### Fonaments matemàtics de la criptografia

#### Carlos Borrego

Carlos.Borrego@uab.cat

Departament d'Enginyeria de la Informació i de les Comunicacions
Universitat Autònoma de Barcelona

Criptografia i Seguretat

#### Content

- 1 Matemática Discreta
- 2 Algoritmo de Eucliedes
- 3 Inversos en  $\mathcal{Z}_n$
- 4 Función de Euler
- 5 Exponenciación Rápida
- 6 Números Primos

Material adaptat de:

Curso de Seguridad Informática y Criptografía

Dr. Jorge Ramió Aguirre

Universidad Politécnica de Madrid

http://jramio.etsisi.upm.es/

# Contingut

- 1 Matemática Discreta
- 2 Algoritmo de Eucliedes
- 3 Inversos en  $\mathbb{Z}_n$
- 4 Función de Euler
- 5 Exponenciación Rápida
- 6 Números Primos

# Matemática discreta y congruencia

- La congruencia es la base en la que se sustentan las operaciones de cifra en matemática discreta.
- Concepto de congruencia:
  - Sean dos números enteros a y b: se dice que a es congruente con b en el módulo o cuerpo n (Z<sub>n</sub>) si y sólo si existe algún entero k que divide de forma exacta la diferencia (a - b).
  - Esto podemos expresarlo así:

a - b = k \* n $a \equiv b$  $a \equiv b \mod n$ 

# Propiedades de la congruencia en Z<sub>n</sub>

• Propiedad Reflexiva:

$$a \equiv a \mod n \quad \forall \ a \in Z$$

• Propiedad Simétrica:

$$a \equiv b \mod n \Rightarrow b \equiv a \mod n \quad \forall \ a,b \in Z$$

• Propiedad Transitiva:

Si 
$$a \equiv b \mod n$$
 y  $b \equiv c \mod n$   
 $\Rightarrow a \equiv c \mod n$   $\forall a,b,c \in Z$ 

# Propiedades de las operaciones en $Z_n(1)$

• Propiedad Asociativa:

$$a + (b + c) \mod n \equiv (a + b) + c \mod n$$

• Propiedad Conmutativa:

$$a + b \mod n \equiv b + a \mod n$$

$$a * b \mod n \equiv b * a \mod n$$

Normalmente usaremos el signo = en vez de = que denotaba congruencia. Esto es algo propio de los Campos de Galois que veremos más adelante.

• Propiedad Distributiva:

$$a * (b+c) \mod n \equiv ((a * b) + (a * c)) \mod n$$

$$a * (b+c) \mod n = ((a * b) + (a * c)) \mod n$$

# Propiedades de las operaciones en $Z_n(2)$

• Existencia de Identidad:

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

$$a * 1 \mod n = 1 * a \mod n = a \mod n = a$$

• Existencia de Inversos:

a + (-a) mod n = 0

Ambos serán muy importantes en criptografía

$$a * (a^{-1}) \mod n = 1$$
 (si  $a \ne 0$ )  $\longrightarrow$  No siempre existe

• Reducibilidad: ✓

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

 $(a * b) \mod n = [(a \mod n) * (b \mod n)] \mod n$ 

#### Conjunto completo de restos CCR

Para cualquier entero positivo n, el conjunto completo de restos será  $CCR = \{0, 1, 2, ... n-1\}$ , es decir:

 $\forall a \in Z \quad \exists ! r_i \in CCR / a \equiv r_i \mod n$ 



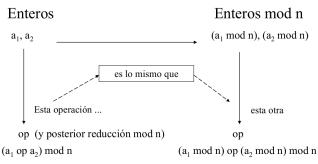
$$CCR(11) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

CCR 
$$(6) = \{0, 1, 2, 3, 4, 5\} = \{12, 7, 20, 9, 16, 35\}$$

El segundo conjunto es equivalente:  $12 \rightarrow 0, 7 \rightarrow 1...$ 

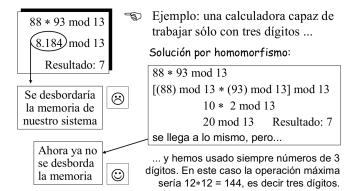
Normalmente se trabajará en la zona canónica: 0 – n-1

