



**Universidad de las
Ciencias Informáticas**

**Especialidad Seguridad
Informática**

**ENTRENAMIENTO EN ELEMENTOS DE
CRIPTOGRAFÍA**

**25 de junio del 2018.
Año 60 de la Revolución"**

Elementos de la Teoría de números

Sumario

1. Congruencias. Propiedades de las congruencias
2. Operaciones en \mathbb{Z}_n .
3. Conjunto completo de restos CCR
4. Divisibilidad de los números. Algoritmo de Euclides. AEE
5. Inversos en \mathbb{Z}_n
6. Conjunto reducido de restos CRR
7. Función de Euler $\phi(n)$
8. Teorema de Euler
9. Pequeño teorema de Fermat
10. Teorema chino de los restos
11. La exponenciación en la cifra asimétrica. Algoritmo de exponenciación rápida.
12. Generadores de un cuerpo primo.
13. Cálculos en campos de Galois.

Objetivos

1. Estudiar elementos básicos de la Criptografía, la aritmética modular y la teoría de la información.

Dra.C. Oristela Cuellar Justiz oristelacj@uci.cu
Facultad CITEC. Centro de Estudio de Matemática Computacional (CEMC)
Dpto. Bioinformática

Congruencia

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

$$a - b = k * n$$

$$a \equiv_n b$$

$$a \equiv b \pmod{n}$$

Propiedades de la congruencia en \mathbb{Z}_n

- Propiedad Reflexiva:

$$a \equiv a \pmod{n} \quad \forall a \in \mathbb{Z}$$

- Propiedad Simétrica:

$$a \equiv b \pmod{n} \Rightarrow b \equiv a \pmod{n} \quad \forall a, b \in \mathbb{Z}$$

- Propiedad Transitiva:

$$\begin{aligned} \text{Si } a \equiv b \pmod{n} \text{ y } b \equiv c \pmod{n} \\ \Rightarrow a \equiv c \pmod{n} \quad \forall a, b, c \in \mathbb{Z} \end{aligned}$$

Operaciones de congruencia en Z_n

¿Es 18 congruente con 3 módulo 5?

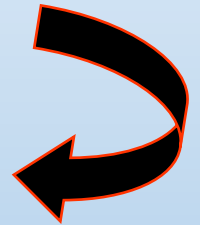
$$¿18 \equiv 3 \pmod{5}?$$

Sí, porque: $18 - 3 = 15 = k * 5$ con $k = 3$

¿Cómo se usará esto en criptografía?

Esta operación en Z_n se expresará así:

$$18 \pmod{5} = 3$$



El valor 3 será el **resto o residuo**.

El conjunto de números que forman los restos dentro de un cuerpo Z_n será muy importante en criptografía.

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 \equiv que denotaba congruencia. Esto es algo propio de los Campos de Galois que veremos más adelante.

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

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:



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

$$a * (a^{-1}) \bmod n = 1 \text{ (si } a \neq 0 \text{)}$$

✓ Ambos serán
muy importantes
en criptografía

- Reducibilidad:



