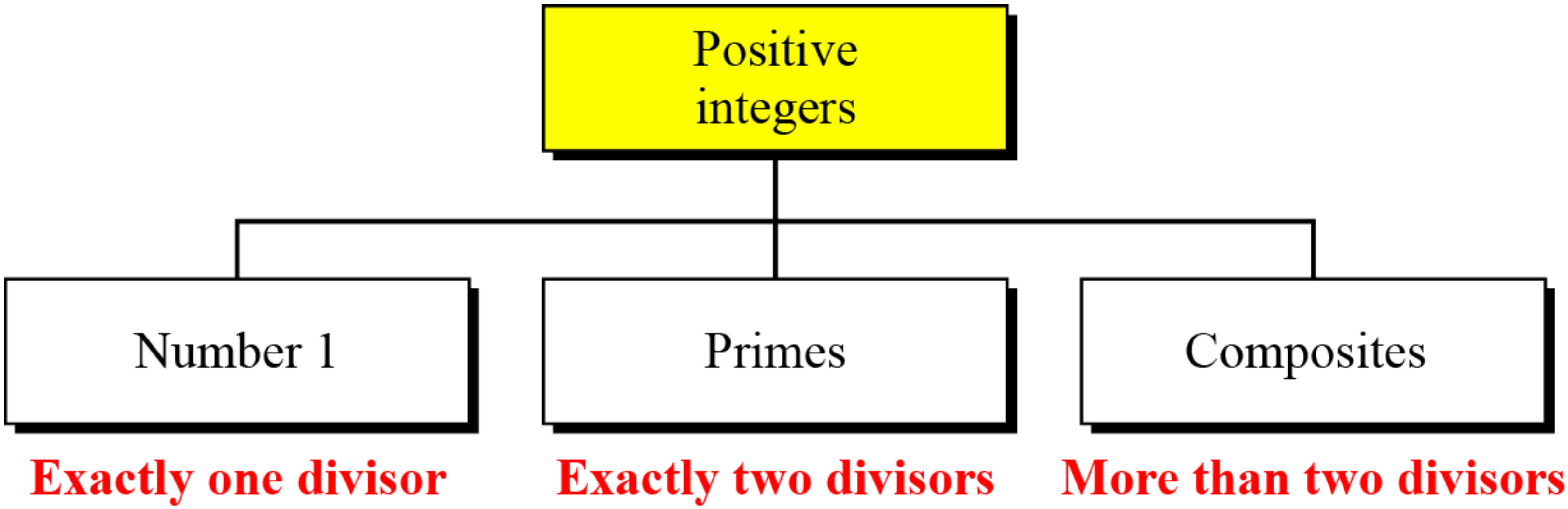
# PRIMOS,
# TEORIA FUNDAMENTAL DE LA ARITMETICA,
# TEOREMA CHINO DEL RESTO
# RSA

CRYPTOGRAPHY AND NETWORK SECURITY
(BEHROUZ FOROUZAN)
MATHEMATICS OF CRYPTOGRAPHY
CAP. 9, 10

# NÚMEROS PRIMOS

# Definición




Un primo es divisible por el mismo y por 1

# Definición

Ejemplo:

- ¿Cuál es el primo más pequeño?

  2

- Liste los primos más pequeños que 10

  10: 2, 3, 5, y 7.

# Definición

- Coprimos:
  - Dos enteros a y b son **coprimos**, o **relativamente primos**, si mcd(a,b)=1
  - Si p es un primo, entonces todos los enteros desde 1 a p-1 son relativamente primos con p.

# Primos: Cardinalidad

- Hay infinitos números primos o existe un límite?

Hay infinitos números primos

- Dado un número n, cuántos primos son más pequeños o menores que n?

  $\Pi(n)$= encuentra números primos más pequeños o iguales a n.

$$\pi(1) = 0 \quad \pi(2) = 1 \quad \pi(3) = 2 \quad \pi(10) = 4 \quad \pi(20) = 8 \quad \pi(50) = 15 \quad \pi(100) = 25$$

$$\boxed{[n / (\ln n)] \quad < \quad \pi(n) \quad < \quad [n/(\ln n - 1.08366)]}$$

# Comprobación de Primalidad

- Dado un número "n" ¿cómo podemos determinar si "n" es un primo?
  - Si el número es divisible por todos los primos menores que

$$\sqrt{n}$$

Ejemplo:

- **Es 97 primo?**
  - $\sqrt{97}$ = 9. Los primos menores a 9 son 2, 3, 5, y 7. Es primo

- **Es 301 primo?**
  - $\sqrt{301}$ = 17. Los primos menores a 17 son 17 2, 3, 5, 7, 11, 13, y 17. 7 divide a 301.

