

# Capítulo 7

## Teoría de los Números

### Seguridad Informática y Criptografía



v 4.1



Material Docente de  
Libre Distribución

Ultima actualización del archivo: 01/03/06  
Este archivo tiene: 75 diapositivas

Dr. Jorge Ramió Aguirre  
Universidad Politécnica de Madrid

Este archivo forma parte de un curso completo sobre Seguridad Informática y Criptografía. Se autoriza el uso, reproducción en computador y su impresión en papel, sólo con fines docentes y/o personales, respetando los créditos del autor. Queda prohibida su comercialización, excepto la edición en venta en el Departamento de Publicaciones de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, España.

# 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:

$$\begin{aligned}a - b &= k * n \\ a &\equiv_n b \\ a &\equiv b \pmod n\end{aligned}$$

Desde esta página Web podrá realizar diversos cálculos en matemática discreta:

<http://www.numbertheory.org/php/php.html>



# Operaciones de congruencia en $Z_n$

¿Es 18 congruente con 3 módulo 5?

¿ $18 \equiv 3 \pmod{5}$ ?

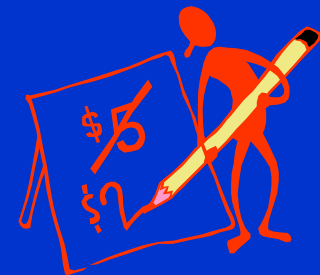
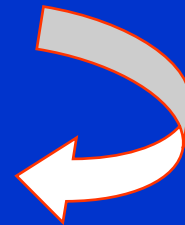
Sí, porque:  $18 - 3 = 15 = k * 5$  con  $k = 3$

¿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 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:

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

# 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:**

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

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

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

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

- **Existencia de Identidad:**

$$a + 0 \text{ mod } n = 0 + a \text{ mod } n = a \text{ mod } n = a$$

$$a * 1 \text{ mod } n = 1 * a \text{ mod } n = a \text{ mod } n = a$$

- **Existencia de Inversos:**



$$a + (-a) \text{ mod } n = 0$$

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

✓ Ambos serán muy importantes en criptografía

→ No siempre existe

- **Reducibilidad:**



$$(a + b) \text{ mod } n = [(a \text{ mod } n) + (b \text{ mod } n)] \text{ mod } n$$

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

# 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:

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

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

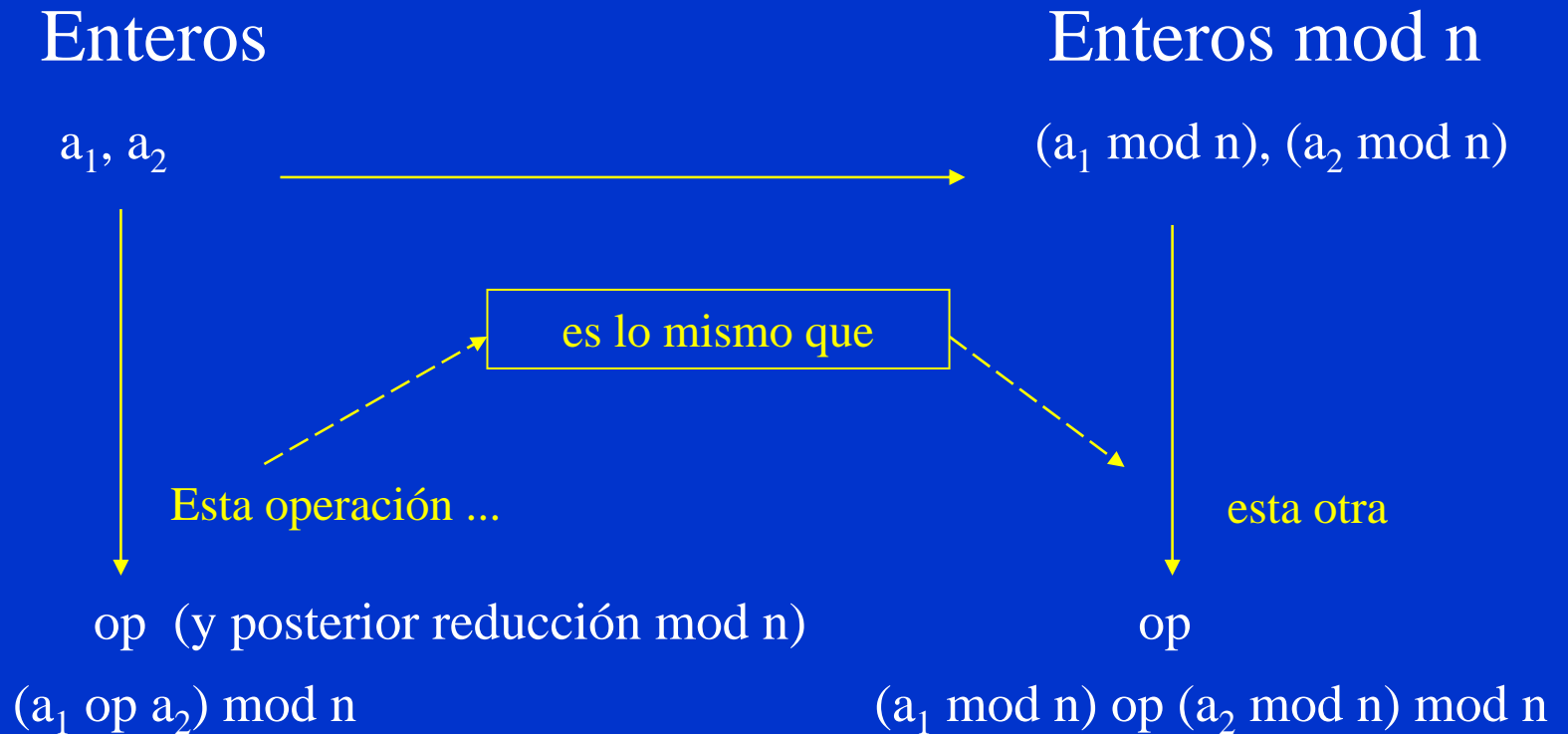
$$CCR(6) = \{0, 1, 2, 3, 4, 5\} = \{12, 7, 20, 9, 16, 35\}$$

