

# Matematicas Discretas y Criptografia

Julino

4 de abril de 2024

## 1. Aritmetica Modular

Todo numero entero 'p'  $\in \mathbb{Z}$  puede expresarse de la siguiente manera:

$$\begin{aligned} p &= nq + r \mid n, q \in \mathbb{Z} \text{ y } r \geq 0 \\ r &= p - nq \end{aligned} \tag{1}$$

Eso significa que si  $R = 0$ , tanto  $n$  como  $q$  dividen a  $p$ , sin embargo nos enfocaremos en  $n$  para trabajar en la aritmetica modular

Definimos entonces la aritmetica modular como el resto que se obtiene al dividir un numero 'A' entre un modulo 'N'.

$$A \bmod N = R \implies A = Nq + R \mid 0 \leq R < N$$

**Ejemplo:**  $16 \bmod 5 = 1$

Se puede calcular el modulo  $N$  de un numero negativo, sin embargo en la Criptografia nos interesa trabajar solamente con numeros positivos. Aun asi esta es la manera de trabajar con numeros negativos:

$$A \bmod N = (A + Nk) \bmod N$$

**Ejemplo:**  $-5 \bmod 3 = (-5 + 3 \times 2) \bmod 3 = 1 \bmod 3 = 1$

Nota: Siempre que  $A < N$ ,  $A \bmod N = A$

### 1.1. Congruencia

Definimos la congruencia de dos numeros enteros  $A$  y  $B$  con el modulo  $N$  ( $A \equiv B \bmod N$ ) si ambos numeros generan el mismo resto  $R$  ( $A = Nq + R$  y  $B = Nq' + R$ )

**Teorema 1.1.**

$$A \equiv B \bmod N \implies N \mid (a - b)$$

*Demostración.* Usando 1.1 tenemos que  $a - b = Nq + r - Nq' - R = N(q - q')$ . Pero  $q - q' \in \mathbb{Z} \implies a - b = Nq'' \implies N \mid (a - b)$

## 2. Numeros Primos y Compuestos

Las operaciones modulares en Criptografia se realizan dentro de un modulo de cifra, cuyo valor puede ser un numero primo o compuesto.

Un numero primo es aquel numero que solo es divisible por 1 y por el mismo, es decir, sea  $P$  primo  $P \in \mathbb{N}$ ,  $P > 1$  y los unicos divisores de  $P$  son 1 y  $P$

Un numero compuesto es un numero natural no primo. Sea  $N$  un numero compuesto,  $N \in \mathbb{N}$ ,  $N > 1$  y  $\exists n \in \mathbb{N}$ ,  $n \notin \{1, N\} \mid n/N$

## 2.1. Cardinalidad de los Numeros Primos

Existen infinitos primos pero la cantidad de primos que existe en el intervalo  $[2, X]$  viene dada por el teorema de los numeros primos:

**Teorema 2.1.**

$$|[2, x]| \approx \pi(x) \approx \frac{x}{\ln(x)}$$

*Demostración.* La demostracion es sencilla y se deja como ejercicio al lector

**Ejemplo:** ¿Cuántos primos hay de  $2^{1024}$  bits?

Es decir, queremos calcular cuantos primos existen que se representen con 1024 bits. Para ello calculamos cuantos primos en total hay en  $2^{1024}$  y le restamos los primos que necesiten menos de 1024 bits para representarlos. Sea  $P_{1024}$  la cantidad de primos de 1024 bits:

$$P_{1024} \approx \pi(2^{1024}) - \pi(2^{1023}) \approx 1,26 \times 10^{305}$$

En 'cristiano': ¡Es una cantidad inmesamente grande!

## 2.2. Primo Seguro

Se define primo seguro al numero primo  $P$  que satisface la siguiente condicion:  $P = 2P' + 1$ , donde  $P'$  es primo tambien. **Ejemplo:**  $23 = 2 \times 11 + 1$ . Claramente no todos los  $2P' + 1$  son numeros primos, como ejemplo tenemos al numero **15**

## 2.3. Primo Relativo

Es un concepto comparativo, se dice que 2 numeros  $a, b \in \mathbb{P}$  (donde  $\mathbb{P}$  es el conjunto de los primos) son coprimos o primos relativos  $\iff \text{mcd}(a, b) = 1$ . Es decir no tienen factores en comun.

## 2.4. Numero Compuesto

Un numero compuesto es cualquier numero que tiene mas de 2 divisores, es decir se puede dividir por 1, por el mismo y por algun otro conjunto de factores primos. Todo numero no primo es compuesto. En este caso solo nos interesan los numeros compuestos del tipo  $p \times q$ , con  $p$  y  $q$  primos.