    301 no es primo.

# La criba de Eratosthenes

- Método que encuentra todos los primos menores que n.
- Encontrar todos los números menores que 100.

**Table 9.1** *Sieve of Eratosthenes*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 2 | 3 | ~~4~~ | 5 | ~~6~~ | 7 | ~~8~~ | ~~9~~ | ~~10~~ |
| 11 | ~~12~~ | 13 | ~~14~~ | ~~15~~ | ~~16~~ | 17 | ~~18~~ | 19 | ~~20~~ |
| ~~21~~ | ~~22~~ | 23 | ~~24~~ | ~~25~~ | ~~26~~ | ~~27~~ | ~~28~~ | 29 | ~~30~~ |
| 31 | ~~32~~ | ~~33~~ | ~~34~~ | ~~35~~ | ~~36~~ | 37 | ~~38~~ | ~~39~~ | ~~40~~ |
| 41 | ~~42~~ | 43 | ~~44~~ | ~~45~~ | ~~46~~ | 47 | ~~48~~ | ~~49~~ | ~~50~~ |
| ~~51~~ | ~~52~~ | 53 | ~~54~~ | ~~55~~ | ~~56~~ | ~~57~~ | ~~58~~ | 59 | ~~60~~ |
| 61 | ~~62~~ | ~~63~~ | ~~64~~ | ~~65~~ | ~~66~~ | 67 | ~~68~~ | ~~69~~ | ~~70~~ |
| 71 | ~~72~~ | 73 | ~~74~~ | ~~75~~ | ~~76~~ | ~~77~~ | ~~78~~ | 79 | ~~80~~ |
| ~~81~~ | ~~82~~ | 83 | ~~84~~ | ~~85~~ | ~~86~~ | ~~87~~ | ~~88~~ | 89 | ~~90~~ |
| ~~91~~ | ~~92~~ | ~~93~~ | ~~94~~ | ~~95~~ | ~~96~~ | 97 | ~~98~~ | ~~99~~ | ~~100~~ |

# La función Phi-Euler

- La función phi-Euler, $\phi\,(n)$, encuentra el número de enteros que son más pequeños que $n$ y relativamente primos a $n$.

1. $\phi(1) = 0$.
2. $\phi(p) = p - 1$ if $p$ is a prime.
3. $\phi(m \times n) = \phi(m) \times \phi(n)$ if $m$ and $n$ are relatively prime
4. $\phi(p^e) = p^e - p^{e-1}$ if $p$ is a prime.

# La función Phi-Euler

- Las 4 reglas pueden ser combinadas,

  n puede ser factorizado como

$$n = p_1^{e_1} \times p_2^{e_2} \times \ldots \times p_k^{e_k}$$

$$\boxed{\phi(n) = (p_1^{e_1} - p_1^{e_1-1}) \times (p_2^{e_2} - p_2^{e_2-1}) \times \cdots \times (p_k^{e_k} - p_k^{e_k-1})}$$

La dificultad de encontrar $\phi$(n) depende de la dificultad de encontrar la factorización de *n*.

# La función Phi-Euler

Ejemplos:

- **¿Cuál es el valor de $\phi(13)$?**

  $\phi(13) = (13 - 1) = 12$. (propiedad 2)

- **¿Cuál es el valor de $\phi(10)$?**

  $\phi(10) = \phi(2) \times \phi(5) = 1 \times 4 = 4$ (Propiedad 3)

- **¿Cuál es el valor de $\phi(240)$?**

  $240 = 2^4 \times 3^1 \times 5^1$.

