

## RECAP

- Congruent structures:  $\mathbb{Z}_n$   
Operations on  $\mathbb{Z}_n$ :  $+$ ,  $\cdot$ .
- $(\mathbb{Z}_n, +)$  is an additive group
- $\mathbb{Z}_n^* = \{z \in \mathbb{Z}_n : \boxed{\gcd(z, n) = 1}\}$   
 $(\mathbb{Z}_n^*, \cdot)$  is a multiplicative group  
COPRIME
- Computation of  $\bar{a}^{-1}$ ,  $a \in \mathbb{Z}_n^*$   
viz  $\bar{a} \in (\bar{a}, n)$
- Euler's function  $\varphi(n) = |\mathbb{Z}_n^*|$
- Euler's theorem  $\forall z : \gcd(z, n) = 1$   
 $\bar{a}^{\varphi(n)} \pmod{n} = 1$
- Fermat's little theorem
- If prime  $\forall a \in \mathbb{Z}_p^* (= \mathbb{Z}_p^+)$   
 $\bar{a}^{p-1} \pmod{n} = 1$

# THE CHINESE REMAINDER 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  $M$ , solution of systems  
of congruences ...  
↑  
large

THEOREM Let  $M = m_1 \cdot m_2 \cdot \dots \cdot m_k$ , with

$k > 1$  and  $\gcd(m_i, m_j) = 1$ ,  $1 \leq i \neq j \leq k$ .

Let  $f: \mathbb{Z}_M \rightarrow \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \dots \times \mathbb{Z}_{m_k}$

defined as  $f(\bar{a}) = (\bar{a} \bmod m_1, \bar{a} \bmod m_2, \dots, \bar{a} \bmod m_k)$

Then:

1.  $f$  is bijective
2.  $f((a+b) \bmod M) = ((a+b) \bmod m_1, \dots, (a+b) \bmod m_k)$
3.  $f((ab) \bmod M) = ((ab) \bmod m_1, \dots, (ab) \bmod m_k)$

Function  $f$  "converts" remainders mod  $M$  into  
 $k$ -tuples of remainders  $(\bmod m_1, \dots, \bmod m_k)$   
so that arithmetic mod  $M$  can be executed

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

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

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

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

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

$\Rightarrow$  Computing  $\{x \bmod p, x \bmod q\}$   
 costs about  $\frac{1}{2}$  half as computing  
 $x \bmod n$

## PROOF

Let us first prove:

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

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

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

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

$$\begin{aligned} \{(a+b) \bmod n\} \bmod n_i &= (a+b) \bmod n_i \\ \{(ab) \bmod n\} \bmod n_i &= (ab) \bmod n_i \end{aligned}$$

Therefore

$$\begin{aligned} f((a+b) \bmod n) &= ([a+b]_{n_1} \bmod n_1, \dots, [a+b]_{n_k} \bmod n_k) \\ &= ((a+b) \bmod n_1, \dots, (a+b) \bmod n_k) \end{aligned}$$

Let us prove 1. Since

$$|\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \dots \times \mathbb{Z}_{m_k}| = m_1 \cdot m_2 \cdot \dots \cdot m_k = |\mathbb{Z}_{m_1}| \times |\mathbb{Z}_{m_2}| \times \dots \times |\mathbb{Z}_{m_k}|$$

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

$$|\mathbb{Z}_{m_1}| = |\mathbb{Z}_n| \dots |\mathbb{Z}_{m_k}| \rightarrow \text{one implication}$$

Given  $(\alpha_1, \dots, \alpha_k) \in \mathbb{Z}_{m_1} \times \dots \times \mathbb{Z}_{m_k}$ , we will determine  $x \in \mathbb{Z}_n$  such that

$$f(x) = (x \bmod m_1, \dots, x \bmod m_k) = (\alpha_1, \dots, \alpha_k)$$

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

$$\begin{cases} x \equiv \alpha_1 \pmod{m_1} \\ x \equiv \alpha_2 \pmod{m_2} \\ \vdots \\ x \equiv \alpha_k \pmod{m_k} \end{cases}$$

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

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

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

In other words,  $m_i = \frac{n}{m_i} = m_1 \cdot \dots \cdot m_i \cdot m_{i+1} \cdot \dots \cdot m_k$

Clearly  $\gcd(m_i, m_i) = 1$  (since  $\gcd(a, a) = 1$ )  
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  $c_i = m_i \cdot (m_i^{-1} \bmod n_i)$

We set

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

principal  
representative  
of  $([m_i]_{n_i})^{-1}$

and prove that

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

Observe that for  $1 \leq i \neq j \leq k$ :

$$c_i \bmod n_j = 0$$

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

Therefore

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

$$\stackrel{m_4}{=} \left( \sum_{i=1}^k a_i \cdot c_i \right) \bmod n_j \quad = 0 \text{ for } i \neq j$$

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

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

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

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

$$\begin{aligned} m_2 &= \left( \sum_j \left( (m_j m_j^{-1}) \text{mod } u_j \right) \right) \text{mod } u_j \\ &= \sum_j \sum_i a_{ij} \text{mod } u_j = a_j. \end{aligned}$$

Q.E.D.

