# Grupos

El par $(G, *)$

$G$: conjunto — $*$: operador binario: $G \times G \to G$

debe cumplir:

- cerrado
- identidad
- inverso
- asociatividad

Nota:

dado $m = |G|$

entonces $g^m = \text{identidad}$

## ¿Que es $\mathbb{Z}_N$?

1. Defina el grupo $\mathbb{Z}_N$
   - Sea un entero $N > 1$.
   - Definimos el conjunto $G = \{1, \ldots, N-1\}$
   - Definimos el operador "+" tal que $a + b = (a+b) \bmod N \quad \forall a, b \in N$

Nota:

a veces se usa $\mathbb{Z}_N$ como el conjunto y a veces como el grupo

2. demuestre que $\mathbb{Z}_N$ es un grupo

3. ¿Qué es la multiplicación y exponenciación de elementos?

- $mg = m \cdot g = \underbrace{g + g + \ldots + g}_{m \text{ veces}}$
- $g^m = \underbrace{g \cdot g \cdot \ldots \cdot g}_{m \text{ veces}}$

Nota: acá, "+" puede ser cualquier operador válido sobre el Grupo

## ¿Qué es $\mathbb{Z}_N^*$ y $\phi(N)$?

$$G = \{ b \in \{1, \dots, N-1\} \mid \gcd(b, N) = 1 \}$$

($\mathbb{Z}_N$ y $b$ coprimo)

Es decir, el conjunto de números enteros menores a $N$, tal que son coprimos de $N \leftrightarrow b$ invertible en $N$

$*: G \times G \to G$ es el operador multiplicación en módulo $N$.

- $\mathbb{Z}_N^* = (G, *)$

$\Rightarrow$

- $\phi(N) = |\mathbb{Z}_N^*|$ (función phi de Euler)

## ¿Qué es $\mathbb{Z}_p^*$?

- Un caso especial de $\mathbb{Z}_N^*$, cuando $N$ es un número primo $p$.
- Notamos que $\phi(p) = |\mathbb{Z}_p^*| = p-1$

$$a^{\phi(p)} = 1 \bmod p \longrightarrow a^{p-1} = 1 \bmod p$$

(Nota: esto es el pequeño teo. de Fermat)

# ¿Cómo funciona El Gamal?

Grupo $G$, generador $g$, orden subgrupo $q$

## Receiver / Holder

init:
- ¿es $q$ un primo?
- ¿es $g^q$ = identidad?
- $x \in \{1, \dots, q-1\}$

get_public_key: $(G, g, q, g^x)$

decrypt: input: $(e, g^y)$

$e = m \cdot s$

$pub = g^y$

$s^{-1} = (g^y)^{(q-x)}$

$\Rightarrow m = e \cdot s^{-1} = m \cdot s \cdot s^{-1}$

return: $(m)$

## Sender / Holder

init: recibir $(G, g, q, g^x)$

encrypt:

$y \in \{1, \dots, q-1\}$

encryptor $= g^{xy}$

$e = m \cdot$ encryptor

$\Rightarrow$ return $(e, g^y)$

# ¿Cómo funciona Schnorr?

## firmar m:

input: $m$

$k \in \{1, \dots, q-1\}$

$r = g^k$

$v = H(r \,\|\, m)$ ← función Hash criptográfica

$s = k + (v \cdot x)$ ← llave secreta

return $(v, s)$

$G, g, q, g^x \longrightarrow$ (llave pública)

## verificar m:

input: $m, v, s$

$\alpha = g^s$

$\beta = \alpha \cdot (g^x)^{(q-v)}$

$v' = H(\beta \,\|\, m)$

return $(v' == v)$

# Nota:

- str.encode() : str → bytes
- H : bytes → hexdigest (string b16)
- int(hex,16) → int

## Curvas Elípticas

Forma de Weierstrass, lo pedido en $T_3$

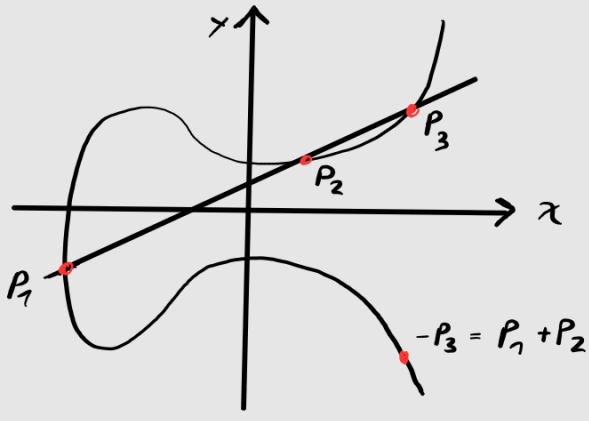
$$E(\mathbb{Z}_p) = \{(x,y) \; x,y \in \mathbb{Z}_p \wedge y^2 = x^3 + Ax + B \bmod p\} \cup \{\mathcal{O}\}$$

$(x,y)$: puntos sobre la curva elíptica

$y^2 = x^3 + Ax + B \bmod p$: ecuación de la curva

$\mathcal{O}$: elemento neutro, el "$\infty$" en "$y$" está en toda curva.

- operador: "+", es el reflejo en x del intercepto de $\overline{P_1P_2}$.




↗ Por definición

- Acá, "exponenciar" es suma repetida
  - pow -- : -- mul --

## Operación suma:

def) $P_1 == P_2 \iff x_1 == x_2 \wedge y_1 == y_2$

1. si $x_1 \neq x_2$: $P_1 + P_2 = (x_3, y_3)$ con:

$$s = \frac{y_2 - y_1}{x_2 - x_1} \bmod p$$

⇒ división es multiplicación por el inverso

luego: $x_3 = s^2 - x_1 - x_2 \bmod p$ , $y_3 = s \cdot (x_1 - x_3) - y_1 \bmod p$

2. $x_1 == x_2 \wedge y_1 \neq y_2 : \quad P_1 + P_2 = \mathcal{O}$

3. $P_1 == P_2 \wedge y_1 \neq 0 : \quad P_1 + P_2 = 2P_1 = (x_3, y_3)$ con:

$$s = \frac{3x_1 + A}{2y_1} \bmod p$$

$$x_3 = s^2 - x_1 - x_2 \bmod p \;, \quad y_3 = s \cdot (x_1 - x_3) - y_1 \bmod p$$

4. $P_1 == P_2 \wedge y_1 == 0$, luego $P_1 + P_2 = \mathcal{O}$