  $\phi(240) = (2^4 - 2^3) \times (3^1 - 3^0) \times (5^1 - 5^0) = 64$ (propiedad 4)

- **Podemos decir que $\phi(49) = \phi(7) \times \phi(7) = 6 \times 6 = 36$?**
  - No, m y n deben ser relativamente primos
  - $49 = 7^2$ : $\phi(49) = 7^2 - 7^1 = 42$ (propiedad 4)

# La función Phi-Euler: Ejemplos

- **Cuál es el número de elementos en $Z_{14}$*?**

  $\phi(14) = \phi(7) \times \phi(2) = 6 \times 1 = 6.$

  Los elementos son: 1, 3, 5, 9, 11, y 13.

Si $n > 2$, el valor de $\phi(n)$ es par

# Pequeño Teorema de Fermat

- Primera versión
  - Si “p” es primo y “a” es un entero, tal que “p” no divide a “a”

$$\boxed{a^{p-1} \equiv 1 \bmod p}$$

- Segunda versión
  - Si “p” es primo y “a” es un entero

$$\boxed{a^{p} \equiv a \bmod p}$$

# Pequeño Teorema de Fermat: Aplicaciones

- **Exponenciación**

Ejemplos:

- Encontrar el resultado de $6^{10}$ mod 11
  - $6^{10}$ mod 11 = 1. (Primera versión p=11)

- Encontrar el resultado de $3^{12}$ mod 11 (segunda versión)

$$3^{12} \bmod 11 = (3^{11} \times 3) \bmod 11 = (3^{11} \bmod 11)(3 \bmod 11) = (3 \times 3) \bmod 11 = 9$$

# Pequeño Teorema de Fermat: Aplicaciones

- **Inversa multiplicativa:**
  - **Si "p" es primo y "a" es un enero y "p" no divide "a",**

$$\boxed{a^{-1} \; mod \; p = a^{p-2} \; mod \; p}$$

Ejemplos:

a. $8^{-1} \bmod 17 = 8^{17-2} \bmod 17 = 8^{15} \bmod 17 = 15 \bmod 17$

b. $5^{-1} \bmod 23 = 5^{23-2} \bmod 23 = 5^{21} \bmod 23 = 14 \bmod 23$

c. $60^{-1} \bmod 101 = 60^{101-2} \bmod 101 = 60^{99} \bmod 101 = 32 \bmod 101$

d. $22^{-1} \bmod 211 = 22^{211-2} \bmod 211 = 22^{209} \bmod 211 = 48 \bmod 211$

# Teorema de Euler

- Primera versión
  - Si a y n son coprimos, entonces

$$\boxed{a^{\phi(n)} \equiv 1 \pmod{n}}$$

- Segunda versión
  - Si n = pxq, a<n, y k un entero

$$\boxed{a^{k \times \phi(n) + 1} \equiv a \pmod{n}}$$

La segunda versión es usada en el RSA

# Teorema de Euler

- **Exponenciación**

Ejemplos:

- $6^{24} \bmod 35$
  - $6^{24} \bmod 35 = 6^{\phi(35)} \bmod 35 = 1.$
- $20^{62} \bmod 77$
  - $k = 1$ (segunda versión),
  - $20^{62} \bmod 77 = (20 \bmod 77)\ (20^{\phi(77)\ +\ 1} \bmod 77) \bmod 77$

    $= (20)(20) \bmod 77 = 15.$

# Teorema de Euler

- **Inversa multiplicativa:**
  - Si "n" y "a" son coprimos:

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

Ejemplos:

a. $8^{-1} \bmod 77 = 8^{\phi(77)-1} \bmod 77 = 8^{59} \bmod 77 = 29 \bmod 77$

b. $7^{-1} \bmod 15 = 7^{\phi(15)-1} \bmod 15 = 7^{7} \bmod 15 = 13 \bmod 15$

c. $60^{-1} \bmod 187 = 60^{\phi(187)-1} \bmod 187 = 60^{159} \bmod 187 = 53 \bmod 187$

d. $71^{-1} \bmod 100 = 71^{\phi(100)-1} \bmod 100 = 71^{39} \bmod 100 = 31 \bmod 100$

# Test de Primalidad

- Algoritmos determinísticos
  - Algoritmo de la divisibilidad
  - Algoritmo AKS
- Algoritmos probabilísticos
  - Test de Fermat
  - Test de la raíz cuadrada
  - Test de Miller-Rabin

# Algoritmo Determinístico

- Algoritmo de la divisibilidad

**Algorithm 9.1** *Pseudocode for the divisibility test*

```
Divisibility_Test (n)                      // n is the number to test for primality
{
  r ← 2
  while (r < √n )
  {
   if (r | n) return "a composite"
   r ← r + 1
  }
  return "a prime"
}
```