#### Homomorfismo de los enteros



Esto nos permitirá trabajar con números muy grandes

### Un ejemplo de homomorfismo



# En criptografía muchas veces nos interesará encontrar el máximo común denominador med entre dos números *a* y *b*.

Para la existencia de inversos en un cuerpo n, la base a y el módulo n deberán ser primos entre sí.  $\Rightarrow$  mcd (a, n) = 1

#### Algoritmo de Euclides:

- a) Si x divide a  $a y b \implies a = x * a' y b = x * b'$
- b) Por lo tanto: a k \* b = x \* a' k \* x \* b'
  - a k \* b = x (a' k \* b')
- c) Entonces se concluye que x divide a (a k \* b)

#### El máximo común denominador med

Como hemos llegado a que x divide a (a - k \* b) esto nos permitirá encontrar el mcd (a, b):

Si 
$$a > b$$
 entonces  $a = d_1 * b + r$  (con  $d_i$  un entero y r un resto)

Luego  $mcd(a, b) = mcd(b, r)$   $(a > b > r \ge 0)$  porque:

Si  $b > r$  entonces  $b = d_2 * r + r$ 

Si b > r entonces  $b = d_2 * r + r'$ (con r un entero y r' un resto)

- Matemática Discreta
- 2 Algoritmo de Eucliedes
- 3 Inversos en  $\mathbb{Z}_n$
- 4 Función de Euler
- **5** Exponenciación Rápida
- 6 Números Primos

### Divisibilidad con algoritmo de Euclides



- Matemática Discreta
- 2 Algoritmo de Eucliedes
- 3 Inversos en  $\mathcal{Z}_n$
- 4 Función de Euler
- 5 Exponenciación Rápida
- 6 Números Primos

## Inversos en Z<sub>n</sub>

Si  $a * x \equiv 1 \mod n$ 

se dice que x es el inverso multiplicativo de a en  $Z_n$  y se denotará por  $a^{-1}$ .

- No siempre existen el inverso de un elemento en Z<sub>n</sub>.
   Por ejemplo, si n = 6, en Z<sub>6</sub> no existe el inverso del 2, pues la ecuación 2\*x = 1 mod 6 no tiene solución.
- Si n es un número primo p, entonces todos los elementos de Z<sub>p</sub> salvo el cero tienen inverso. Por ejemplo, en Z<sub>5</sub> se tiene que:

 $1^{-1} \mod 5 = 1$ ;  $2^{-1} \mod 5 = 3$ ,  $3^{-1} \mod 5 = 2$ ;  $4^{-1} \mod 5 = 4$ .

### Existencia del inverso por primalidad

 $\exists$  inverso  $a^{-1}$  en mod n ssi mcd(a, n) = 1

Si mcd(a,n) = 1, el resultado de a\*i mod n (para i todos los restos de n) serán valores distintos dentro del cuerpo n.



Si mcd  $(a,n) \neq 1$ 

### Inexistencia de inverso (no primalidad)

¿Y si no hay primalidad entre a y n?

Si mcd 
$$(a, n) \neq 1$$

No existe ningún x que 0 < x < n / a \* x mod n = 1

Sea: 
$$a = 3$$
 y  $n = 6$  Valores de  $i = \{1, 2, 3, 4, 5\}$ 

$$3*1 \mod 6 = 3$$
  $3*2 \mod 6 = 0$   $3*3 \mod 6 = 3$ 

$$3*4 \mod 6 = 0$$
  $3*5 \mod 6 = 3$ 

No existe el inverso para ningún resto del cuerpo.



### Inversos aditivo y multiplicativo

(A+B) mod 5					(A*B) mod 5							
B +	0	1	2	3	4	В	*	0	1	2	3	4
A 0	0	1	2	3	4	$\overline{A}$	0	0	0	0	0	0
1	1	2	3	4	0	0+0 = 0	1	0	1	2	3	4
2	2	3	4	0	1	1*1 = 1	2	0	2	4	1	3
3	3	4	0	1	2	Es trivial	3	0	3	1	4	2
4	4	0	1	2	3		4	0	4	3	2	1

- o En la operación suma siempre existirá el inverso o valor identidad de la adición (0) para cualquier resto del cuerpo. Su valor es único.
- o En la operación producto, de existir un inverso o valor de identidad de la multiplicación (1) éste es único y la condición para ello es que el número y el módulo sean primos entre sí. Por ejemplo para n = 4, el resto 2 no tendrá inverso multiplicativo, en cambio el resto 3 sí.