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

$$88 * 93 \text{ mod } 13$$

$$8.184 \text{ mod } 13$$

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:

$$88 * 93 \text{ mod } 13$$

$$[(88) \text{ mod } 13 * (93) \text{ mod } 13] \text{ mod } 13$$

$$10 * 2 \text{ mod } 13$$

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

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

Algoritmo de Euclides:

- a) Si  $x$  divide a  $a$  y  $b \Rightarrow a = x * a'$  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 mcd

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

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

(con  $d_1$  un entero y  $r$  un resto)

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

porque:

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

(con  $r$  un entero y  $r'$  un resto)

# 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.

# 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  $Z_n$  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 $Z_n$

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

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  $2 * x \equiv 1 \pmod 6$  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:

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

# Existencia del inverso por primalidad

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

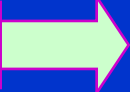
Si  $\text{mcd}(a, n) = 1$ , el resultado de  $a * i \text{ mod } n$  (para  $i$  todos los restos de  $n$ ) serán valores distintos dentro del cuerpo  $n$ .

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

Sea:  $a = 4$  y  $n = 9$ .

Valores de  $i = \{1, 2, 3, 4, 5, 6, 7, 8\}$

S O L U C I Ó N	$4 * 1 \text{ mod } 9 = 4$	$4 * 2 \text{ mod } 9 = 8$	$4 * 3 \text{ mod } 9 = 3$
	$4 * 4 \text{ mod } 9 = 7$	$4 * 5 \text{ mod } 9 = 2$	$4 * 6 \text{ mod } 9 = 6$
	$4 * 7 \text{ mod } 9 = 1$	$4 * 8 \text{ mod } 9 = 5$	
	Ú N I C A		

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

# 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.

# No existencia de inversos multiplicativos

$(A*B) \bmod 10$

	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.

<http://www.cut-the-knot.org/blue/Modulo.shtml>



# 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 el grupo  $n$ .
- Si  $n$  es primo, todos los restos serán primos con él.
- Como el cero no es una solución, entonces:

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

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

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

# 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 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.



<http://es.wikipedia.org/wiki/Euler>



# 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:

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

<http://mathworld.wolfram.com/TotientFunction.html>



# 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.

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

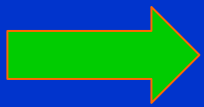
Si  $n$  es primo, entonces  $CRR = CCR - 1$  ya que todos los restos de  $n$ , excepto el cero, serán primos entre sí.

Ejemplo

$$CRR(7) = \{1, 2, 3, 4, 5, 6\} \text{ seis elementos}$$

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

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



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

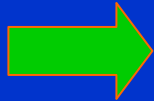
# Función $\phi(n)$ de Euler cuando $n = p^k$

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

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

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

Ejemplo



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

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

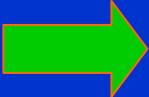
De los  $p*q$  elementos del CCR, restaremos todos los múltiplos de  $p = 1*p, 2*p, \dots (q - 1)*p$ , todos los múltiplos de  $q = 1*q, 2*q, \dots (p - 1)*q$  y el cero.

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

Esta expresión se usará en el sistema de cifra RSA.



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

Ejemplo  $CRR(15) = \{1, 2, 4, 7, 8, 11, 13, 14\}$  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:  $n = p_1^{e_1} * p_2^{e_2} * \dots * p_t^{e_t}$  ( $p_i$  son primos)

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

Ejemplo



(Esta demostración no es inmediata)

$$\begin{aligned} \text{CRR}(20) &= \{1, 3, 7, 9, 11, 13, 17, 19\} \text{ ocho elementos} \\ \therefore \phi(20) &= \phi(2^2 * 5) = 2^{2-1}(2-1) * 5^{1-1}(5-1) = 2^1 * 1 * 1 * 4 = 8 \\ \phi(360) &= \phi(2^3 * 3^2 * 5) = 2^{3-1}(2-1) * 3^{2-1}(3-1) * 5^{1-1}(5-1) = 96 \end{aligned}$$

# Teorema de Euler

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

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

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

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

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

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

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

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

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

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

Y se cumple también que:

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

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

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

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

# Pequeño teorema de Fermat

Si el cuerpo de trabajo es un primo  $p$ :

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

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

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 el inverso de  $a$  en el primo  $p$ .