# FACTORIZACION

## TEOREMA FUNDAMENTAL DE LA ARITMÉTICA

# Definición

- **Teorema**: Un número entero n>1 ó es primo o puede ser escrito de manera única, como un producto de números primos,

$$n = p_1^{e1} \times p_2^{e2} \times \cdots \times p_k^{ek}$$

Ejemplo:

- $360 = 2^3 \times 3^2 \times 5$

# Definición

- **Aplicación:**
  - Máximo común divisor:

$$a = p_1^{a_1} \times p_2^{a_2} \times \cdots \times p_k^{a_k} \qquad b = p_1^{b_1} \times p_2^{b_2} \times \cdots \times p_k^{bk}$$

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} \times p_2^{\min(a_2, b_2)} \times \cdots \times p_k^{\min(a_k, b_k)}$$

Ejemplo:

- a= 588, b=936

$588 = 2^2.3.7^2$ $\quad 936 = 2^3.3^2.13$ $\quad$ mcd(a,b) = 12

# Métodos de Factorización

- Método de división

**Algorithm 9.3** *Pseudocode for trial-division factorization*

```
Trial_Division_Factorization (n)                // n is the number to be factored
{
    a ← 2
    while (a ≤ √n )
    {
        while (n mod a = 0)
        {
            output a                            // Factors are output one by one
            n = n / a
        }
        a ← a + 1
    }
    if (n > 1) output n                         // n has no more factors
}
```

# Métodos de Factorización

- Método de división

Ejemplo:

- Encuentre los factores de 1233

$$1233 = 3^2 \times 137$$

- Encuentre los factores de 1523357784

$$1523357784 = 2^{\mathbf{3}} \times 3^2 \times 13 \times 37 \times 43987$$

# Métodos de Factorización

- Método de Fermat
- Método de Pollard p-1
- Método  rho Pollard
- Criba cuadrática
- Criba de campos númericos

# TEOREMA CHINO DEL RESTO

# Teorema chino del resto (TCR)

- TCR es usado para resolver un conjunto de ecuaciones de congruencia con una variable, pero con diferentes módulos, que son coprimos entre si

$$
\begin{gathered}
x \equiv a_1 \pmod{m_1} \\
x \equiv a_2 \pmod{m_2} \\
\cdots \\
x \equiv a_k \pmod{m_k}
\end{gathered}
$$

# Teorema chino del resto (TCR)

- Solución:

1. Encontrar $M = m_1 \times m_2 \times \ldots \times m_k$ , M es el módulo común.
2. Encontrar

   $M_1 = M/m_1$,

   $M_2 = M/m_2, \ldots,$

   $M_k = M/m_k$
3. Encontrar la inversa multiplicativa $M_1^{-1}, M_2^{-1}, , M_k^{-1}$ de $M_1, M_2, \ldots, M_k$ usando su correspondiente módulo ($m_1, m_2, \ldots, m_k$).
4. La solución de esta ecuación es:

-

$$x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \cdots + a_k \times M_k \times M_k^{-1}) \bmod M$$

# Teorema chino del resto (TCR)

**Ejemplo:**

$$x \equiv 2 \pmod{3}$$
$$x \equiv 3 \pmod{5}$$
$$x \equiv 2 \pmod{7}$$

- $M = 3 \times 5 \times 7 = 105$
  - $M_1 = 105 / 3 = 35$,
  - $M_2 = 105 / 5 = 21$,
  - $M_3 = 105 / 7 = 15$
- Las inversas son:
  - $M_1^{-1} = 2$,
  - $M_2^{-1} = 1$,
  - $M_3^{-1} = 1$
- $x = (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \bmod 105 = 23 \bmod 105$

