \cdot 2 = 26, c_2 = 5 \cdot 8 = 40$

and $x = f^{-1}(2, 3) = (2 \cdot 26 + 3 \cdot 40) \mod 65$

$= (52 + 120) \mod 65 =$

$= 172 \mod 65 = 172 - 130 = \boxed{42}$

THE RSA PUBLIC-KEY CRYPTOSYSTEM

↳ [Rivest, Shamir, Adleman, 1978]

A cryptosystem provides a family of coding functions. A function from the family can be used by a sender to send a message to a recipient along a nonsecure communication channel. An essential property of a cryptosystem is that the encoded message can be decoded efficiently only in the presence of extra information (key)

FORMALLY:

$\mathcal{D}$ = message domain (usually, binary strings of given length representing plain text: e.g. concatenation of ASCII codes) ← (TREAT AS MEMBERS OF $\mathbb{Z}_M$)

Cryptosystem $\mathcal{C} = \{e_K : \mathcal{D} \to \mathcal{D} : K \in \mathcal{K}\}$ family of encodings parametric in $K \in \mathcal{K}$ (key)

PROPERTIES:

- e_K is invertible and efficiently computable, given K.

- $\forall M \in \mathcal{D}$: given $e_K(M) = y$, it is computationally hard to compute $M = e_K^{-1}(y)$ in the essence of any other information (not necessarily K)

Public-Key Cryptosystem

Each participant x has two keys:

- P_x : public key
- S_x : secret key

EXAMPLE

Alice : P_A, S_A

Bob : P_B, S_B

Each participant creates its own keys

- The public key P_x is distributed to all participants
- The secret key S_x is known only to x

Each key corresponds to an encoding function:

EXAMPLE: $M \in \mathcal{D}$

$$e_{P_A}(M),\ e_{S_A}(M),\ e_{P_B}(M),\ e_{S_B}(M)$$

$$\| \|$$

$$P_A(M),\ S_A(M),\ P_B(M),\ S_B(M)$$

PROPERTIES

1. $P_X(M)$, $S_X(M)$ "efficiently computable," given key P_X or S_X
2. $S_X(M) = P_X^{-1}(M)$, that is

$$S_X(P_X(M)) = P_X(S_X(M)) = M$$

3. Even if $P_X(M)$ is known, computing $S_X(M) = P_X^{-1}(M)$ is computationally unfeasible if S_X is not known

Point 3. is critical. For many functions knowing the analytical description of $P_X(M)$ yields $S_X(M) = P_X^{-1}(M)$ easily

EXAMPLE $\mathcal{D} = \mathbb{Z}_n$

$P_A(M) = M + 7 \bmod n$

$\Rightarrow S_A(Y) = Y - 7 \bmod n$

$P_A(M) = aM \bmod n$, $a \in \mathbb{Z}_n^*$

$\Rightarrow S_A(Y) = a^{-1}Y \bmod n$

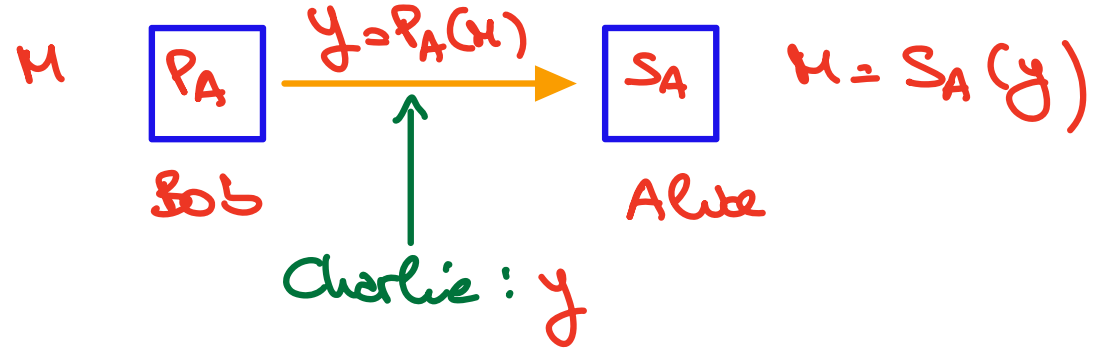
NOTE For long messages we use repeated application on k-tuples, $k \geq 1$:

$P_A(\langle M_1, \ldots, M_k \rangle) = \langle P_A(M_1), \ldots, P(M_k) \rangle$