[http://es.wikipedia.org/wiki/Peque%C3%B1o\\_teorema\\_de\\_Fermat](http://es.wikipedia.org/wiki/Peque%C3%B1o_teorema_de_Fermat)



## ¿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

<http://en.wikipedia.org/wiki/Euclid>





# Algoritmo Extendido de Euclides AEE

Si  $\text{mcd}(a, n) = 1$  y  $a \cdot x \pmod n = 1 \Rightarrow x = \text{inv}(a, n)$

Luego podemos escribir:

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

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

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

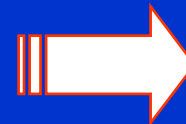
...

...

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

$$r_{n-1} = C_{n+1} \cdot 1 + 0$$

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

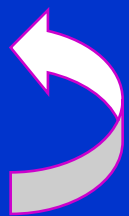
Ordenando por restos desde el valor 1 se llega a una expresión del tipo  $(k_1 * n + k_2 * a) \bmod n = 1$ , en donde el inverso de  $a$  en  $n$  lo dará el coeficiente  $k_2$  puesto que  $k_1 * n \bmod n = 0$ .

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

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

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)  $25 = 2 \cdot 9 + 7$

b)  $9 = 1 \cdot 7 + 2$

c)  $7 = 3 \cdot 2 + 1$

d)  $2 = 2 \cdot 1 + 0$

$$7 = 25 - 2 \cdot 9$$

$$2 = 9 - 1 \cdot 7$$

$$1 = 7 - 3 \cdot 2$$

$$7 = 25 - 2 \cdot 9$$

$$2 = 9 - 1 \cdot (25 - 2 \cdot 9) = 3 \cdot 9 - 1 \cdot 25$$

$$1 = (25 - 2 \cdot 9) - 3 \cdot (3 \cdot 9 - 1 \cdot 25)$$

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

restos

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  $x = v_{i-1}$

**Ejemplo** →



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

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

Diagrama de flujo: Una línea punteada azul muestra la actualización de  $u_i$  y  $v_i$  desde  $i=1$  hasta  $i=3$ . Un círculo rojo con 'x' está sobre  $u_3$  y un círculo rojo con '=' está sobre  $v_3$ . Una flecha roja apunta desde  $v_4 = -11$  hacia el resultado final.

# 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  $a * x \text{ mod } n = b$  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 ...

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

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

$$x_1 = 4 \quad \Rightarrow \quad 6 * 4 \text{ mod } 10 = 24 \text{ mod } 10 = 4$$

$$x_2 = 9 \quad \Rightarrow \quad 6 * 9 \text{ mod } 10 = 54 \text{ mod } 10 = 4$$



# 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  $[0, n-1]$

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

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

☞ En algunos textos lo verá como “Teorema Chino de los Restos”... aunque es obvio que la autoría pertenece a los matemáticos chinos, alguien podría poner en duda si el teorema es chino o bien si los restos son los chinos 😊.

<http://www.math.hawaii.edu/~lee/courses/Chinese.pdf>



# Ejemplo de aplicación del TRC (1)

Encontrar  $x$  de forma que :  $12 * x \bmod 3.960 = 36$

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

$$n = 3.960 \Rightarrow n = 2^3 * 3^2 * 5 * 11 = d_1 * d_2 * d_3 * d_4 = 8 * 9 * 5 * 11$$

$$a = 12$$

$$b = 36$$

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

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

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

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

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

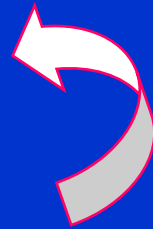
4 ecuaciones en  $x$

Resolviendo para  $x_i$

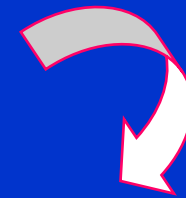


# Ejemplo de aplicación del TRC (2)

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



4 ecuaciones en  $x$



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

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

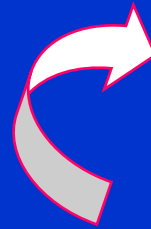
$$12*x_3 \bmod 5 = 1 \Rightarrow 2*x_3 \bmod 5 = 1 \Rightarrow x_3 = 3$$