→ No siempre existe

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

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

Conjunto completo de restos CCR

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

$$\forall a \in \mathbb{Z} \quad \exists ! r_i \in CCR / a \equiv r_i \pmod{n}$$

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

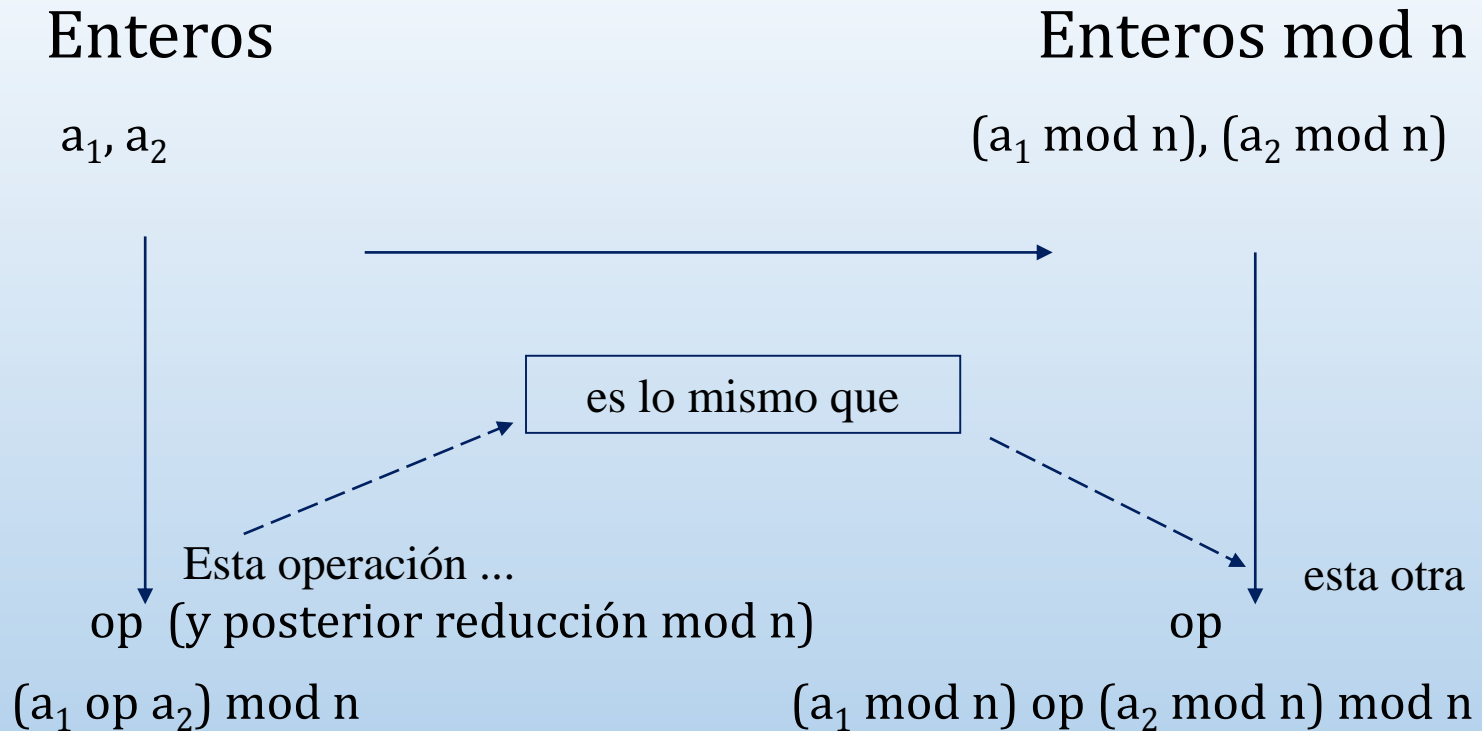


$$CCR(6) = \{0, 1, 2, 3, 4, 5\} = \{12, 7, 20, 9, 16, 35\}$$

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

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

$$88 * 93 \bmod 13$$

$$8.184 \bmod 13$$

Resultado: 7

Se desbordaría
la memoria de
nuestro sistema

Ahora ya
no se
desborda la
memoria



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

Solución por homomorfismo:

$$88 * 93 \bmod 13$$

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

$$10 * 2 \bmod 13$$

$$20 \bmod 13 \quad \text{Resultado: 7}$$

se llega a lo mismo, pero...

... 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 divisor **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. $\Rightarrow \text{mcd}(a, n) = 1$

Algoritmo de Euclides:

- a) Si x divide a a y $b \Rightarrow a = x * a' \text{ y } b = x * b'$
- 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 $(a - k * b)$

Máximo común divisor mcd

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

$$\textit{Si } a > b \textit{ entonces } a = d_1 * b + r$$

(con d_1 un entero y r un resto)

$$\text{Luego } \text{mcd}(a, b) = \text{mcd}(b, r) \quad (a > b > r \geq 0)$$

porque:

$$\text{Si } b > r \textit{ entonces } b = d_2 * r + r'$$

(con r un entero y r' un resto)

Divisibilidad con algoritmo de Euclides

$$\text{mcd}(148, 40)$$

$$148 = 3 * 40 + 28$$

$$40 = 1 * 28 + 12$$

$$28 = 2 * 12 + 4$$

$$12 = 3 * 4 + 0$$

$$\text{mcd}(148, 40) = 4$$

Esta condición
será importante
en criptografía.



$$148 = 2^2 * 37$$

$$40 = 2^3 * 5$$

Factor común
 $2^2 = 4$

No hay
factor común

$$385 = 5 * 7 * 11$$

$$78 = 2 * 3 * 13$$

$$\text{mcd}(385, 78)$$

$$385 = 4 * 78 + 73$$

$$78 = 1 * 73 + 5$$

$$73 = 14 * 5 + 3$$

$$5 = 1 * 3 + 2$$

$$3 = 1 * 2 + 1$$

$$2 = 2 * 1 + 0$$

$$\text{mcd}(385, 78) = 1$$

Inversión de una operación de cifra

- En criptografía deberá estar permitido invertir una operación para recuperar un cifrado \Rightarrow descifrar.
- Aunque la cifra es una función, en lenguaje coloquial la operación de cifrado podría interpretarse como una “multiplicación” y la operación de descifrado como una “división”, si bien hablaremos en este caso de una multiplicación por el inverso.
- La analogía anterior sólo será válida en el cuerpo de los enteros \mathbb{Z}_n con inverso.
- Luego, si en una operación de cifra la función es el valor a dentro de un cuerpo n , deberemos encontrar el inverso $a^{-1} \bmod n$ para descifrar; en otras palabras ...