COMMUNICATION PROTOCOLS

SECRET MESSAGE PASSING

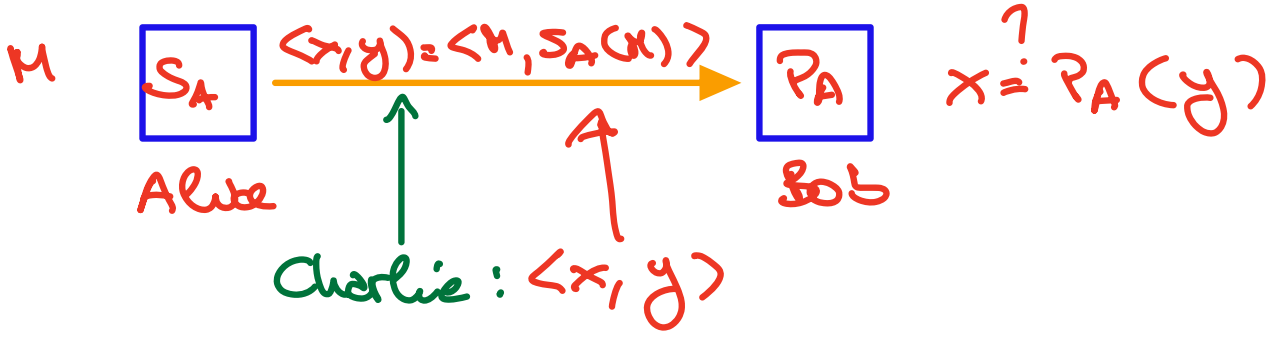
Bob wants to send M to Alice secretly:



Charlie (catching $y = P_A(M)$ over channel) is unable to obtain M

AUTHENTICATION

Alice wants to send message M to Bob so that Bob can be sure that Alice is the true sender. Assume that secrecy of M is not required.



If $x = P_A(y)$, Bob can be sure that the message was generated by Alice, since $S_A(M)$ cannot be computed by Charlie

APPLICATIONS: Electronic checks: bank "signs" credit messages. Certification authorities that distribute public keys

The two protocols can be combined: if Alice does not want M to be visible to everyone:

Alice sends $z = P_B(\langle M, S_A(M) \rangle)$

Bob computes $\langle x, y \rangle = S_B(z)$ and then checks $x = P_A(y)$

THE RSA (PUBLIC-KEY) CRYPTOSYSTEM

The asymmetry needed in constructing $P_A(M)$ and $S_A(M)$ in the RSA cryptosystem relies on the different complexity of these two decision problems:

PRIMALITY

$\begin{cases} I : \langle n \rangle : \ n \in \mathbb{N}^+ \\ Q : \text{is } n \text{ prime?} \end{cases}$

FACTORING

$\begin{cases} I : \langle n, k \rangle \\ Q : \exists p,q \in \mathbb{N} - \{1,0\} \ n = p \cdot q, \ 1 < p \leq k \end{cases}$

In FACTORING the second parameter is needed to actually compute the factorization.

Without k, we get

COMPOSITE

$\begin{cases} I : \langle n \rangle \\ Q : \text{is } n \text{ composite?} \end{cases}$

which is just PRIMES^c

These decision problems are related to these non decision ones (used by RSA)

LARGE-PRIMES: $I = \mathbb{N}^+$, $J = \mathbb{N}^+$

$n \Pi p \Leftrightarrow p$ prime, $p \geq n$

(ability to compute arbitrarily large primes)

FACTORS: $I = \mathbb{N}^+$, $J = \mathbb{N}^+ \times \mathbb{N}^+ \cup \{1\}$

$n \Pi 1$ if n prime

$n \Pi (p,q)$ if $p, q > 1$ and $n = p \cdot q$

(ability to factor composite numbers)

PRIMES $\in P$

- 2004: first (inefficient) deterministic polynomial algorithm
- 1976: Miller-Rabin (efficient) randomized test (can be incorrect)

FACTORING $\in NP$

- No efficient algorithms are known for FACTORING

Clearly COMPOSITE = $(\text{PRIMES})^C \Rightarrow$ COMPOSITE $\in P$!

- It is not known whether
  FACTORING $\in$ NPC
  (Probably not, since
  FACTORING $\in$ NP $\cap$ co-NP
  (PROVE IT))

- Difficult instances:
  $n = p \cdot q$, with p, q large primes

These two problems are somehow "inverse problems": I can easily generate two large primes p, q and multiply them together into $n = p \cdot q$
However given only $n = p \cdot q$ I am not able to obtain p, q efficiently.
This asymmetry is at the base of the RSA cryptosystem.