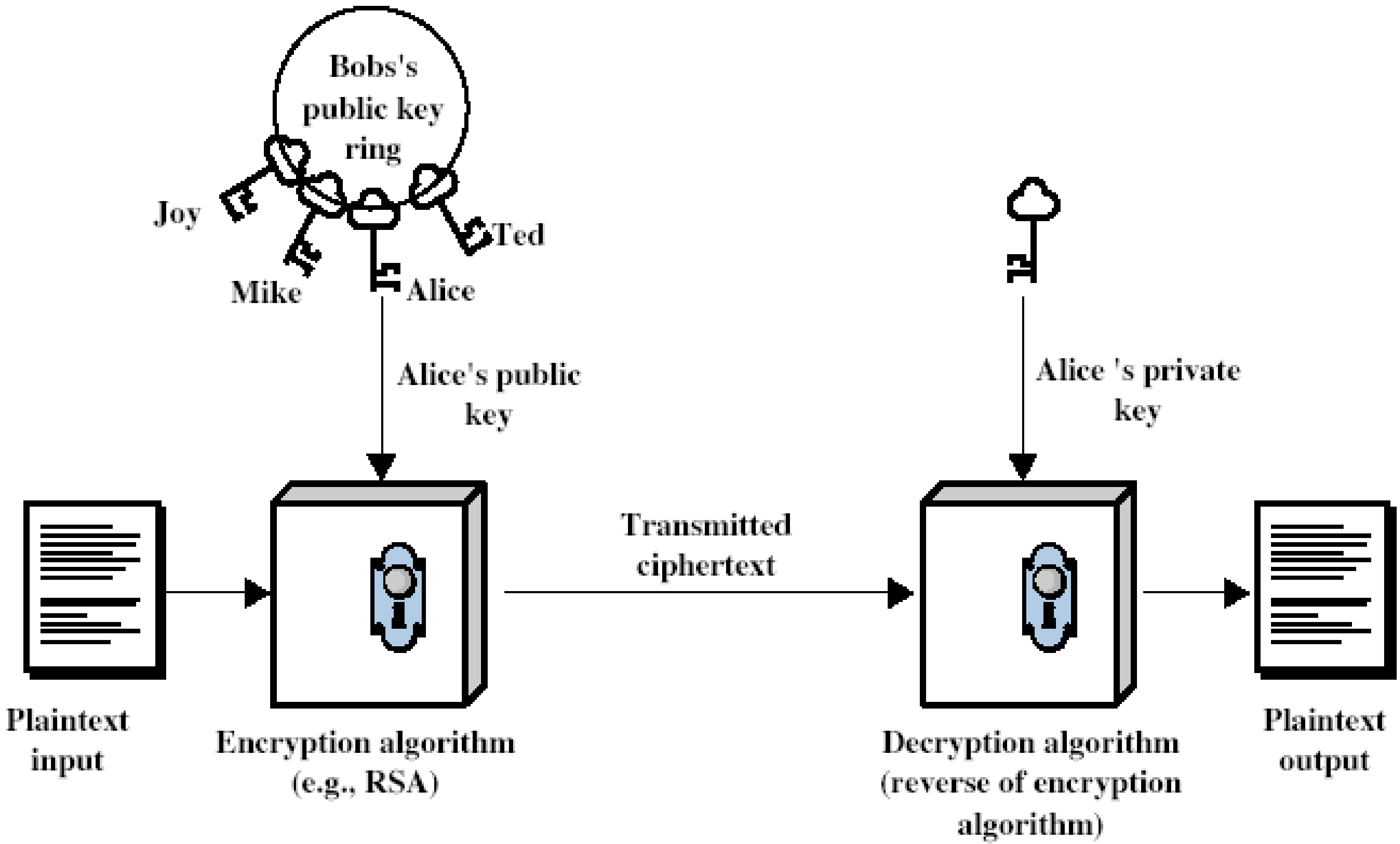
# Teorema chino del resto (TCR)

**Ejemplo:**

- Encuentre un entero que tiene resto 3 cuando es dividido por 7 y 13, pero es divisible por 12

$$x = 3 \bmod 7$$
$$x = 3 \bmod 13$$
$$x = 0 \bmod 12$$

# CRIPTOGRAFIA DE CLAVE PUBLICA


Bobs's
public key
ring
Joy
Mike
Alice
Ted
Alice's public
key
Alice 's private
key
Transmitted
ciphertext
Plaintext
input
Encryption algorithm
(e.g., RSA)
Decryption algorithm
(reverse of encryption
algorithm)
Plaintext
output

# RSA

- Rivest, Shamir & Adleman  of MIT in 1977
- Usa un esquema de clave pública
- Basado en exponenciación módulo “n”
- Usa enteros grandes (Ej. 1024 bits)
- Seguridad debido al costo de factorización de números grandes