Inversos en \mathbb{Z}_n

Si $a * x \equiv 1 \pmod{n}$

se dice que x es el inverso multiplicativo de a en \mathbb{Z}_n y se denotará por a^{-1} .

- No siempre existen el inverso de un elemento en \mathbb{Z}_n . Por ejemplo, si $n = 6$, en \mathbb{Z}_6 no existe el inverso del 2, pues la ecuación $2 * x \equiv 1 \pmod{6}$ no tiene solución.
- Si n es un número primo p , entonces todos los elementos de \mathbb{Z}_p salvo el cero tienen inverso. Por ejemplo, en \mathbb{Z}_5 se tiene que:

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

Existencia del inverso por primalidad

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

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

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

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

$$4 * 1 \text{ mod } 9 = 4 \quad 4 * 2 \text{ mod } 9 = 8 \quad 4 * 3 \text{ mod } 9 = 3$$

$$4 * 4 \text{ mod } 9 = 7 \quad 4 * 5 \text{ mod } 9 = 2 \quad 4 * 6 \text{ mod } 9 = 6$$

$$4 * 7 \text{ mod } 9 = 1 \quad 4 * 8 \text{ mod } 9 = 5$$

S
O
L
U
C
I
Ó
N

Ú
N
I
C
A

Inexistencia de inverso (no primalidad)

¿Y si no hay primalidad entre a y n ?

Si $\text{mcd}(a, n) \neq 1$

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

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

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

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

No existe el inverso para ningún resto del cuerpo.

Inversos aditivo y multiplicativo

$$(A+B) \bmod 5$$

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

$$(A*B) \bmod 5$$

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.

Conjunto reducido de restos CRR

- El conjunto reducido de restos, conocido como CRR de n , es el subconjunto $\{0, 1, \dots, n_i, \dots, n-1\}$ de restos, primos con n .
- Si n es primo, todos los restos serán primos con él.
- Como el cero no es una solución, entonces:

$$\text{CRR} = \{1, \dots, n_i, \dots, n-1\} \quad / \quad \text{mcd}(n_i, n) = 1$$

$$\text{Ejemplo: } \text{CRR mod } 8 = \{1, 3, 5, 7\}$$

$$\text{CRR mod } 5 = \{1, 2, 3, 4\}$$

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 campo 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 y más aún en los sistemas asimétricos y en particular el RSA ya que los cálculos de claves pública y privada se harán dentro del campo $\phi(n)$. En ambos casos la cifra y las claves estarán relacionadas con el CRR.



Función de Euler $\phi(n)$

- La 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:
$$n = p_1^{e_1} * p_2^{e_2} * p_3^{e_3} * \dots * p_t^{e_t} = \prod_{i=1}^t p_i^{e_i}$$

Función $\phi(n)$ de Euler

Caso 1: n es un número primo (p)

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

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

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 el sistema de cifra de El Gamal.

Función $\phi(n)$ de Euler

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

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

Ejemplo

$\text{CRR}(16) = \{1, 3, 5, 7, 9, 11, 13, 15\}$ ocho elementos

$$\therefore \phi(16) = \phi(2^4) = 2^{4-1}(2-1) = 2^3 * 1 = 8$$

$$\phi(125) = \phi(5^3) = 5^{3-1} * (5-1) = 5^2 * 4 = 25 * 4 = 100$$

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

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

Ejemplo

$\text{CRR}(15) = \{1, 2, 4, 7, 8, 11, 13, 14\}$ ocho elementos

$$\therefore \phi(15) = \phi(3 * 5) = (3-1)(5-1) = 2 * 4 = 8$$

$$\phi(143) = \phi(11 * 13) = (11-1)(13-1) = 10 * 12 = 120$$

La función de Euler $\phi(n)$ cuando $n = p*q$

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

- 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, 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: $n = p_1^{e_1} * p_2^{e_2} * p_3^{e_3} * \dots * p_t^{e_t}$ (p_i son primos)

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

Ejemplo

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

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

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

Teorema de Euler

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

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

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

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

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 $\text{mcd}(a, n) = 1$ y por lo tanto hay un único valor inverso en el cuerpo Z_n que lo permite.

Cálculo de inversos con Teorema Euler

Ejemplo

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

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

$$\text{mcd}(4, 9) = 1$$

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

Resulta obvio que: $\text{inv}(4, 9) = 7$ e $\text{inv}(7, 9) = 4$

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:

$$\text{Si } n = p*q \Rightarrow \phi(n) = (p - 1)(q - 1) \\ \forall a / \text{mcd} \{a, (p, q)\} = 1$$

se cumple que:

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

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

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

Pequeño teorema de Fermat

Si el cuerpo de trabajo es un primo p :

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

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

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

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

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

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

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

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

Luego podemos escribir:

$$n = C_1 * a + r_1 \quad a > r_1$$

$$a = C_2 * r_1 + r_2 \quad r_1 > r_2$$

$$r_1 = C_3 * r_2 + r_3 \quad r_2 > r_3$$

...