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

# Ejemplo de aplicación del TRC (3)

Resolvemos ahora la ecuación auxiliar del Teorema Resto Chino

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



$y_1 = 7$	$y_2 = 8$
$y_3 = 3$	$y_4 = 7$

$$y_1 = \text{inv} [(n/d_1), d_1] \Rightarrow y_1 = \text{inv} [(3.960/8), 8] = \text{inv} (495, 8)$$

$$y_2 = \text{inv} [(n/d_2), d_2] \Rightarrow y_2 = \text{inv} [(3.960/9), 9] = \text{inv} (440, 9)$$

$$y_3 = \text{inv} [(n/d_3), d_3] \Rightarrow y_3 = \text{inv} [(3.960/5), 5] = \text{inv} (792, 5)$$

$$y_4 = \text{inv} [(n/d_4), d_4] \Rightarrow y_4 = \text{inv} [(3.960/11), 11] = \text{inv} (360, 11)$$

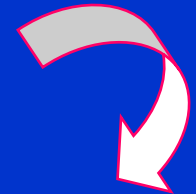
# Ejemplo de aplicación del TRC (4)

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

$$\begin{array}{ll} y_1 = 7 & y_2 = 8 \\ y_3 = 3 & y_4 = 7 \end{array}$$

Aplicando ecuación del Resto Chino para el caso  $12 * x \bmod 3.960 = 36$  con  $d_1 = 8, d_2 = 9, d_3 = 5, d_4 = 11$ :

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



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

$$x = [495*7*1 + 440*8*0 + 792*3*3 + 360*7*3] \bmod 3.960$$

$$x = [3.465 + 0 + 7.128 + 7.560] \bmod 3.960 = 2.313$$

# ¿Todo marcha bien en este ejemplo?

¿Es la solución de  $12*x \bmod 3.960 = 36$  única?

**NO**

¿Qué ha sucedido?

Puesto que  $\text{mcd}(a, n) = \text{mcd}(12, 3.960) = 12$ , ya hemos visto en una diapositiva anterior que habrá 12 soluciones válidas.

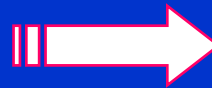
$$x_1 = 3; x_2 = 333; x_3 = 663; x_4 = 993 \quad \dots \quad x_8 = \underline{2.313} \dots$$
$$x_i = 3 + (i-1)*330 \bmod 3.960 \quad \dots \quad \text{hasta llegar a } x_{12} = 3.633$$

Observe que  $x = 2.313$ , uno de los valores solución, fue el resultado encontrado en el ejercicio anterior.

# Otros casos de aplicación del TRC

¿Qué sucede ahora con:

$$12*x \bmod 3.960 = 35?$$

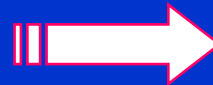


$$12*x \bmod 3.960 = 35$$

$\text{mcd}(a, n) = 12$  no es un divisor de  $b = 35$ , luego aquí no existe solución.

Teníamos que

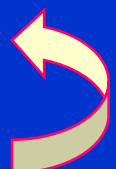
$$3.960 = 2^3 * 3^2 * 5 * 11$$



¿Qué sucede ahora con:

$$49*x \bmod 3.960 = 1?$$

Primero encuentre  $x$ .  
Luego vea la solución.



$$49*x \bmod 3.960 = 1$$

Sí existirá  $x$ , y en este caso es el inverso de 49. Será único ya que  $49 = 7*7$  no tiene factores en  $n$ .

# Cálculo de inversos usando el TRC (1)

Si  $49 * x \bmod 3.960 = 1$ , se pide encontrar  $x = \text{inv}(49, 3.960)$

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

$$n = 3.960 \Rightarrow n = 2^3 * 3^2 * 5 * 11 = d_1 * d_2 * d_3 * d_4 = 8 * 9 * 5 * 11$$

$$a = 49$$

$$b = 1$$

Como  $n = d_1 * d_2 * d_3 * d_4$  existirán 4 soluciones de  $x_i$

$$a * x_1 \bmod d_1 = b \bmod d_1$$

$$49 * x_1 \bmod 8 = 1 \bmod 8 = 1$$

$$a * x_2 \bmod d_2 = b \bmod d_2$$

$$49 * x_2 \bmod 9 = 1 \bmod 9 = 1$$

$$a * x_3 \bmod d_3 = b \bmod d_3$$

$$49 * x_3 \bmod 5 = 1 \bmod 5 = 1$$

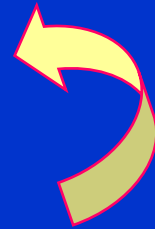
$$a * x_4 \bmod d_4 = b \bmod d_4$$

$$49 * x_4 \bmod 11 = 1 \bmod 11 = 1$$

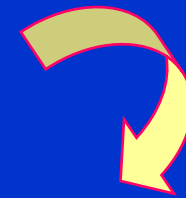
Resolviendo para  $x_i$

# Cálculo de inversos usando el TRC (2)

$$\begin{array}{ll} x_1 = 1 & x_2 = 7 \\ x_3 = 4 & x_4 = 9 \end{array}$$



4 ecuaciones en  $x$



$$49 * x_1 \bmod 8 = 1 \Rightarrow 1 * x_1 \bmod 8 = 1 \Rightarrow x_1 = 1$$

$$49 * x_2 \bmod 9 = 1 \Rightarrow 4 * x_2 \bmod 9 = 1 \Rightarrow x_2 = 7$$

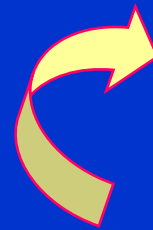
$$49 * x_3 \bmod 5 = 1 \Rightarrow 4 * x_3 \bmod 5 = 1 \Rightarrow x_3 = 4$$