### No existencia de inversos multiplicativos

(A*B) mod 10											
	1	2	3	4	5	6	7	8	9		
1	1	2	3	4	5	6	7	8	9		
2	2	4	6	8	0	2	4	6	8		
3	3	6	9	2	5	8	1	4	7		
4	4	8	2	6	0	4	8	2	6		
5	5	0	5	0	5	0	5	0	5		
6	6	2	8	4	0	6	2	8	4		
7	7	4	1	8	5	2	9	6	3		
8	8	6	4	2	0	8	6	4	2		
9	9	8	7	6	5	4	3	2	1		

Para módulo 10 sólo encontramos inversos multiplicativos en los restos 3, 7 y 9, puesto que los demás restos tienen factores 2 y 5 en común con el módulo.

#### Conjunto reducido de restos CRR

- El conjunto reducido de restos, conocido como CRR de n, es el subconjunto {0, 1, ... n<sub>i</sub>, ... n-1} de restos, primos con el grupo n.
- Si n es primo, todos los restos serán primos con él.
- Como el cero no es una solución, entonces:

CRR = 
$$\{1, ..., n_i, ... n-1\} / mcd (n_i, n) = 1$$
  
Ejemplo: CRR mod  $8 = \{1, 3, 5, 7\}$   
CRR mod  $5 = \{1, 2, 3, 4\}$ 

#### Utilidad del CRR

¿Qué utilidad tiene esto en criptografía?

El conocimiento del CRR permitirá aplicar un algoritmo para el cálculo del inverso multiplicativo de un número x dentro de un cuerpo n a través de la función  $\phi(n)$ , denominada Función de Euler o Indicador de Euler.

Será importante en todos los sistemas simétricos que trabajan en un módulo (con excepción del DES que es un caso muy especial de cifra no modular) y más aún en los sistemas asimétricos y en particular RSA ya que los cálculos de claves pública y privada se harán dentro del cuerpo  $\phi(n)$ . En ambos casos la cifra y las claves estarán relacionadas con el CRR.

# Contingut

- Matemática Discreta
- 2 Algoritmo de Eucliedes
- 3 Inversos en  $\mathbb{Z}_n$
- 4 Función de Euler
- 5 Exponenciación Rápida
- 6 Números Primos

### Función de Euler $\phi(n)$

- El Indicador o Función de Euler φ(n) nos entregará el número de elementos del CRR.
- Podremos representar cualquier número n de estas cuatro formas:
  - a) n es un número primo.
  - b) n se representa como  $n = p^k$  con p primo y k entero.
  - c) n es el producto n = p\*q con p y q primos.
  - d) n es un número cualquiera, forma genérica:

$$n = p_1^{e1} * p_2^{e2} * ... * p_t^{et} = \prod_{i=1}^{t} p_i^{ei}$$

### Función $\phi(n)$ de Euler cuando n = p

Caso 1: n es un número primo Si n es primo, φ(n) será igual a CCR menos el 0.

$$\phi(n) = n - 1$$

Si n es primo, entonces CRR = CCR - 1 ya que todos los restos de n, excepto el cero, serán primos entre sí.

Ejemplo 
$$CRR(7) = \{1,2,3,4,5,6\}$$
 seis elementos  $\therefore \phi(7) = n - 1 = 7 - 1 = 6$   
 $\phi(11) = 11 - 1 = 10; \quad \phi(23) = 23 - 1 = 22$ 

Esta expresión se usará en los sistemas de cifra de ElGamal y DSS.

### Función $\phi(n)$ de Euler cuando $n = p^k$

Caso 2:  $n = p^k$  (con p primo y k un entero)

$$\varphi(n)=\varphi(p^k)=p^k\text{ - }p^{k\text{-}1}\qquad \boxed{\varphi(p^k)=p^{k\text{-}1}(p\text{-}1)}$$

De los  $p^k$  elementos del CCR, restaremos todos los múltiplos 1\*p, 2\*p, 3\*p, ... $(p^{k-1}-1)*p$  y el cero.