...

$$r_{n-2} = C_n * r_{n-1} + 1 \quad r_{n-1} > 1$$

$$r_{n-1} = C_{n+1} * 1 + 0$$

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

El número de pasos del algoritmo de Euclides, es a lo más 5 veces el n° de dígitos del número más pequeño

Concluye aquí el algoritmo.

Tabla de restos del AEE

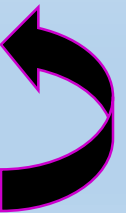
Ordenando por restos desde el valor 1 se llega a una expresión del tipo $(k_1 * n + k_2 * a) \bmod n = 1$, en donde el inverso de a en n lo dará el coeficiente k_2 puesto que $k_1 * n \bmod n = 0$.

	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


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

Vuelta hacia atrás

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

$$7 = 25 - 2*9$$

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

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

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

Tabla de Restos

	2	1	3	2	
25	9	7	2	1	0

El inv (9,25) = -11

$$-11 + 25 = 14$$

$$\text{inv}(9, 25) = 14$$

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

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

Hacer $x = v_{i-1}$



$x = \text{inv}(A, B)$
 $x = \text{inv}(9, 25)$

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

Ejemplo

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 $a = 3, 6, 9, 12, 15, 18, 21, 24$.

$$\text{inv}(x, n) = a \Leftrightarrow \text{inv}(a, n) = x$$

$$\text{inv}(1, n) = 1; \text{inv}(n-1, n) = n-1$$

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

¿Qué pasa si $\text{mcd}(a, n) \neq 1$?

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

En principio esto **no nos sirve** en criptografía ...



$$6 * x \bmod 10 = 4 \quad \text{mcd}(6, 10) = 2$$

No existe $\text{inv}(6, 10)$ pero ... habrá 2 soluciones válidas

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

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



Teorema del Resto Chino TRC

Si $n = d_1 * d_2 * d_3 * \dots * d_t$ con $d_i = p_i^{e_i}$ (p primo)

El sistema de ecuaciones:

$$x \bmod d_i = x_i \quad (i = 1, 2, 3, \dots t)$$

tiene una solución común en $[0, n-1]$

$$x = \sum_{i=1}^t \frac{n}{d_i} * y_i * x_i \bmod n$$

con $y_i = \text{inv} [(n/d_i), d_i]$

En algunos textos lo
verá como “Teorema
Chino de los Restos”....

Algunas aplicaciones interesantes en : el sistema de cifra RSA y Protocolos Criptográficos.

Ejemplo de aplicación del TRC (1)

Encontrar x de forma que : $12 * x \bmod 3960 = 36$

Tenemos la ecuación genérica: $a * x_i \bmod d_i = b$

$$n = 3960 \Rightarrow n = 2^3 * 3^2 * 5 * 11 = d_1 * d_2 * d_3 * d_4 = 8 * 9 * 5 * 11$$

$$a = 12$$

$$b = 36$$

Como $n \Rightarrow d_4$, existirán 4 soluciones de x_i

$$a * x_1 \bmod d_1 = b \bmod d_1 \quad 12 * x_1 \bmod 8 = 36 \bmod 8 = 4$$

$$a * x_2 \bmod d_2 = b \bmod d_2 \quad 12 * x_2 \bmod 9 = 36 \bmod 9 = 0$$

$$a * x_3 \bmod d_3 = b \bmod d_3 \quad 12 * x_3 \bmod 5 = 36 \bmod 5 = 1$$

$$a * x_4 \bmod d_4 = b \bmod d_4 \quad 12 * x_4 \bmod 11 = 36 \bmod 11 = 3$$

4 ecuaciones en x

Resolviendo para x_i

Ejemplo de aplicación del TRC (2)

$$\begin{array}{ll} x_1 = 1 & x_2 = 0 \\ x_3 = 3 & x_4 = 3 \end{array}$$

4 ecuaciones en x

$$12*x_1 \bmod 8 = 4 \Rightarrow 4*x_1 \bmod 8 = 4 \Rightarrow x_1 = 1$$

$$12*x_2 \bmod 9 = 0 \Rightarrow 3*x_2 \bmod 9 = 0 \Rightarrow x_2 = 0$$

$$12*x_3 \bmod 5 = 1 \Rightarrow 2*x_3 \bmod 5 = 1 \Rightarrow x_3 = 3$$

$$12*x_4 \bmod 11 = 3 \Rightarrow 1*x_4 \bmod 11 = 3 \Rightarrow x_4 = 3$$

Ejemplo de aplicación del TRC (3)

ExtendedGCD[x,y]={g,{a,b}} \Rightarrow $g = ax + by$

Cuando $g=1 \Rightarrow 1=ax+by$ como $by \equiv 0 \pmod{y} \Rightarrow \text{inv}(x,y)=a$