**Ejemplo:**  $3 \times 5 \times 7$  es un numero compuesto del tipo que no nos interesa

## 3. Conjunto de Restos

El conjunto de resto es el conjunto que contiene todos los posibles restos de un modulo  $N$ . Existen 2 tipos, el completo y el reducido. Es de hacer notar que este ultimo es muy importante en la Criptografia asimetrica.

### 3.1. Conjunto Completo de Restos

Es el conjunto que contiene a todos los restos del modulo n:

$$CCR = \{0, 1, 2, \dots, n-2, n-1\}$$

### 3.2. Conjunto Reducido de Restos

Es el conjunto que contiene todos los restos que son primos relativos(2.3) con n.

$$CRR = \{1, n_1, n_2, \dots, n_i\} / n_i < n, \text{ mcd}(n_i, n) = 1$$

Si n es primo entonces  $CRR = \{1, 2, 3, \dots, n-2, n-1\}$ . Vemos que el 0 se descarta ya que no es solución ( $\text{mcd}(0, n) \neq 1$ ). Por supuesto que  $CRR \subset CCR$ .

Ejemplo:  $n = 8$ ,  $CCR = \{0, 1, 2, 3, 4, 5, 6, 7\}$ ,  $CRR = \{1, 3, 5, 7\}$

### 3.3. Funcion de Euler( $\phi(n)$ )

Gracias al gran matematico Leonhard Euler tenemos la funcion que lleva su nombre que nos permite calcular el cardinal del CRR ( $\phi(n) = |CRR|$ ). Este numero es muy importante y servira para conseguir el inverso de un modulo n, sobre eso hablaremos mas adelante.

Existen 4 casos donde  $\phi(n)$  funciona:

- n es primo
- $n = p^k$ , p es primo y  $k \in \mathbb{Z}$
- $n = p \times q$ , p y q primos
- $n = p_1^{e_1} \times p_2^{e_2} \times \dots \times p_n^{e_n} = \prod_{i=1}^n p_i^{e_i}$

Estudiaremos los unicos 2 casos que nos importa para los objetivos de este documento (que es la Criptografia)

#### 3.3.1. Caso n primo

Si n es primo entonces  $\phi(n) = |\{1, 2, \dots, n-2, n-1\}| = |CRR| = n-1$

**Ejemplo:** Sea  $n=7$ ,  $\phi(7) = 7-1 = 6$ .  $CRR_7 = \{1, 2, 3, 4, 5, 6\} \implies |CRR| = 6$

#### 3.3.2. Caso n 'compuesto'

Si p y q son primos entonces  $\phi(n) = \phi(p \times q) = \phi(p) \times \phi(q) = (p-1) \times (q-1)$

*Demostración.* Queda pendiente

**Ejemplo:** Sea  $n = 15(5 \times 3)$ ,  $\phi(15) = (p-1) \times (q-1) = (5-1) \times (3-1) = 8$ .  
 $CRR_{15} = \{1, 2, 4, 7, 8, 11, 13, 14\}$

## 4. Inversos

Esta es sin duda la parte mas importante ya que en Criptografia (al menos en la asimetrica) el inverso es lo que nos permite deshacer o revertir una operacion, si ciframos con un numero  $C$  el inverso de  $C$  ( $inv(c)$ ) nos permitira descifrar, y si ciframos con  $inv(C)$  entonces  $C$  nos permitira descifrar, es decir, uno deshace lo que hizo el otro.

Existen 3 tipos de inversos que veremos: Aditivo, XOR y Multiplicativo, siendo este ultimo el que se usa en la Criptografia asimetrica

### 4.1. Inverso Aditivo

Cifrado y Descifrado, donde 'K' es la clave:

$$\begin{aligned}c_i &= m_i + K \quad \text{mód } n \\m_i &= c_i + inv_+(K, n) \quad \text{mód } n\end{aligned}\tag{2}$$

En el caso aditivo es muy sencillo encontrar el inverso:

$$inv_+(K, n) = n - K, \quad n > K$$

Entonces el inverso aditivo (2) queda de la siguiente manera:

$$m_i = c_i + (n - K) \quad \text{mód } n$$

### 4.2. Inverso XOR

Es la misma operacion para cifrar y describrar (K es la clave):

$$\begin{aligned}\text{Cifrado} : C &= M \oplus K \\ \text{Descifrado} : M &= C \oplus K\end{aligned}$$

### 4.3. Inverso Multiplicativo

El inverso multiplicativo de un **resto**  $k$  (logicamente  $k < n$ ), no siempre existe.