Ejemplo CRR(16) = 
$$\{1,3,5,7,9,11,13,15\}$$
 ocho elementos  $\therefore \phi(16) = \phi(2^4) = 2^{4-1}(2-1) = 2^3*1 = 8$   $\phi(125) = \phi(5^3) = 5^{3-1}*(5-1) = 5^2*4 = 25*4 = 100$ 

### Función $\phi(n)$ de Euler cuando n = p\*q

Caso 3: n = p\*q (con p y q primos)

$$\phi(n) = \boxed{\phi(p*q) = \phi(p)*\phi(q) = (p-1)(q-1)}$$

De los p\*q elementos del CCR, restaremos todos los múltiplos de p = 1\*p, 2\*p, ... (q - 1)\*p, todos los múltiplos de q = 1\*q, 2\*q, ... (p - 1)\*q y el cero.

$$\phi(p*q) = p*q - [(q-1) + (p-1) + 1] = p*q - q - p + 1$$

$$(p-1)(q-1)$$

Esta expresión se usará en el sistema de cifra RSA.

### Ejemplo de $\phi(n)$ cuando n = p\*q

Ejemplo CRR(15) = 
$$\{1,2,4,7,8,11,13,14\}$$
 ocho elementos  
 $\therefore \phi(15) = \phi(3*5) = (3-1)(5-1) = 2*4 = 8$   
 $\phi(143) = \phi(11*13) = (11-1)(13-1) = 10*12 = 120$ 

Esta será una de las operaciones más utilizadas en criptografía.

Es la base del sistema RSA que durante muchos años ha sido un estándar y, de hecho, continúa siéndolo en el año 2006, al menos a nivel de uso empresarial y comercial.

Uno de sus usos más típicos podemos encontrarlo en las comunicaciones seguras del entorno Internet mediante SSL, tanto para el intercambio de claves como en los formatos de certificados digitales X.509 para firma digital.

### Función φ(n) de Euler para n genérico

Caso 4: 
$$n = p_1^{e1} * p_2^{e2} * ... * p_t^{et}$$
 (p<sub>i</sub> son primos)

$$\phi(n) = \prod_{i=1}^{t} p_i^{ei-1} (p_i - 1)$$

Ejemplo

(Esta demostración no es inmediata)

#### Teorema de Euler

Dice que si mcd  $(a,n) = 1 \implies a^{\phi(n)} \mod n = 1$ Ahora igualamos  $a*x \mod n = 1$  y  $a^{\phi(n)} \mod n = 1$ 

$$\therefore \quad a^{\phi(n)} * a^{-1} \bmod n = x \bmod n$$

$$\therefore$$
  $x = a^{\phi(n)-1} \mod n$ 

El valor x será el inverso de a en el cuerpo n

Nota: Observe que se ha *dividido* por a en el cálculo anterior. Esto se puede hacer porque mcd(a, n) = 1 y por lo tanto hay un único valor inverso en el cuerpo n que lo permite.

#### Cálculo de inversos con Teorema Euler

Ejemplo

¿Cuál es el inverso de 4 en módulo 9?  $\Rightarrow$  inv (4, 9)

Pregunta: ¿Existe  $a * x \mod n = 4 * x \mod 9 = 1$ ?

Como  $mcd(4, 9) = 1 \implies Si \dots aunque 4 y 9 no sean primos.$ 

$$\phi(9) = 6$$
 :  $x = 4^{6-1} \mod 9 = 7$   $\Rightarrow$   $7*4 = 28 \mod 9 = 1$ 

Resulta obvio que: inv (4, 9) = 7 e inv (7, 9) = 4



### Teorema de Euler para n = p\*q

Si el factor a es primo relativo con n y el valor n es el producto de 2 primos, seguirá cumpliéndose el Teorema de Euler también en dichos primos.

#### Por ejemplo:

Si 
$$n = p*q \Rightarrow \phi(n) = (p-1)(q-1)$$
  
 $\forall a / mcd \{a, (p,q)\} = 1$  considering the second of the second o

En el capítulo dedicado a la cifra con clave pública RSA, relacionaremos este tema con el Teorema del Resto Chino.