ExtendedGCD[495,8]={1,{-1,62}} $\Rightarrow y_1 = \text{inv}(495,8) = -1+8=7$

ExtendedGCD[440, 9]={1, {-1, 49}} $\Rightarrow y_2 = \text{inv}[440, 9] = -1+9=8$

ExtendedGCD[792, 5]={1, {-2, 317}} $\Rightarrow y_3 = \text{inv}(792,5) = -2+5=3$

ExtendedGCD[360, 11] = {1, {-4, 131}} $\Rightarrow y_4 = \text{inv}(360, 11) = -4+11=7$

Ejemplo de aplicación del TRC (4)

$$x_1 = 1 \quad x_2 = 0$$

$$x_3 = 3 \quad x_4 = 3$$

$$y_1 = 7 \quad y_2 = 8$$

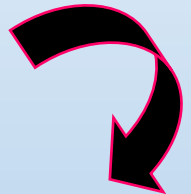
$$y_3 = 3 \quad y_4 = 7$$

Aplicando ecuación del Resto Chino para el caso

$$12 * x \bmod 3960 = 36$$

con $d_1 = 8, d_2 = 9, d_3 = 5, d_4 = 11$:

$$x = \sum_{i=1}^t \frac{n}{d_i} * y_i * x_i \bmod n$$



$$x = [(n/d_1)y_1x_1 + (n/d_2)y_2x_2 + (n/d_3)y_3x_3 + (n/d_4)y_4x_4]$$

$$x = [495*7*1 + 440*8*0 + 792*3*3 + 360*7*3] \bmod 3\,960$$

$$x = [3465 + 0 + 7\,128 + 7\,560] \bmod 3960 = 2\,313$$

La exponenciación en la cifra asimétrica

- Una de las aplicaciones más interesantes de la matemática discreta en la criptografía es en la cifra asimétrica en la que la operación básica es una exponenciación $A^B \bmod n$, en donde n es un primo grande o un producto de primos grandes.
- Esta operación $A^B \bmod n$ se realizará para el intercambio de clave y en la firma digital.
- ¿Cómo hacer estos cálculos de forma rápida y eficiente, sin tener que aplicar reducibilidad? Los algoritmos de exponenciación rápida serán la solución.

Un método de exponenciación rápida

- En $A^B \bmod n$ 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

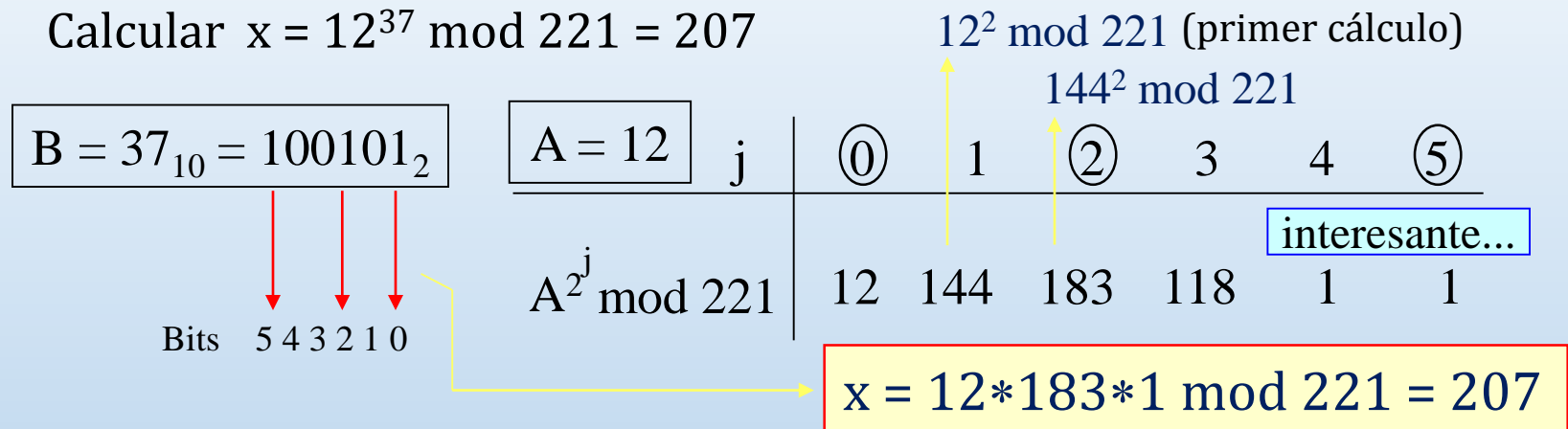
Calcular $x = 1237 \bmod 221 = 207$

12^{37} es un número de 40 dígitos:

8505622499821102144576131684114829934592

Ejemplo de exponenciación rápida

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



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:

$$B_2 \rightarrow b_{k-1}b_{k-2}\dots b_i\dots b_1b_0$$

- Hacer $x = 1$
- Para $i = k-1, \dots, 0$ hacer
 $x = x^2 \bmod n$
Si $(b_i = 1)$ entonces
 $x = x * A \bmod n$