**COROLLARY 1** Let  $m_1, \dots, m_k : \gcd(u_i, u_j) = 1$  for  $1 \leq i \neq j \leq k$ . Then the system of congruences

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

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

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

**COROLLARY 2** Let  $m_1, \dots, m_k : \gcd(u_i, u_j) = 1$  for  $1 \leq i \neq j \leq k$ ,  $n = \prod m_i$ . Then

$$(x \equiv y) \pmod{n} \iff (x \equiv y) \pmod{m_i}, 1 \leq i \leq k$$

⇒ If  $x \pmod{n} = y \pmod{n}$

then  $(x \pmod{n}) \pmod{m_i} = (y \pmod{n}) \pmod{m_i}$

that is, from m4:  $x \pmod{m_i} = y \pmod{m_i}$

for all  $1 \leq i \leq k$

⇐ If  $x \equiv y \pmod{m_i}, 1 \leq i \leq k$ , then  $f(x) = f(y)$

(since  $f(a) = (a \pmod{m_1}, \dots, a \pmod{m_k})$ )

Therefore  $x \equiv y \pmod{n}$  because  $f$  is injective over  $\mathbb{Z}_n$

## EXAMPLE

Consider the system

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

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

$$m_1 = 5 \quad m_2 = 13 \quad \gcd(5, 13) = 1 \quad \text{OK}$$

$$m_1^{-1} = \frac{13}{m_1} = 13 \quad m_2^{-1} = \frac{5}{m_2} = 5$$

$$m_1^{-1} \pmod{5} = 13^{-1} \pmod{5} = 2, \text{ since } 13 \cdot 2 \pmod{5} = 26 \pmod{5} = 1$$

$$13 \cdot 2 \pmod{5} = 26 \pmod{5} = 1$$

$$m_2^{-1} \pmod{13} = 5^{-1} \pmod{13} = 8, \text{ since } 5 \cdot 8 \pmod{13} = 40 \pmod{13} = 1$$

$$5 \cdot 8 \pmod{13} = 40 \pmod{13} = 1$$

$$\text{Therefore } C_1 = 13 \cdot 2 = 26, C_2 = 5 \cdot 8 = 40$$

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

$$= (52 + 120) \pmod{65} =$$

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

# 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  $\mathbb{Z}_N$ )

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

PROPERTIES :

- $e_K$  is invertible and efficiently computable, given  $K$ .

- **HED**: 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:

- $R_x$  : public Key
- $S_x$  : secret Key

### EXAMPLE

Alice :  $R_A, S_A$

Bob :  $R_B, S_B$

Each participant creates its own keys

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

Each key corresponds to an encoding function:

### EXAMPLE: HED

$e_{R_A}(M), e_{S_A}(M), e_{R_B}(N), e_{S_B}(N)$

|||

$R_A(M), S_A(M), R_B(N), S_B(N)$

## PROPERTIES

1.  $P_x(H)$ ,  $S_x(H)$  "efficiently computable," given key  $P_x$  or  $S_x$
2.  $S_x(H) = P_x^{-1}(H)$ , that is
 
$$S_x(P_x(H)) = P_x(S_x(H)) = H$$
3. Even if  $P_x(H)$  is known, computing  $S_x(H) = P_x^{-1}(H)$  is computationally infeasible if  $S_x$  is not known