$$49 * x_4 \bmod 11 = 1 \Rightarrow 5 * x_4 \bmod 11 = 1 \Rightarrow x_4 = 9$$

# Cálculo de inversos usando el TRC (3)

Resolvemos ahora la ecuación auxiliar del Teorema Resto Chino

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



$y_1 = 7$	$y_2 = 8$
$y_3 = 3$	$y_4 = 7$

$$y_1 = \text{inv} [(3.960/8), 8] \Rightarrow y_1 = \text{inv} (495, 8) = \text{inv} (7, 8) = 7$$

$$y_2 = \text{inv} [(3.960)/9, 9] \Rightarrow y_2 = \text{inv} (440, 9) = \text{inv} (8, 9) = 8$$

$$y_3 = \text{inv} [(3.960)/5, 5] \Rightarrow y_3 = \text{inv} (792, 5) = \text{inv} (2, 5) = 3$$

$$y_4 = \text{inv} [(3.960)/11, 11] \Rightarrow y_4 = \text{inv} (360, 11) = \text{inv} (8, 11) = 7$$



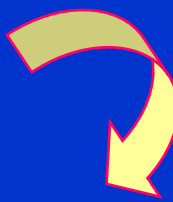
# Cálculo de inversos usando el TRC (4)

$$x_1 = 1 \quad x_2 = 7$$

$$x_3 = 4 \quad x_4 = 9$$

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

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

$$x = \sum_{i=1}^t (n/d_i) * y_i * x_i \pmod n$$


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

$$x = [495*7*1 + 440*8*7 + 792*3*4 + 360*7*9] \pmod{3.960}$$

$$x = [3.465 + 880 + 1.584 + 2.880] \pmod{3.960} = 889$$

En efecto,  $\text{inv}(49, 3.960) = 889$  ... pero

## Utilidad del Teorema del Resto Chino

Calcular el inverso de 49 en el cuerpo  $\mathbb{Z}/3960$  por medio del Teorema del Resto Chino es algo tedioso ..... ☹ y bastante absurdo como ya lo habrá comprobado ☺.

En el desarrollo del propio algoritmo del Teorema del Resto Chino para encontrar un inverso hay que calcular otros inversos lo que no tiene sentido alguno...

**¿Para qué sirve entonces este algoritmo?**

Entre otras cosas, cuando veamos el sistema de cifra RSA y el tema dedicado a Protocolos Criptográficos, encontraremos una **interesante** aplicación del teorema.

# La exponenciación en la cifra asimétrica

- ✓ Una de las aplicaciones más interesantes de la matemática discreta en criptografía es 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. Uno de ellos es el que se presenta en la siguiente diapositiva.

# 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

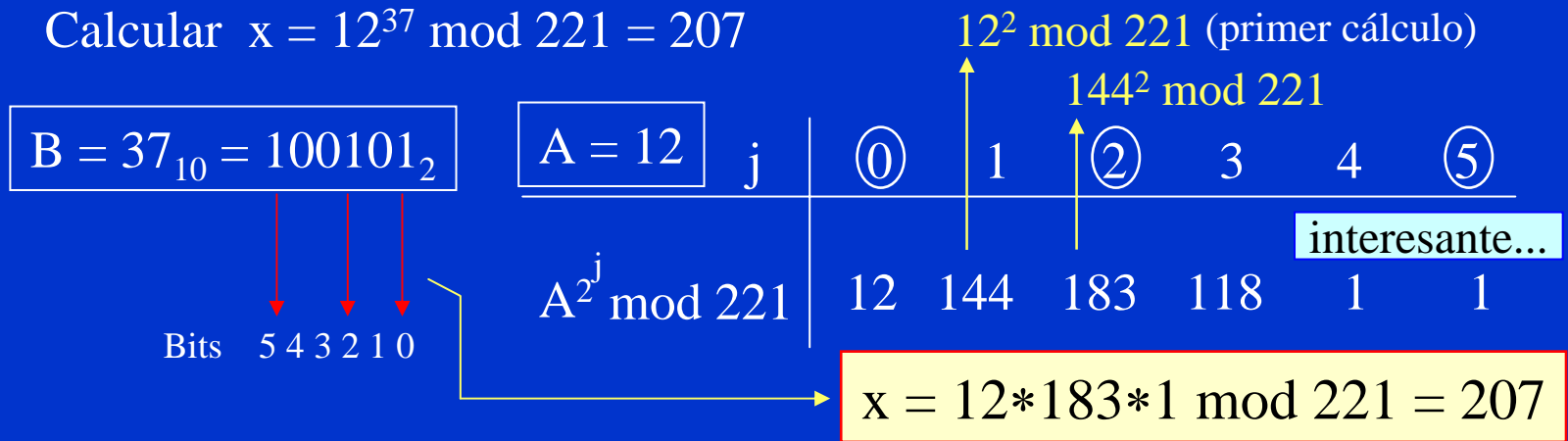
Calcular  $x = 12^{37} \bmod 221 = 207$

$12^{37}$  es un número de 40 dígitos:

8505622499821102144576131684114829934592

# Ejemplo de exponenciación rápida

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



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  $x = A^B \pmod n$

- 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  
 $x = x^2 \pmod n$   
 Si  $(b_i = 1)$  entonces  
 $x = x * A \pmod n$

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

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

$$x = 1$$

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

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

# ¿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 $2^5 = 32$	$x/\ln x = 32/3,46 = 9$	Probabilidad x sea primo: 30,00 %
• Primos entre 2 y $2^6 = 64$	$x/\ln x = 64/4,16 = 15$	Probabilidad x sea primo: 24,00 %
• Primos entre 2 y $2^7 = 128$	$x/\ln x = 128/4,85 = 26$	Probabilidad x sea primo: 20,63 %
• Primos entre 2 y $2^8 = 256$	$x/\ln x = 256/5,54 = 46$	Probabilidad x sea primo: 18,11 %
• Primos entre 2 y $2^9 = 512$	$x/\ln x = 512/6,23 = 82$	Probabilidad x sea primo: 16,08 %
• Primos entre 2 y $2^{10} = 1.024$	$x/\ln x = 1.024/6,93 = 147$	Probabilidad x sea primo: 14,38 %
• Primos entre 2 y $2^{11} = 2.048$	$x/\ln x = 2.048/7,62 = 268$	Probabilidad x sea primo: 13,10 %
• Primos entre 2 y $2^{12} = 4.096$	$x/\ln x = 4.096/8,32 = 492$	Probabilidad x sea primo: 12,02 %

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

<http://www.utm.edu/research/primes/>



# Raíz primitiva o generador de un primo

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

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

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

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  $n = 3$



## ¿Cuántas raíces hay en un primo $p$ ?

- 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  $p$  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.

<http://mathworld.wolfram.com/PrimitiveRoot.html>



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

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

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

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

Si se cumple  $g^{(p-1)/q_i} \pmod p \neq 1 \quad \forall q_i$   
 entonces  $g$  será un generador de  $p$

Generadores en  $Z_{13}$

$g: 2,$

$$2^{(13-1)/2} \pmod{13} = 2^6 \pmod{13} = 12$$

$$2^{(13-1)/3} \pmod{13} = 2^4 \pmod{13} = 3$$

$$3^{(13-1)/2} \pmod{13} = 3^6 \pmod{13} = 1$$

$$3^{(13-1)/3} \pmod{13} = 3^4 \pmod{13} = 3$$



Resto 2

*El resto 2 sí es generador*



Resto 3

*El resto 3 no es generador*

# Búsqueda de raíces primitivas en $Z_{13}$ (2)

Generadores en  $Z_{13}$

g: 2, 6, 7,

$4^{(13-1)/2} \bmod 13 = 4^6 \bmod 13 = 1$	Resto 4
$4^{(13-1)/3} \bmod 13 = 4^4 \bmod 13 = 9$	<i>El resto 4 no es generador</i>
$5^{(13-1)/2} \bmod 13 = 5^6 \bmod 13 = 12$	Resto 5
$5^{(13-1)/3} \bmod 13 = 5^4 \bmod 13 = 1$	<i>El resto 5 no es generador</i>
$6^{(13-1)/2} \bmod 13 = 6^6 \bmod 13 = 12$	 Resto 6
$6^{(13-1)/3} \bmod 13 = 6^4 \bmod 13 = 9$	<i>El resto 6 sí es generador</i>
$7^{(13-1)/2} \bmod 13 = 7^6 \bmod 13 = 12$	 Resto 7
$7^{(13-1)/3} \bmod 13 = 7^4 \bmod 13 = 9$	<i>El resto 7 sí es generador</i>

# Búsqueda de raíces primitivas en $Z_{13}$ (3)

Generadores en  $Z_{13}$

g: 2, 6, 7, 11

$$8^{(13-1)/2} \bmod 13 = 8^6 \bmod 13 = 12 \quad \text{Resto 8}$$

$$8^{(13-1)/3} \bmod 13 = 8^4 \bmod 13 = 1 \quad \text{El resto 8 no es generador}$$

$$9^{(13-1)/2} \bmod 13 = 9^6 \bmod 13 = 1 \quad \text{Resto 9}$$

$$9^{(13-1)/3} \bmod 13 = 9^4 \bmod 13 = 9 \quad \text{El resto 9 no es generador}$$

$$10^{(13-1)/2} \bmod 13 = 10^6 \bmod 13 = 1 \quad \text{Resto 10}$$

$$10^{(13-1)/3} \bmod 13 = 10^4 \bmod 13 = 3 \quad \text{El resto 10 no es generador}$$

$$11^{(13-1)/2} \bmod 13 = 11^6 \bmod 13 = 12 \quad \text{Resto 11}$$

$$11^{(13-1)/3} \bmod 13 = 11^4 \bmod 13 = 3 \quad \text{El resto 11 sí es generador}$$