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

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

$$x = 1$$

$$i=6 \quad b_6=1 \quad x = 1^2 * 19 \bmod 91 = 19 \quad x = 19$$

$$i=5 \quad b_5=0 \quad x = 19^2 \bmod 91 = 88 \quad x = 88$$

$$i=4 \quad b_4=1 \quad x = 88^2 * 19 \bmod 91 = 80 \quad x = 80$$

$$i=3 \quad b_3=0 \quad x = 80^2 \bmod 91 = 30 \quad x = 30$$

$$i=2 \quad b_2=0 \quad x = 30^2 \bmod 91 = 81 \quad x = 81$$

$$i=1 \quad b_1=1 \quad x = 81^2 * 19 \bmod 91 = 80 \quad x = 80$$

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

$19^{83} = 1,369458509879505101557376746718 \text{ e}+106$ (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: $(160 \text{ bits})^{(1.024 \text{ bits})} \bmod 1.024 \text{ bits}$.

Raíz primitiva o generador de un cuerpo primo

Un generador o raíz primitiva de un cuerpo primo Z_p es aquel valor que, elevado a todos los restos del cuerpo reducido módulo n , genera todo el cuerpo.

Así, g es un generador si: $\forall 1 \leq a \leq p - 1$

$$g^a \bmod p = b \quad (\text{con } 1 \leq b \leq p - 1, \text{ todos los } b \neq 0)$$

Sea $p = 3 \Rightarrow \text{CCR} = \{1, 2\}$ (el cero no es solución)

Resto 1: no generará nada porque $1^k \bmod p = 1$

Resto 2: $2^1 \bmod 3 = 2$; $2^2 \bmod 3 = 1$

Luego el 2 es un generador del cuerpo Z_3

¿Cuántas raíces hay en un cuerpo primo ?

- Existen muchos números dentro del CRR que son generadores del cuerpo ... pero:
- Su búsqueda no es algo fácil ... ¿alguna solución?
- Conociendo la factorización de $p-1$ (q_1, q_2, \dots, q_n) con q_i los factores primos de $p-1$, diremos que un número g será generador en \mathbb{Z}_p si $\forall q_i$:

$$g^{(p-1)/q_i} \bmod p \neq 1$$

En cambio, si algún resultado es igual a 1, g no será generador.

Búsqueda de raíces primitivas en Z_{13} (1)

BÚSQUEDA DE RAÍCES EN EL CUERPO Z_{13}^*

Como $p = 13 \Rightarrow p-1 = 12 = 2^2 * 3$

Luego: $q_1 = 2 \quad q_2 = 3$

Si se cumple $g^{\frac{p-1}{q_i}} \bmod p \neq 1 \quad \forall q_i$ entonces g será un generador de p

Si se cumple $g^{\frac{p-1}{2}} \bmod p \neq 1 \wedge g^{\frac{p-1}{3}} \bmod p \neq 1$,
entonces g será un generador de p

Búsqueda de raíces primitivas en Z_{13} (1)

BÚSQUEDA DE RAÍCES EN EL CUERPO Z_{13}^*

$ZP13 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\};$

Table [{Mod[(ZP13[[i]])^6, 13], i}, {i, 1, Length[ZP13]}]

{{1, 1}, {12, 2}, {1, 3}, {1, 4}, {12, 5}, {12, 6}, {12, 7}, {12, 8}, {1, 9},
{1, 10}, {12, 11}, {1, 12}}

Table [{Mod[(ZP13[[i]])^4, 13], i}, {i, 1, Length[ZP13]}]

{{1, 1}, {3, 2}, {3, 3}, {9, 4}, {1, 5}, {9, 6}, {9, 7}, {1, 8}, {9, 9}, {3, 10},
{3, 11}, {1, 12}}

Generadores en Z_{13}

g:2,6,7,11

Búsqueda de raíces primitivas en Z_{17} (1)

BÚSQUEDA DE RAÍCES EN EL CAMPO Z_{17}^*

Como $p = 17 \Rightarrow p - 1 = 16 = 2^4$. Luego: $q = 2$

Si se cumple $g^{\frac{p-1}{2}} \bmod p \neq 1$ entonces g será un generador de p

$ZP17 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\};$

$\text{Table}[\{\text{Mod}[(ZP17[[i]])^8, 17], i\}, \{i, 1, \text{Length}[ZP17]\}]$

$\{\{1, 1\}, \{1, 2\}, \{16, 3\}, \{1, 4\}, \{16, 5\}, \{16, 6\}, \{16, 7\}, \{1, 8\}, \{1, 9\},$
 $\{16, 10\}, \{16, 11\}, \{16, 12\}, \{1, 13\}, \{16, 14\}, \{1, 15\}, \{1, 16\}\}$

Generadores en Z_{17}

$g: 3, 5, 6, 7, 10, 11, 12, 14$

Búsqueda de raíces primitivas en $Z_p(4)$

Generadores en Z_{13}

g: 2,6,7,11

Generadores en Z_{17}

g: 3,5,6,7,10,11,12,14

La tasa de generadores en el grupo p será aproximadamente $\tau = \phi(p-1)/(p-1)$.

