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

Material adaptat de:
Curso de Seguridad Informática y Criptografía

Dr. Jorge Ramió Aguirre

Universidad Politécnica de Madrid

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

Contingut

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_n) si y sólo si existe algún entero k que divide de forma exacta la diferencia (a - b) .
  - Esto podemos expresarlo así:

$$\boxed{\begin{array}{c} a - b = k * n \\ a \equiv_n b \\ a \equiv b \bmod n \end{array}}$$

Propiedades de la congruencia en Z_n

- Propiedad Reflexiva:

$$a \equiv a \bmod n \quad \forall\, a \in Z$$

- Propiedad Simétrica:

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

- Propiedad Transitiva:

$$\text{Si} \quad a \equiv b \bmod n \quad \text{y} \quad b \equiv c \bmod n$$
$$\Rightarrow a \equiv c \bmod n \quad \forall\, a,b,c \in Z$$

Propiedades de las operaciones en Z_n (1)

- Propiedad Asociativa:

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

- Propiedad Conmutativa:

  $$a + b \bmod n \equiv b + a \bmod n$$

  $$a * b \bmod n \equiv b * a \bmod n$$

- Propiedad Distributiva:

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

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

$$\boxed{a * (b+c) \bmod n = ((a * b) + (a * c)) \bmod n}$$

Propiedades de las operaciones en Z_n (2)

- Existencia de Identidad:

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

$$a * 1 \bmod n = 1 * a \bmod n = a \bmod n = a$$

- Existencia de Inversos: ✓

✓ Ambos serán muy importantes en criptografía

$$a + (-a) \bmod n = 0$$

$$a * (a^{-1}) \bmod n = 1 \text{ (si } a \neq 0) \longrightarrow \text{No siempre existe}$$

- Reducibilidad: ✓

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

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

Conjunto completo de restos CCR

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

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

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

Enteros

Enteros mod n

$a_1, a_2 \longrightarrow (a_1 \bmod n), (a_2 \bmod n)$

es lo mismo que

Esta operación ...

esta otra

op (y posterior reducción mod n)

op

$(a_1 \text{ op } a_2) \bmod n$

$(a_1 \bmod n) \text{ op } (a_2 \bmod n) \bmod n$

Esto nos permitirá trabajar con números muy grandes

Un ejemplo de homomorfismo

$88 * 93 \bmod 13$

$8.184 \bmod 13$

Resultado: 7

Se desbordaría la memoria de nuestro sistema ☹

Ejemplo: una calculadora capaz de trabajar sólo con tres dígitos ...

Solución por homomorfismo:

$88 * 93 \bmod 13$

$[(88) \bmod 13 * (93) \bmod 13] \bmod 13$

$10 * 2 \bmod 13$

$20 \bmod 13$ Resultado: 7

se llega a lo mismo, pero...

Ahora ya no se desborda la memoria ☺

... y hemos usado siempre números de 3 dígitos. En este caso la operación máxima sería $12*12 = 144$, es decir tres dígitos.

Divisibilidad de los números

En criptografía muchas veces nos interesará encontrar el máximo común denominador mcd 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 $\Rightarrow$ $a = \mathrm{x} * a'$ y $b = \mathrm{x} * b'$
- b) Por lo tanto: $a - \mathrm{k} * b = \mathrm{x} * a' - \mathrm{k} * \mathrm{x} * b'$

  $a - \mathrm{k} * b = \mathrm{x}\,(a' - \mathrm{k} * b')$
- c) Entonces se concluye que x divide a $(a - k * b)$

El máximo común denominador mcd

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) $(a > b > r \geq 0)$

porque:

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

(con r un entero y r' un resto)

Contingut

Divisibilidad con algoritmo de Euclides

mcd (148, 40)

$148 = 3 * 40 + 28$

$40 = 1 * 28 + 12$

$28 = 2 * 12 + 4$

$12 = 3 * 4 + 0$

mcd (148, 40) = 4

$148 = 2^2 * 37$

$40 = 2^3 * 5$

Factor común

$2^2 = 4$

mcd (385, 78)