## Ejemplo Teorema de Euler para n = p\*q

Sea 
$$n = p*q = 7*11 = 77$$
  $\phi(n) = (p-1)(q-1) = (7-1)(11-1) = 6*10 = 60$  Si  $k = 1, 2, 3, ...$  Para  $a = k*7$   $a^{\phi(n)} \mod n = k*7^{60} \mod 77 = 56$  Para  $a = k*11$   $a^{\phi(n)} \mod n = k*11^{60} \mod 77 = 22$  Para  $\forall a \neq k*7, k*11$   $a^{\phi(n)} \mod n = a^{60} \mod 77 = 1$  Y se cumple también que: Para  $\forall a \neq k*7, k*11$   $a^{\phi(n)} \mod p = a^{60} \mod 7 = 1$   $a^{\phi(n)} \mod q = a^{60} \mod 11 = 1$  En caso contrario:  $a^{\phi(n)} \mod p = 0$   $a^{\phi(n)} \mod p = 0$ 

#### Pequeño teorema de Fermat

Si el cuerpo de trabajo es un primo p:  $mcd (a, p) = 1 \implies a^{\phi(p)} mod p = 1$ Entonces  $a * x mod p = 1 y a^{\phi(n)} mod p = 1$ 

Además, en este caso  $\phi(p) = p-1$  por lo que igualando las dos ecuaciones de arriba tenemos:

- $\therefore \quad a^{\phi(p)} * a^{-1} \bmod p = x \bmod p$
- $\therefore$   $x = a^{p-2} \mod p$

Luego x será e inverso de a en el primo p.

### ¿Qué hacemos si no se conoce $\phi(n)$ ?

- Calcular ai mod n cuando los valores de i y a son grandes, se hace tedioso pues hay que utilizar la propiedad de la reducibilidad repetidas veces.
- Si no conocemos φ(n) o no queremos usar los teoremas de Euler o Fermat, siempre podremos encontrar el inverso de a en el cuerpo n usando el

Algoritmo Extendido de Euclides

Este es el método más rápido y práctico

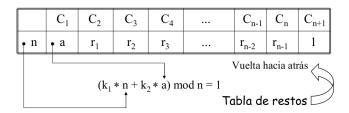
#### Algoritmo Extendido de Euclides AEE

Si mcd (a, n) = 1 y  $a*x mod n = 1 \Rightarrow x = inv (a, n)$ Luego podemos escribir:

Concluye aquí el algoritmo.

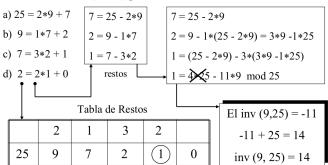
#### Tabla de restos del AEE

Ordenando por restos desde el valor 1 se llega a una expresión del tipo  $(k_1 * n + k_2 * a) \mod n = 1$ , en donde el inverso de a en n lo dará el coeficiente  $k_2$  puesto que  $k_1 * n \mod n = 0$ .



#### Cálculo de inversos mediante el AEE

Encontrar el inv (9, 25) por el método de restos de Euclides.



## Algoritmo para el cálculo de inversos

$$\begin{array}{c} \text{Para encontrar } x = \text{inv } (A,B) \\ \text{Hacer } (g_0,g_1,u_0,u_1,v_0,v_1,i) = (B,A,1,0,0,1,1) \\ \text{Mientras } g_i \neq 0 \text{ hacer} \\ \text{Hacer } y_{i+1} = \text{parte entera } (g_{i-1}/g_i) \\ \text{Hacer } g_{i+1} = g_{i-1} - y_{i+1} * g_i \\ \text{Hacer } u_{i+1} = u_{i-1} - y_{i+1} * u_i \\ \text{Hacer } v_{i+1} = v_{i-1} - y_{i+1} * v_i \\ \text{Hacer } i = i+1 \\ \text{Si } (v_{i-1} < 0) \\ \text{X} = \text{inv } (A,B) \\ x = \text{inv } (9,25) \\ \hline \\ i & y_i & g_i & u_i & v_i \\ 0 & - & 25 & 1 & 0 \\ 1 & - & 9 & 0 & 1 \\ 2 & 2 & 7 & 1 & -2 \\ 3 & 1 & 2 & -1 & = 3 \\ \hline \\ 4 & 3 & 1 & 4 & -11 \\ 5 & 2 & 0 & -9 & 25 \\ \hline \end{array}$$