# Búsqueda de raíces primitivas en $Z_{13}$ (4)

Generadores en  $Z_{13}$

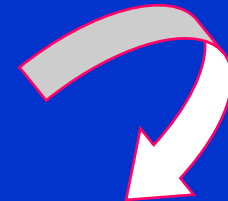
g: 2, 6, 7, 11

$$12^{(13-1)/2} \bmod 13 = 12^6 \bmod 13 = 1$$

Resto 12

$$12^{(13-1)/3} \bmod 13 = 12^4 \bmod 13 = 1 \text{ El resto 12 no es generador}$$

La tasa de generadores en el grupo  $p$  será aproximadamente  $\tau = \phi(p-1)/(p-1)$ . Por lo tanto por lo general el 30% de los elementos del Conjunto Reducido de Restos de  $p$  será un generador en  $p$ .



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

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

# 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%).

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

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



Comprobación

# Comprobación de generadores en $p = 2p' + 1$

$$p' = 11; \quad 2p' = 22; \quad p = 2p' + 1 = 23 \text{ primo seguro}$$

Como  $2p' = p - 1$  existirán:

$\phi(p') = [p' - 1]$  elementos de orden  $(p')$  en el CRR

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

$\phi(2p') = [p' - 1]$  elementos de orden  $(p-1)$  en el CRR

$$\phi(22) = 10 = \{1, 3, 5, 7, 9, 13, 15, 17, 19, 21\}$$

$$\tau = (p' - 1)/(p-1) = (p' - 1)/2p' \approx 1/2$$

*Sigue* 

# Comprobación de generadores en $p = 2p' + 1$

Usando la ecuación  $g^{(p-1)/q_i} \pmod p$

En este caso con  $q_1 = 2$  y  $q_2 = 11$

$$g^{(23-1)/2} \pmod{23} = g^{11} \pmod{23}$$

$$g^{(23-1)/11} \pmod{23} = g^2 \pmod{23}$$

Encontramos los siguientes 10 generadores en  $p = 23$

$$\{5, 7, 10, 11, 14, 15, 17, 19, 20, 21\}$$

Es decir, prácticamente la mitad de los valores de CRR que en este caso es igual a  $23 - 1 = 22$ .

Observe cómo se distribuyen los valores de estas raíces dentro del primo, en forma de grupos y distribuidos uniformemente.



# Utilidad de la raíz primitiva en criptografía

¿Para qué sirve conocer la raíz primitiva de  $p$ ?

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



# Cálculos en campos de Galois (GF)

- Cuando trabajamos en un cuerpo  $K$  con dos operaciones  $+$  y  $*$ , sabemos que todo elemento distinto del cero tiene un único inverso multiplicativo. Si el cuerpo es finito, se denomina también cuerpo o campo de Galois y se denota por  $GF(p^n)$ , donde  $p$  es un primo y  $n$  un entero  $\geq 1$ .
- 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, como se verá en ese capítulo.
  - 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 de AES y los sistemas de curvas elípticas.

<http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Galois.html>



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

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

$$a(x) = a_{n-1} * x^{n-1} + a_{n-2} * x^{n-2} + \dots + a_1 * x + a_0$$

- El cuerpo  $GF(p^n)$  se puede construir escogiendo un polinomio irreducible  $p(x)$  de grado  $n$  a coeficientes en  $\mathbb{Z}_p$ . Entonces cada elemento  $a(x)$  del cuerpo  $GF(p^n)$  es un resto módulo  $p(x)$ .
- Así, los elementos de  $GF(2^n)$  son polinomios de grado  $< n$  con coeficientes en  $\{0, 1\}$ . De esta manera,  $GF(2^3)$  tiene 8 elementos o restos polinómicos que son:  $0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1$ , los 8 restos de un polinomio de grado  $n-1$  ( $n = 3$ ).
- En el capítulo 21 encontrará una tabla de polinomios primitivos.

<http://mathworld.wolfram.com/FiniteField.html>



# 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:

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

$CG(2^2)$

$\oplus$	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:

$$x + (x + 1) = 2x + 1 \text{ mod } 2 = 1$$

$$1 + (x + 1) = 2 + x \text{ mod } 2 = x$$

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

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

**CG(2<sup>2</sup>)**

$\otimes$	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 el polinomio irreducible de grado 2 el siguiente:  $p(x) = x^2 + x + 1$ .