$385 = 4 * 78 + 73$

$78 = 1 * 73 + 5$

$73 = 14 * 5 + 3$

$5 = 1 * 3 + 2$

$3 = 1 * 2 + 1$

$2 = 2 * 1 + 0$

mcd (385, 78) = 1

No hay factor común

$385 = 5 * 7 * 11$

$78 = 2 * 3 * 13$

Esta condición será importante en criptografía. ☞

Contingut

Inversos en Z_n

$$\text{Si} \quad a * x \equiv 1 \bmod n$$

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_n. Por ejemplo, si n = 6, en Z_6 no existe el inverso del 2, pues la ecuación $2*x \equiv 1 \bmod 6$ no tiene solución.
- Si n es un número primo p, entonces todos los elementos de Z_p salvo el cero tienen inverso. Por ejemplo, en Z_5 se tiene que:

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

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.

$$\text{mcd}(a, n) = 1 \quad \Rightarrow \quad \exists\, x\, !\;\; 0 < x < n \;\; / \; a * x \bmod n = 1$$

Sea: a = 4 y n = 9. Valores de i = {1, 2, 3, 4, 5, 6, 7, 8}

SOLUCIÓN ÚNICA

4∗1 mod 9 = 4	4∗2 mod 9 = 8	4∗3 mod 9 = 3
4∗4 mod 9 = 7	4∗5 mod 9 = 2	4∗6 mod 9 = 6
4∗7 mod 9 = 1	4∗8 mod 9 = 5	

Si mcd (a,n) ≠ 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 $0 < x < n$ / $a * x \bmod n = 1$

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

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

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

No existe el inverso para ningún resto del cuerpo.

Inversos aditivo y multiplicativo

(A+B) mod 5

B +	0	1	2	3	4
A 0	0	1	2	3	4
1	1	2	3	4	<u>0</u>
2	2	3	4	<u>0</u>	1
3	3	4	<u>0</u>	1	2
4	4	<u>0</u>	1	2	3

0+0 = 0
1∗1 = 1
Es trivial

(A∗B) mod 5

B ∗	0	1	2	3	4
A 0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	<u>1</u>	3
3	0	3	<u>1</u>	4	2
4	0	4	3	2	<u>1</u>

- En la operación suma siempre existirá el inverso o valor identidad de la adición (0) para cualquier resto del cuerpo. Su valor es único.
- 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 $\{0, 1, \ldots n_i, \ldots n-1\}$ 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:

$$CRR = \{1, \ldots, n_i, \ldots n-1\} \ / \ mcd(n_i, n) = 1$$

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

Función de Euler $\phi(n)$

- El Indicador o Función de Euler $\phi(n)$ 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:

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

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

Caso 1: n es un número primo

Si n es primo, $\phi(n)$ será igual a CCR menos el 0.

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

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

$$\therefore \phi(7) = n - 1 = 7\text{-}1 = 6$$

$$\phi(11) = 11\text{-}1 = 10; \quad \phi(23) = 23\text{-}1 = 22$$

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)

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

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

$$\phi(p*q) = p*q - [(q-1) + (p-1) + 1] = \underbrace{p*q - q - p + 1}_{(p-1)(q-1)}$$

Esta expresión se usará en el sistema de cifra RSA.

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

Ejemplo

$\Rightarrow$

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 $\phi(n)$ de Euler para n genérico

Caso 4: $n = p_1^{e1} * p_2^{e2} * \ldots * p_t^{et}$ (p_i son primos)

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

Ejemplo

(Esta demostración no es inmediata)

$CRR(20) = \{1, 3, 7, 9, 11, 13, 17, 19\}$ ocho elementos

$\therefore \phi(20) = \phi(2^2 * 5) = 2^{2-1}(2-1) * 5^{1-1}(5-1) = 2^1 * 1 * 1 * 4 = 8$

$\phi(360) = \phi(2^3 * 3^2 * 5) = 2^{3-1}(2-1) * 3^{2-1}(3-1) * 5^{1-1}(5-1) = 96$

Teorema de Euler

Dice que si mcd (a,n) = 1 $\Rightarrow$ $a^{\phi(n)}$ mod n = 1

Ahora igualamos $a * x$ mod n = 1 y $a^{\phi(n)}$ mod n = 1

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

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

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 $a * x \bmod n = 4 * x \bmod 9 = 1$?

Como mcd (4, 9) = 1 $\Rightarrow$ Sí ... aunque 4 y 9 no sean primos.