#### Características de inversos en n = 27

Para el alfabeto castellano con mayúsculas (n = 27) tenemos:

X	inv (x, 27)	X	inv (x, 27)	Χ	inv (x, 27)
1	1	10	19	19	10
2	14	11 —	<b>→</b> 5	20	23
4	7	13	25	22	16
5 —	<b>→</b> 11	14	2	23	20
7	4	16	22	25	13
8	17	17	8	26	26

 $27 = 3^3$  luego no existe inverso para a = 3, 6, 9, 12, 15, 18, 21, 24.

inv 
$$(x, n) = a \iff inv (a, n) = x$$
  
inv  $(1, n) = 1$ ; inv  $(n-1, n) = n-1$ 

Inversos en sistema de cifra clásico orientado a alfabeto de 27 caracteres.

# ¿Qué pasa si mcd $(a, n) \neq 1$ ?

- ¿Pueden existir inversos?
- No, pero...
- Si a \* x mod n = b con b ≠ 1 y mcd (a, n) = m, siendo m divisor de b, habrá m soluciones válidas.

En principio esto no nos sirve en criptografía ...

$$6*x \mod 10 = 4 \mod (6, 10) = 2$$

No existe inv (6, 10) pero ... habrá 2 soluciones válidas

$$x_1 = 4 \implies 6*4 \mod 10 = 24 \mod 10 = 4$$

$$x_2 = 9 \implies 6*9 \mod 10 = 54 \mod 10 = 4$$



- Matemática Discreta
- 2 Algoritmo de Eucliedes
- 3 Inversos en  $\mathbb{Z}_n$
- 4 Función de Euler
- 5 Exponenciación Rápida
- 6 Números Primos

### Un método de exponenciación rápida

- En A<sup>B</sup> mod n se representa el exponente B en binario.
- Se calculan los productos A<sup>2<sup>1</sup></sup> con j = 0 hasta n-1, siendo n el número de bits que representan el valor B en binario.
- Sólo se toman en cuenta los productos en los que en la posición j del valor B en binario aparece un 1.

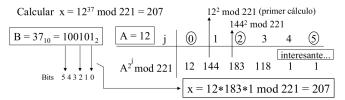
#### Ejemplo

Calcular  $x = 12^{37} \mod 221 = 207$ 

12<sup>37</sup> es un número de 40 dígitos:

8505622499821102144576131684114829934592

# Ejemplo de exponenciación rápida



En vez de 36 multiplicaciones y sus reducciones módulo 221 en cada paso ... 72 operaciones...

Hemos realizado cinco multiplicaciones (para j = 0 el valor es A) con sus reducciones módulo 221, más dos al final y sus correspondientes reducciones; en total 14. Observamos un ahorro superior al 80% pero éste es un valor insignificante dado que los números son muy pequeños.

### Algoritmo de exponenciación rápida

 $Hallar x = A^{B} \bmod n$ 

 Obtener representación binaria del exponente B de k bits:

$$\mathbf{B_2} \! \to b_{k\text{--}1} b_{k\text{--}2} ... b_i ... b_1 b_0$$

- Hacer x = 1
- Para i = k-1, ..., 0 hacer

$$x = x^2 \mod n$$

Si 
$$(b_i = 1)$$
 entonces

$$x = x*A \mod n$$

Ejemplo: calcule  $19^{83} \mod 91 = 24$ 

 $83_{10} = 1010011_2 = b_6b_5b_4b_3b_2b_1b_0$ 

$$x = 1$$
  
 $i=6$   $b_6=1$   $x = 1^2*19 \mod 91$  = 19  $x = 19$   
 $i=5$   $b_5=0$   $x = 19^2 \mod 91$  = 88  $x = 88$   
 $i=4$   $b_4=1$   $x = 88^2*19 \mod 91$  = 80  $x = 80$   
 $i=3$   $b_3=0$   $x = 80^2 \mod 91$  = 30  $x = 30$ 

$$i=2$$
  $b_2=0$   $x=30^2 \mod 91$   $= 81$   $x=81$   
 $i=1$   $b_1=1$   $x=81^2*19 \mod 91$   $= 80$   $x=80$ 

$$i=0$$
  $b_0=1$   $x=80^2*19 \mod 91 = 24$   $x=24$ 

1983 = 1,369458509879505101557376746718e+106 (calculadora Windows). En este caso hemos realizado sólo 16 operaciones frente a 164. Piense ahora qué sucederá en una operación típica de firma digital con hash: (160 bits) (1.024 bits) mod 1,024 bits ©.