Por lo tanto por lo general el 30% o más de los elementos del Conjunto Reducido de Restos de p será un generador en p .

$$\tau = \phi(12)/12$$

$$\tau = 4/12 = 1/3$$

$$\tau = \phi(16)/16$$

$$\tau = 8/16 = 1/2$$

Generadores en cuerpos de primos seguros

Un número primo p se dice que es un primo seguro o primo fuerte si: $p = 2 * p' + 1$ (con p' también primo).

Por ejemplo:

Si $p' = 11$, luego $p = 2 * 11 + 1 = 23$ (es primo y es seguro)

En este caso la tasa de números generadores del cuerpo será mayor que en el caso anterior (con $p = 13$ era del 30%).

Probabilidad: $\tau_{\text{pseguro}} = \phi(p-1)/p-1 \approx 1/2$

Casi la mitad de los números del grupo serán generadores en p .

Comprobación: El

Utilidad de la raíz primitiva en criptografía

¿Para qué sirve conocer la raíz primitiva de p ?

- La utilidad de este concepto en criptografía se ve en los sistemas de clave pública y, en particular, el protocolo de intercambio de claves de Diffie y Hellman.
- También se utiliza esta propiedad de los primos en la firma digital según estándar DSS (ElGamal).



Campos de Galois (GF)

Un campo es un conjunto F no vacío con dos operaciones internas $+$ y \cdot donde $(F, +)$ y $(F \setminus \{0\}, \cdot)$ son grupos conmutativos y se cumple la ley distributiva del producto con respecto a la suma. O sea para todo $a, b, c \in F$

$$a \cdot (b + c) = a \cdot b + a \cdot c;$$

$$(b + c) \cdot a = b \cdot a + c \cdot a$$

Es decir, un campo es un anillo conmutativo y unitario en el que todo elemento distinto de cero posee inverso.

La característica de un campo F es el menor entero positivo n tal que $nr = 0$ para todo r en F , se dice que F tiene característica n .

Un campo finito no es más que un campo con un número finito de elementos y se prueba que su característica es necesariamente un número primo p .

Cálculos en campos de Galois (GF)

- Campos finitos primos $GF(p)$, donde p es un número primo
- $GF(2)=\{0,1\}$
- $GF(3)=\{0,1,2\}$
- $GF(p)=\{0,1,2,\dots, p-1\}$
- Campos finitos extendidos se denotan por $GF(p^n)$, donde p es un primo y n un entero ≥ 2 .

Cálculos en campos de Galois (GF)

- Algunos usos en criptografía:
 - Sistemas de clave pública cuando la operación es $C = M^e \bmod p$ (cifrador ElGamal) o bien RSA usando el Teorema del Resto Chino para descifrar.
 - Aplicaciones en $GF(p^n)$, polinomios módulo p y de grado n de la forma $a(x) = a_{n-1} * x^{n-1} + a_{n-2} * x^{n-2} + \dots + a_1 * x + a_0$: se usará en el cifrador de flujo A5, el algoritmo Rijndael (AES) y los sistemas de curvas elípticas.

Elementos de $\text{GF}(p^n)$ como polinomios

- Los elementos del cuerpo $\text{GF}(p^n)$ se pueden representar como polinomios de grado $< n$ con coeficientes $a_i \in \text{GF}(p)$, es decir, en la forma:

$$a(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x + a_0$$

- El cuerpo $\text{GF}(p^n)$ se puede construir escogiendo un polinomio irreducible $p(x)$ de grado n con coeficientes en $\text{GF}(p)$. Entonces cada elemento $a(x)$ del cuerpo $\text{GF}(p^n)$ es un resto módulo $p(x)$.

Elementos de $GF(p^n)$ como polinomios

- Así, los elementos de $GF(2^n)$ son polinomios de grado $< n$ con coeficientes en $GF(2)=\{0, 1\}$.
- $GF(2^2)$ se considera el polinomio irreducible $x^2 + x + 1$
 $GF(2^2) = \{0, 1, x, x + 1\}$
- $GF(2^3)$ se considera el polinomio irreducible $x^3 + x + 1$
 $GF(2^3) = \{0, 1, x, 1 + x, x^2, x^2 + 1, x^2 + x + 1\}$
- $GF(2^4)$ se considera el polinomio irreducible $x^4 + x + 1$
 $GF(2^4)$
 $= \{0, 1, x, 1 + x, x^2, x^2 + 1, x^2 + x + 1, x^3, x^3 + 1, x^3 + x, x^3 + x^2 + 1\}$