Luego:  $x^2 = x + 1$

Cálculo de  $(x+1)*(x+1) \text{ mod } 2$ :

$(x + 1)*(x + 1) = x^2 + 2x + 1 \text{ mod } 2$

$(x + 1)*(x + 1) = (x + 1) + 2x + 1 \text{ mod } 2$

$(x + 1)*(x + 1) = 3x + 2 \text{ mod } 2 = x$

# Operaciones con campos de Galois en AES

- ✓ La suma y multiplicación de polinomios dentro de un cuerpo 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 capítulo de cifra en bloque con clave secreta encontrará ejemplos de suma y multiplicación polinómica dentro de este cuerpo binario para el AES.

Fin del capítulo

## Cuestiones y ejercicios (1 de 3)

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

## Cuestiones y ejercicios (2 de 3)

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  $8*x \bmod 20 = 12$ ? Explique lo que sucede. ¿Tendría esto interés en criptografía?
12. ¿Qué viene a significar el Teorema del Resto Chino? Aunque aún no lo ha estudiado, ¿le ve alguna utilidad en criptografía?
13. Calcule  $\text{inv}(121, 393)$  usando el Teorema del Resto Chino.
14. Defina lo que es una raíz primitiva o generador de un cuerpo. ¿Es necesario que ese cuerpo sea un primo?
15. ¿Cuántos generadores podemos esperar en el cuerpo  $p = 17$ ? Y si ahora  $p = 7$ , ¿cuántos generadores habrá? Compruébelo calculando todos los exponentes del conjunto completo de restos de  $p = 7$ .



## Cuestiones y ejercicios (3 de 3)

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. Comente el ahorro en número de operaciones del ejercicio anterior.
20. Compruebe los resultados (si puede) con calculadoras de Windows 3.1; Windows 95 y actual. Puede encontrar los ejecutables de estas calculadoras en el software de la asignatura de nombre CripClas.
21. ¿Qué sucede con estas calculadoras para números muy grandes?
22. En  $\text{GF}(2^n)$  reduzca por coeficientes  $5x^5 + x^4 + 2x^3 + 3x^2 + 6x + 2$ .
23. 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  $x^4 = x + 1$ .

Use el portapapeles

## Prácticas del tema 7 (1/2)

### Software CripClas:

[http://www.criptored.upm.es/software/sw\\_m001c.htm](http://www.criptored.upm.es/software/sw_m001c.htm)



1. Calcule  $237 \bmod 10$ ;  $1452 \bmod 314$ ;  $31 \bmod 49$ ;  $3565 \bmod 115$ .
2. Calcule  $\text{mcd}(384, 42)$ ;  $\text{mcd}(1234, 56)$ ;  $\text{mcd}(23, 5)$ ;  $\text{mcd}(371, 97)$ .
3. Calcule  $\phi(7)$ ;  $\phi(77)$ ;  $\phi(131)$ ;  $\phi(200)$ .
4. Calcule  $\text{inv}(5, 27)$ ;  $\text{inv}(12, 133)$ ;  $\text{inv}(21, 25)$ ;  $\text{inv}(63, 189)$ .

### Software Fortaleza:

[http://www.criptored.upm.es/software/sw\\_m001e.htm](http://www.criptored.upm.es/software/sw_m001e.htm)



1. Calcule  $2^8 \bmod 200$ ;  $14^{1001} \bmod 5321$ ;  $4902564^{1053501} \bmod 34090062349$ .
2. Repita estos cálculos usando las calculadoras de Windows 3.1 y Windows 95 que encontrará en la carpeta de CripClas. Calcule ahora estas potencias:  $10^{15} \bmod 61$  y  $300^{21} \bmod 45$ . ¿Qué ha pasado?

Este bug originado por un uso indebido de la operación módulo con números en formato coma flotante (descubierto de forma fortuita, todo hay que decirlo) fue notificado por este autor vía email a Microsoft en el año 1995 y subsanado en la edición de la calculadora de Windows 98 y versiones posteriores.

Use el portapapeles

## Prácticas del tema 7 (2/2)

3. Calcule si son primos: 23; 371; 19841; 27603543067280716373.
4. Calcule  $\text{mcd}(13824552, 9315188)$ ;  $\text{mcd}(6276359, 8725413290)$ .
5. Calcule  $\text{inv}(324762387638768, 893247293874293879873498787987)$ .
6. Calcule  $87363553226^{6530982763424323401728} \bmod 98774655534452522982343$ .
7. Compruebe que una exponenciación de 50 dígitos  $^{100 \text{ dígitos}} \bmod 200 \text{ dígitos}$  tarda aproximadamente 30 segundos en resolverse con este programa.

**Software ExpoCrip:**

[http://www.criptored.upm.es/software/sw\\_m0011.htm](http://www.criptored.upm.es/software/sw_m0011.htm)



1. Calcule las raíces primitivas de los siguientes números: 5; 19; 31; 57; 61.
2. Compruebe que estos números son primos seguros: 23; 503; 1019; 10007.
3. Calcule las raíces primitivas de los primos seguros del apartado 2.
4. Compare el porcentaje de raíces primitivas encontradas en números primos normales y en primos seguros o fuertes.