# Contingut

- Matemática Discreta
- 2 Algoritmo de Eucliedes
- 3 Inversos en  $\mathbb{Z}_n$
- 4 Función de Euler
- 5 Exponenciación Rápida
- 6 Números Primos

# ¿Cuántos números primos hay?

 Por el teorema de los números primos, se tiene que la probabilidad de encontrar números primos a medida que éstos se hacen más grandes es menor:

#### Números primos en el intervalo $[2, x] = x / \ln x$

```
    Primos entre 2 v 2<sup>5</sup> = 32

                                           x/\ln x = 32/3.46 = 9
                                                                                Probabilidad x sea primo: 30,00 %

    Primos entre 2 v 2<sup>6</sup> = 64

                                           x/\ln x = 64/4, 16 = 15
                                                                                Probabilidad x sea primo: 24,00 %

    Primos entre 2 y 2<sup>7</sup> = 128

                                           x/lnx = 128/4.85 = 26
                                                                               Probabilidad x sea primo: 20,63 %

    Primos entre 2 v 2<sup>8</sup> = 256

                                           x/lnx = 256/5.54 = 46
                                                                                Probabilidad x sea primo: 18,11 %

    Primos entre 2 v 2<sup>9</sup> = 512

                                           x/lnx = 512/6,23 = 82
                                                                                Probabilidad x sea primo: 16,08 %

    Primos entre 2 v 2<sup>10</sup> = 1.024

                                           x/lnx = 1.024/6.93 = 147
                                                                               Probabilidad x sea primo: 14,38 %

    Primos entre 2 v 2<sup>11</sup> = 2.048

                                           x/lnx = 2.048/7.62 = 268
                                                                               Probabilidad x sea primo: 13,10 %

    Primos entre 2 y 2<sup>12</sup> = 4.096

                                           x/lnx = 4.096/8,32 = 492
                                                                                Probabilidad x sea primo: 12,02 %
```

En el capítulo 21 encontrará una tabla con números primos hasta el 1.999.

# Ejemplo del Teorema de los números primos

n	$\pi(n)$	$\frac{n}{ln(n)}$
10	4	4.34
100	25	21.7
1000	168	144.8
$10^{6}$	78498	72382
$10^{9}$	50847478	48254942

#### Test de Primalidad de Fermat

Teorema pequeño de Fermat. Sea p un número primo, entonces  $a^{(p-1)} = 1$  (mod p) para cualquier valor a tal que  $1 \le a \le p$ .

```
def Fermat_test(n,k):
    if n <= 3:
        return str(n)+' es primer'
    for i in range(k):
        a = randint(2,n-2)
        if (a^(n-1))/m != 1:
        return str(n)+' es compost'
    return str(n)+' es primer amb probabilitat '+ str(numerical_approx (1-(1/2)^k))</pre>
```

#### Test de Primalidad de Miller-Rabin

El test de primalidad de Miller-Rabin es un test que combina la condición del teorema pequeño de Fermat con la particularidad de los residuos cuadráticos en aritmética modular.

Given two numbers p and e, where p is an odd prime number and  $e \ge 1$ , then the equation:

$$x^2 \equiv 1 \pmod{p^e}$$

has only the solutions x = 1 and x = -1.

A corollary of this theorem states that if there exists a non-trivial square root of  $1 \pmod{n}$ , that is:

$$x^2 \equiv 1 \pmod{n}$$

then n is composite.

On the basis of this corollary, the Miller-Rabin test calculates each modular exponentiation and checks if there's a non-trivial square root of  $1 \pmod{n}$ . In this case, the test ends with the COMPOSITE result.

The Miller-Rabin test is a probabilistic search for proof that n is composite.

#### Carlos Borrego

Carlos.Borrego@uab.cat

Departament d'Enginyeria de la Informació i de les Comunicacions Universitat Autònoma de Barcelona

Criptografia i Seguretat