# RSA – Generación de claves

- Cada usuario genera una clave pública y privada al:
  - Seleccionar dos primos aleatorios grandes: “p” y¨”q”
  - Calcula N= pxq
    - Note que $\varnothing(N)=(p-1)(q-1)$
  - Selecciona una clave pública aleatoria “e”
    - Donde $1<e<\varnothing(N)$, $\gcd(e,\varnothing(N))=1$
  - Resuelve la ecuación para encontrar la clave privada “d”
    - $e.d=1 \bmod \varnothing(N)$ and $0\leq d\leq N$
  - Publica la clave pública: <e,N>
  - Guarda la clave privada: <d,N>, <d,p,q>

# RSA – Generación de claves

**Algorithm 10.2** *RSA Key Generation*

```
RSA_Key_Generation
{
    Select two large primes p and q such that p ≠ q.
    n ← p × q
    φ(n) ← (p − 1) × (q − 1)
    Select e such that 1 < e < φ(n) and e is coprime to φ(n)
    d ← e^−1 mod φ(n)                  // d is inverse of e modulo φ(n)
    Public_key ← (e, n)                // To be announced publicly
    Private_key ← d                    // To be kept secret
    return Public_key and Private_key
}
```

# RSA – Cifrado, Descifrado

- Para cifrar un mensaje M, el emisor
  - Obtiene la **clave pública** del receptor `<e,N>`
  - calcula: $C=M^e \bmod N$, donde $0 \leq M < N$
- Para descifrar el mensaje C el receptor:
  - Usa su clave privada `<d,N>`
  - calcula: $M=C^d \bmod N$
- Note que el mensaje M debe ser más pequeño que el módulo N (bloques)

# RSA – Cifrado, Descifrado

- Cifrado

**Algorithm 10.3** *RSA encryption*

```
RSA_Encryption (P, e, n)                    // P is the plaintext in Z_n and P < n
{
    C ← Fast_Exponentiation (P, e, n)       // Calculation of (P^e mod n)
    return C
}
```

# RSA – Cifrado, Descifrado

- Descifrado

**Algorithm 10.4** *RSA decryption*

```
RSA_Decryption (C, d, n)                        //C is the ciphertext in Zn
{
    P ← Fast_Exponentiation (C, d, n)          // Calculation of (C^d mod n)
    return P
}
```

# RSA

## Ejemplo 1:

- Bob escoge:
  - p= 7
  - q=11
  - calcula $n$ = 77.
  - El valor de $\phi(n) = (7 - 1)(11 - 1) = 60$.
  - Escoge $e$ =13,
  - d=37 (Note que $e \times d \bmod 60 = 1$ (son inversas) )
- Imagine que Alice quiere enviar un texto plano 5 a Bob. Ella usa e=13 para cifrar 5.

| | | |
|---|---|---|
| Plaintext: 5 | $C = 5^{13} = 26 \bmod 77$ | Ciphertext: 26 |

- Bob recibe el texto cifrado 26 y usa la clave privada 37 para descifrar el texto cifrado:

| | | |
|---|---|---|
| Ciphertext: 26 | $P = 26^{37} = 5 \bmod 77$ | Plaintext: 5 |

# RSA

**Ejemplo 1:**

- Asuma que otra persona, John, quiere enviar un mensaje a Bob. John quiere enviar el texto plano 63, usa la clave pública de Bob

| | | |
|---|---|---|
| Plaintext: 63 | $C = 63^{13} = 28 \bmod 77$ | Ciphertext: 28 |

- Bob recibe el texto cifrado 28 y usa su clave privada 37 para descifrar el texto cifrado

| | | |
|---|---|---|
| Ciphertext: 28 | $P = 28^{37} = 63 \bmod 77$ | Plaintext: 63 |

# RSA

**Ejemplo 2:**

1. Selecciona primos: $p=17$ & $q=11$
2. Calcula $n = pq = 17 \times 11 = 187$
3. Calcula $\varnothing(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Selecciona e : $\gcd(e,160)=1$; escoge $e=7$
5. Determina d: $d.e=1 \pmod{160}$ y ($d < 160$) El valor de $d=23$
6. Publica la clave pública <7,187>
7. Guarda la clave privada <23, 187>