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

EXAMPLE  $D = \mathbb{Z}_n$

$$P_A(H) = H + 7 \pmod{n}$$

$$\Rightarrow S_A(Y) = Y - 7 \pmod{n}$$

$$P_A(H) = 2H \pmod{n}, 2 \in \mathbb{Z}_n^*$$

$$\Rightarrow S_A(Y) = 2^{-1}Y \pmod{n}$$

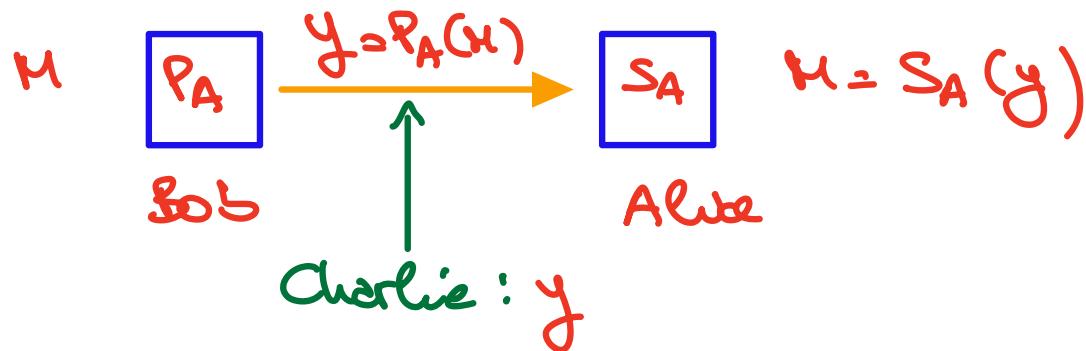
NOTE For long messages we use repeated application on  $K$ -tuples,  $K \geq 1$ :

$$P_A(\langle H_1, \dots, H_K \rangle) = \langle P_A(H_1), \dots, P_A(H_K) \rangle$$

# COMMUNICATION PROTOCOLS

## SECRET MESSAGE PASSING

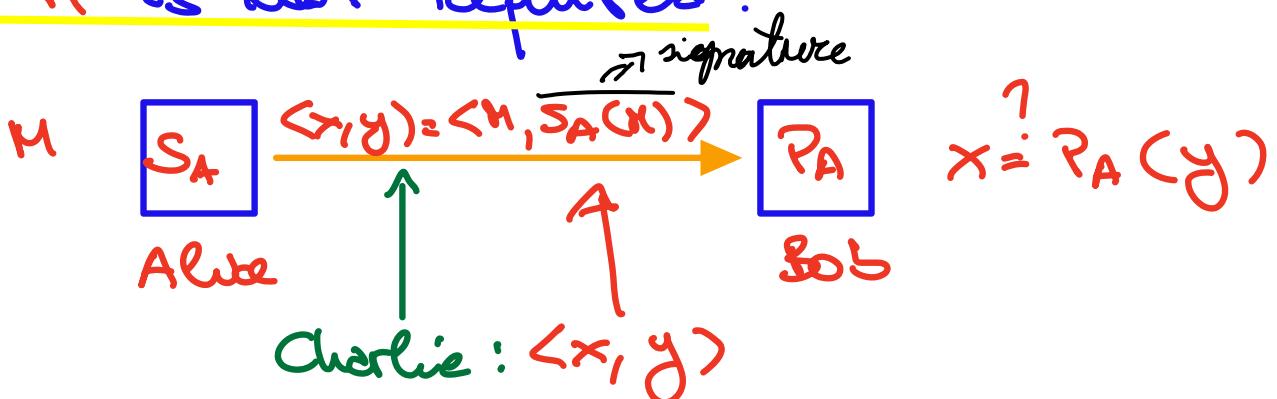
Bob wants to send  $M$  to Alice secretly:



Charlie (catching  $y = P_A(x)$  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  $z = R_B(M, S_A(M))$

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

# THE RSA (PUBLIC-KEY) CRYPTOSYSTEM

The asymmetry needed in constructing  $P_A(M)$  and  $S_A(N)$  in the RSA crypto-system 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

$$\begin{cases} I : \langle n, k \rangle \\ Q : \exists p, q \in \mathbb{N} \setminus \{1\} \text{ s.t. } n = p \cdot q, 1 < p \leq k \end{cases}$$

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<sup>c</sup>

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

LARGE-PRIMES :  $\mathcal{D} = \mathbb{N}^+$ ,  $\mathcal{D} = \mathbb{N}^+$

$n \in \mathbb{N} \Leftrightarrow p$  prime,  $p \geq n$

(ability to compute arbitrarily large primes)

FACTORS :  $\mathcal{D} = \mathbb{N}^+$ ,  $\mathcal{D} = \mathbb{N}^+ \times \mathbb{N}^+ \cup \{\perp\}$

$n \in \mathbb{N} \perp$  if  $n$  prime

$n \in \{(p, q)\}$  if  $p, q > 1$  and  $n = p \cdot q$

(ability to factor composite numbers)

## PRIMES $\in \mathbb{P}$

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

## FACTORING $\in \mathbb{NP}$

- No efficient algorithms are known for FACTORING

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

$\text{COMPOSITE} \in \mathbb{P}^C$

- It is not known whether  
 $\text{FACTOING} \in \text{NP-C}$   
 (possibly not, since  
 $\text{FACTOING} \in \text{NP} \wedge \text{Co-NP}$ )  
 (PROVE IT)  $\uparrow$

- 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.