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

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 $\{a, (p,q)\} = 1$

se cumple que:

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

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

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 $k = 1, 2, 3, \ldots$

Para $a = k*7$ $\quad a^{\phi(n)} \bmod n = k*7^{60} \bmod 77 = 56$

Para $a = k*11$ $\quad a^{\phi(n)} \bmod n = k*11^{60} \bmod 77 = 22$

Para $\forall\ a \neq k*7, k*11$ $\quad a^{\phi(n)} \bmod n = a^{60} \bmod 77 = 1$

Y se cumple también que:

Para $\forall\ a \neq k*7, k*11$ $\quad a^{\phi(n)} \bmod p = a^{60} \bmod 7 = 1$

$a^{\phi(n)} \bmod q = a^{60} \bmod 11 = 1$

En caso contrario: $\quad a^{\phi(n)} \bmod p = 0$

$a^{\phi(n)} \bmod q = 0$

Pequeño teorema de Fermat

Si el cuerpo de trabajo es un primo p:

$$\text{mcd}(a, p) = 1 \quad \Rightarrow \quad a^{\phi(p)} \bmod p = 1$$

$$\text{Entonces} \quad a * x \bmod p = 1 \quad y \quad a^{\phi(n)} \bmod p = 1$$

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

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

$$\therefore \quad x = a^{p-2} \bmod p$$

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

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

- Calcular a^i 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 $\phi(n)$ 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 $a*x \bmod n = 1 \Rightarrow x = \text{inv}(a, n)$

Luego podemos escribir:

$$
\begin{array}{ll}
n = C_1 * a + r_1 & a > r_1 \\
a = C_2 * r_1 + r_2 & r_1 > r_2 \\
r_1 = C_3 * r_2 + r_3 & r_2 > r_3 \\
\ldots & \ldots \\
r_{n-2} = \boxed{C_n * r_{n-1} + 1 \quad r_{n-1} > 1} & \\
r_{n-1} = C_{n+1} * 1 + 0 &
\end{array}
$$

Si volvemos hacia atrás desde este valor, obtenemos el inverso de a en el cuerpo n.

Concluye aquí el algoritmo.

Tabla de restos del AEE

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

	C_1	C_2	C_3	C_4	...	C_{n-1}	C_n	C_{n+1}
n	a	r_1	r_2	r_3	...	r_{n-2}	r_{n-1}	1

Vuelta hacia atrás

$$(k_1 * n + k_2 * a) \bmod n = 1$$

Tabla de restos

Cálculo de inversos mediante el AEE

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

a) $25 = 2*9 + 7$

b) $9 = 1*7 + 2$

c) $7 = 3*2 + 1$

d) $2 = 2*1 + 0$

$7 = 25 - 2*9$

$2 = 9 - 1*7$

$1 = 7 - 3*2$

restos

$7 = 25 - 2*9$

$2 = 9 - 1*(25 - 2*9) = 3*9 - 1*25$

$1 = (25 - 2*9) - 3*(3*9 - 1*25)$

$1 = \cancel{4*25} - 11*9 \mod 25$

Tabla de Restos

	2	1	3	2	
25	9	7	2	1	0

El inv (9,25) = -11

-11 + 25 = 14

inv (9, 25) = 14

Algoritmo para el cálculo de inversos

Para encontrar $x = \text{inv}(A, B)$

Hacer $(g_0, g_1, u_0, u_1, v_0, v_1, i) = (B, A, 1, 0, 0, 1, 1)$

Mientras $g_i \neq 0$ hacer

- Hacer $y_{i+1} = \text{parte entera}(g_{i-1}/g_i)$
- Hacer $g_{i+1} = g_{i-1} - y_{i+1} * g_i$
- Hacer $u_{i+1} = u_{i-1} - y_{i+1} * u_i$
- Hacer $v_{i+1} = v_{i-1} - y_{i+1} * v_i$
- Hacer $i = i+1$

Si $(v_{i-1} < 0)$ $\boxed{x = \text{inv}(9, 25) = -11+25 = 14}$

- Hacer $v_{i-1} = v_{i-1} + B$
- Hacer $x = v_{i-1}$

Ejemplo

$\boxed{\begin{array}{l} x = \text{inv}(A, B) \\ x = \text{inv}(9, 25) \end{array}}$

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
4	3	1	4	-11
5	2	0	-9	25

Características de inversos en n = 27

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

x	inv (x, 27)	x	inv (x, 27)	x	inv (x, 27)
1	1	10	19	19	10
2	14	11	5	20	23
4	7	13	25	22	16
5	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 $\Leftrightarrow$ 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) ≠ 1?