- $inv_{\times}(k, n) \equiv 1 \quad \text{mód } n \iff mcd(k, n) = 1$  (i.e  $k, n$  coprimos)
- $k \times inv_{\times}(k, n) \quad \text{mód } n = 1 \implies mcd(k \times i_k, n) = 1 \mid i_k = inv_{\times}(k, n)$
- Si  $n$  es primo entonces  $k$  tiene inverso
- Si  $n$  es compuesto y  $mcd(k, n) \neq 1$  entonces  $k$  no tiene inverso.

$$\begin{aligned}\text{Cifrado} : C &= k \times m_i \quad \text{mód } n \\ \text{Descifrado} : M &= c_i \times inv_{\times}(k, n) \quad \text{mód } n\end{aligned}$$

Pregunta:  $inv_{\times}(k, n) < n$ ? No lo se pero si existe y es menor a  $n$  entonces el inverso es unico dentro de los restos de  $n$ .

## 5. Calculo del Inverso Multiplicativo ( $inv_{\times}(k, n)$ )

Existen varias maneras para calcular el inverso multiplicativo:

### 5.1. Fuerza Bruta

Se prueban todos los restos de  $n$ . Si  $n$  es primo entonces la cantidad de restos a probar es  $n-1$  (excluyendo el 1). Si  $n$  es compuesto entonces solo se prueba con los restos que sean coprimos con el modulo ( $mcd(i', n) = 1$ )

### 5.2. Pequeño Teorema de Fermat

$$\text{Sea } n \text{ primo y } a > 0 \text{ coprimo con } n, \quad a^{n-1} \equiv 1 \pmod{n} \quad (3)$$

Sea  $a$  un resto dentro del modulo  $n$ , usando 3 tenemos que  $a^{n-1} \pmod{n} = 1$ , por otra parte tenemos que  $a \times x \pmod{n} = 1$  ( $x$  es  $inv_{\times}(a, n)$ ). Entonces tenemos lo siguiente:

$$\begin{aligned} a^{n-1} \pmod{n} &= a \times x \pmod{n} \\ a^{n-2} \pmod{n} &= x \pmod{n} \\ x &= a^{n-2} \pmod{n}^* \end{aligned}$$

Usando la funcion de Euler (3.3) podemos reescribir la ecuacion:

$$\begin{aligned} \phi(n) &= n - 1 \\ x &= a^{\phi(n)-1} \end{aligned}$$

Cuando  $n$  es compuesto tengo que calcular el  $\phi(n)$ . Si  $n$  es muy grande el calculo del inverso usando al pequeño fermat sigue siendo muy costoso por lo que este metodo no sirve para la Criptografia asimetrica donde los  $n$  son enormes.

### 5.3. Algoritmo Extendido de Euclides (AEE)

Si bien es mejor que el pequeño fermat este metodo sigue siendo lento para un  $n$  grande. No obstante aca esta el algoritmo:

```
// Esto es pseudocodigo
// Linea blanca intencional
// Linea blanca intencional
// Linea blanca intencional
// Linea blanca intencional
// Linea blanca intencional
```

---

\* Dado que  $x < n \implies x \pmod{n} = x$

```

func AEE(a, n)
    (g[0],g[1],u[0],u[1],v[0],v[1],i) = (n,a,1,0,0,1,1)

    while g[i] != 0
        y[i+1] = g[i-1]/g[i] // Parte entera
        g[i+1] = g[i-1]-(y[i+1]*g[i])
        u[i+1] = u[i-1]-(y[i+1]*u[i])
        v[i+1] = v[i-1]-(y[i+1]*v[i])
        i = i+1
    endwhile

    if v[i-1] < 0
        v[i-1] = v[i-1] + b
    endif

    return v[i-1]
endfunc

```

#### 5.4. Algoritmo Exponenciación Rápida (AER)

Este es el que se usa en Criptografía asimétrica y queda como tarea pendiente de implementar (bits y programación dinámica)

## 6. Raíces Primitivas en un Primo $p$

Se denomina raíz primitiva  $\alpha$  de  $p$  al  $\alpha$  que cumple lo siguiente:

$$\alpha^{x_i} \bmod p = CRR, \quad 0 \leq x_i \leq p-1$$

La raíz primitiva genera el CRR (conjunto reducido de restos) del primo  $p$ . Si  $\alpha$  es raíz primitiva de  $p$  entonces tenemos las siguientes propiedades:

- $\alpha^0 \bmod p = y_0 = 1$
- $\alpha^{p-1} \bmod p = y_{p-1} = 1$
- $\alpha^{x_i} \bmod p = y_{x_i}, \quad x_i = \{1, 2, \dots, p-3, p-2\}$
- $y_{x_i} = \{2, 3, \dots, p-1\}$  (sin orden particular)

### 6.1. Problema del Logaritmo Discreto

La fórmula es:  $x = \log_{\alpha} \beta \bmod p$ , si  $\alpha, \beta$  y  $p$  son conocidos