Suma en campos de Galois $GF(2^n) \oplus$

Si el módulo de trabajo es 2 (con restos bits 0 y 1), las operaciones suma y resta serán un OR Exclusivo:

$$\begin{aligned} 0 \oplus 1 \bmod 2 &= 1 & 1 \oplus 0 \bmod 2 &= 1 \\ 0 \oplus 0 \bmod 2 &= 0 & 1 \oplus 1 \bmod 2 &= 0 \end{aligned}$$

$GF(2^2)$

\oplus	0	1	x	x+1
0	0	1	x	x+1
1	1	0	x+1	x
x	x	x+1	0	1
x+1	x+1	x	1	0

Restos: 0, 1, x, x+1

Como los resultados deberán pertenecer al cuerpo 2^2 , vamos a aplicar **Reducción por Coeficientes**.
Ejemplo de cálculos en mod 2:
 $x + (x + 1) = 2x + 1 \bmod 2 = 1$
 $1 + (x + 1) = 2 + x \bmod 2 = x$

Producto en campos de Galois $GF(2^n)$

La operación multiplicación puede entregar elementos que no pertenezcan al campo, potencias iguales o mayores que $n \Rightarrow$ **Reducción por Exponente.**

$GF(2^2)$

\otimes	0	1	x	x+1
0	0	0	0	0
1	0	1	x	x+1
x	0	x	x+1	1
x+1	0	x+1	1	x

Restos: 0, 1, x, x+1

Para la reducción por exponente, sea el polinomio irreducible de grado 2 el siguiente: $p(x) = x^2 + x + 1$.

Luego: $x^2 = x + 1$

Cálculo de $(x+1)*(x+1) \bmod 2$:

$$(x+1)*(x+1) = x^2 + 2x + 1 \bmod 2$$

$$(x+1)*(x+1) = (x+1) + 2x + 1 \bmod 2$$

$$(x+1)*(x+1) = 3x + 2 \bmod 2 = x$$

Operaciones con campos de Galois en AES

- La suma y multiplicación de polinomios dentro de un campo binario descritas en diapositivas anteriores conforman las operaciones básicas del algoritmo de cifra **A**dvanced **E**ncryption **S**tandard AES, que con el nombre Rijndael es el estándar mundial desde finales de 2001, desplazando al ya viejo DES.
- En este caso, se trabaja con 8 bits por lo que las operaciones se realizan en $GF(2^8)$. En el tema de cifra en bloque con clave secreta encontrará ejemplos de suma y multiplicación polinómica dentro de este cuerpo binario para el AES.

Cuestiones y ejercicios (1 de 2)

1. ¿Qué significa para la criptografía el homomorfismo de los enteros?
2. En un cuerpo de cifra n , ¿existen siempre los inversos aditivos y los inversos multiplicativos? ¿Debe cumplirse alguna condición?
3. En un cuerpo n el inverso de a es a^{-1} , ¿es ese valor único? ¿Por qué?
4. Cifraremos en un cuerpo $n = 131$. ¿Cuál es el valor del CCR? ¿Cuál es valor del CRR? ¿Qué valores podemos cifrar?
5. Para cifrar un mensaje $M = 104$ debemos elegir el cuerpo de cifra entre el valor $n = 127$ y $n = 133$. ¿Cuál de los dos usaría y por qué?
6. ¿Qué nos dice la función $\phi(n)$ de Euler? ¿Para qué sirve?
7. ¿Qué papel cumple el algoritmo extendido de Euclides en la criptografía? ¿Por qué es importante? ¿En qué se basa?

Cuestiones y ejercicios (2 de 2)

9. Si en el cuerpo $n = 37$ el $\text{inv}(21, 37) = 30$, ¿cuál es el $\text{inv}(30, 37)$?
10. Usando el algoritmo extendido de Euclides calcule los siguientes inversos: $\text{inv}(7, 19)$; $\text{inv}(21, 52)$, $\text{inv}(11, 33)$, $\text{inv}(47, 41)$.
11. ¿Cuántas soluciones x_i hay en la expresión $8 \cdot x \bmod 20 = 12$? Explique lo que sucede. ¿Tendría esto interés en criptografía?
16. Cómo se define un primo seguro? ¿Cuántos generadores tiene?
17. A partir de los valores $p' = 13$, $p' = 17$, $p' = 19$ y $p' = 23$ queremos obtener un primo seguro, ¿con cuál o cuáles de ellos lo logramos?
18. Usando el algoritmo de exponenciación rápida calcule los siguientes valores: $23^{32} \bmod 51$; $100^{125} \bmod 201$; $1.000^{100.000} \bmod 2.500$.
19. En $\text{GF}(2^n)$ reduzca por coeficientes $5x^5 + x^4 + 2x^3 + 3x^2 + 6x + 2$.
20. Reduzca $(x^3 + 1)(x^2 + x + 1)$ por exponente en $\text{GF}(2^n)$ usando como polinomio primitivo $p(x) = x^4 + x + 1$, es decir $x^4 = x + 1$.