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

En principio esto no nos sirve en criptografía ...

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

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

$x_1 = 4 \Rightarrow 6*4 \bmod 10 = 24 \bmod 10 = 4$

$x_2 = 9 \Rightarrow 6*9 \bmod 10 = 54 \bmod 10 = 4$

?

Contingut

Un método de exponenciación rápida

- En A^B mod n se representa el exponente B en binario.
- Se calculan los productos A^{2^j} 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 $x = 12^{37} \bmod 221 = 207$

12^{37} es un número de 40 dígitos:

8505622499821102144576131684114829934592

Ejemplo de exponenciación rápida

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

$12^2 \bmod 221$ (primer cálculo)

$144^2 \bmod 221$

$\boxed{B = 37_{10} = 100101_2}$ $\boxed{A = 12}$

Bits 5 4 3 2 1 0

j	0	1	2	3	4	5
$A^{2^j} \bmod 221$	12	144	183	118	1	1

interesante...

$\boxed{x = 12 * 183 * 1 \bmod 221 = 207}$

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:

  $B_2 \rightarrow b_{k-1}b_{k-2}\ldots b_i\ldots b_1 b_0$
- Hacer $x = 1$
- Para $i = k-1, \ldots, 0$ hacer

  $x = x^2 \bmod n$

  Si ($b_i = 1$) entonces

  $x = x*A \bmod n$

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

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

$x = 1$

$i=6$	$b_6=1$	$x = 1^2*19 \bmod 91$	$= 19$	$x = 19$
$i=5$	$b_5=0$	$x = 19^2 \bmod 91$	$= 88$	$x = 88$
$i=4$	$b_4=1$	$x = 88^2*19 \bmod 91$	$= 80$	$x = 80$
$i=3$	$b_3=0$	$x = 80^2 \bmod 91$	$= 30$	$x = 30$
$i=2$	$b_2=0$	$x = 30^2 \bmod 91$	$= 81$	$x = 81$
$i=1$	$b_1=1$	$x = 81^2*19 \bmod 91$	$= 80$	$x = 80$
$i=0$	$b_0=1$	$x = 80^2*19 \bmod 91$	$= 24$	$x = 24$

19^{83} = 1,3694585098795051015573767467l8e+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) $^{(1.024 \text{ bits})}$ mod 1.024 bits ☺.

Contingut

¿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 y 2^5 = 32	x/lnx = 32/3,46 = 9	Probabilidad x sea primo: 30,00 %
• Primos entre 2 y 2^6 = 64	x/lnx = 64/4,16 = 15	Probabilidad x sea primo: 24,00 %
• Primos entre 2 y 2^7 = 128	x/lnx = 128/4,85 = 26	Probabilidad x sea primo: 20,63 %
• Primos entre 2 y 2^8 = 256	x/lnx = 256/5,54 = 46	Probabilidad x sea primo: 18,11 %
• Primos entre 2 y 2^9 = 512	x/lnx = 512/6,23 = 82	Probabilidad x sea primo: 16,08 %
• Primos entre 2 y 2^{10} = 1.024	x/lnx = 1.024/6,93 = 147	Probabilidad x sea primo: 14,38 %
• Primos entre 2 y 2^{11} = 2.048	x/lnx = 2.048/7,62 = 268	Probabilidad x sea primo: 13,10 %
• Primos entre 2 y 2^{12} = 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 <= a < 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))%n != 1:
            return str(n)+' es compost'
    return str(n)+' es primer amb probabilitat '+ str(numerical_approx
    (1-(1/2)^k))
```

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.

```
def Miller_Rabin_test(n,k):
    tmp = n-1
    s = 0
    while tmp%2 == 0:
        tmp = tmp // 2
        s= s +1
    r = (n-1) / (2^s)
    for i in range(k):
        a = randint(2,n-2)
        y = a^r%n
        if (y != 1) and (y != n-1):
            j = 1
            while (j >= (s-1)) and (y != (n-1)):
                y = (y^2)%n
                if y==1:
                    return str(n)+' es compost'
                j = j+1
            if y != n-1:
                return str(n)+' es compost'
    return str(n)+' es primer amb probabilitat '+ str(numerical_approx
    (1-(1/4)^k))
```

Given two numbers p and e, where p is an odd prime number and $e \geq 1$, 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.

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