# RSA

**Ejemplo 2:**

- Dado un mensaje M=88

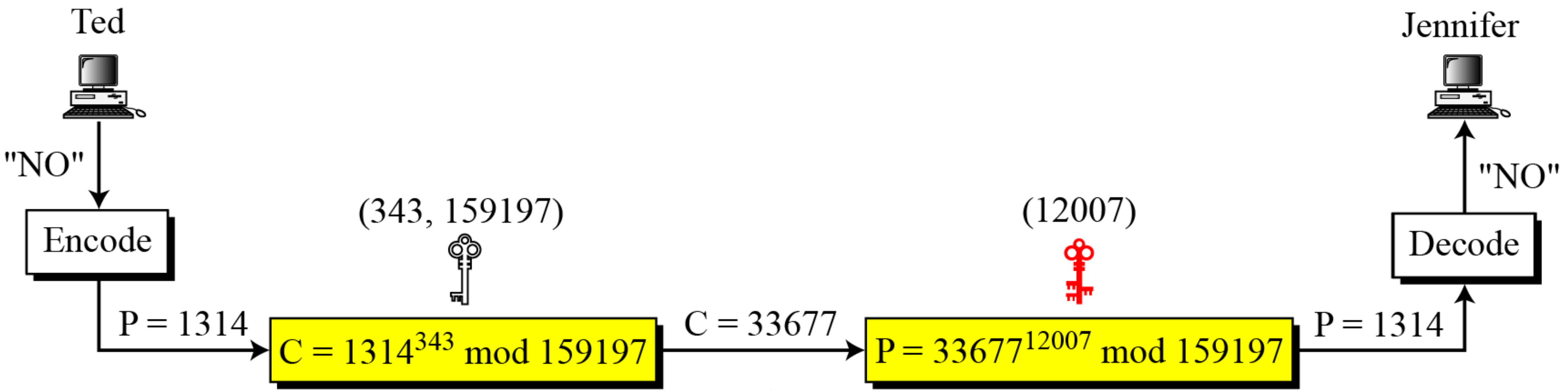
- Cifrado:
  - $C = 88^7 \bmod 187 = 11$

- Descifrado
  - $M = 11^{23} \bmod 187 = 88$

# RSA

## Ejemplo 3:

- Jennifer crea un par de claves. Ella escoge $p = 397$ y $q = 401$. Calcula $n = 159197$. Entonces calcula $\phi(n) = 158400$. Escoge e = 343 y d = 12007. Muestre como Ted puede enviar un mensaje a Jennifer si conoce e y n
- Suponga que Ted quiere enviar el mensaje "NO" a Jennifer.



Ejemplo 4:

# RSA

| | |
|---|---|
| $p =$ | 961303453135835045741915812806154279093098455949962158225831508796<br>479404550564706384912571601803475031209866660649242019180878066742<br>1096063354219926661209 |
| $q =$ | 120601919572314469182767942044508960015559250546370339360617983217<br>314821484837646592153894532091752252732268301071206956046025138871<br>45524969000359660045617 |
| $n =$ | 115935041739676149688925098646158875237714573754541447754855261376<br>147885408326350817276878815968325168468849300625485764111250162414<br>552339182927162507656772727460097082714127730434960500556347274566<br>628060099924037102991424472292215772798531727033839381334692684137<br>327622000966676671831831088373420823444370953 |
| $\phi(n) =$ | 115935041739676149688925098646158875237714573754541447754855261376<br>147885408326350817276878815968325168468849300625485764111250162414<br>552339182927162507656751054233608492916752034482627988117554787657<br>013923444405716989581728196098226361075467211864612171359107358640<br>614008885170265377277264467341066243857664128 |

## Ejemplo 4: RSA

| | |
|---|---|
| $e$ = | 35535 |
| $d$ = | 5800830286003776393609366128967791759466906208965096218042286611138059385282235873170628691003002171085904433840217072986908760061153062025249598844480475682409662470814858171304632406440777048331340108509473852956450719367740611973265574242372176176746207763716420760033708533328853214470885955136670294831 |

- Alicia quiere enviar "THIS IS A TEST"

| | |
|---|---|
| P = | 1907081826081826002619041819 |

- $C = P^e$

| | |
|---|---|
| **C =** | 47530912364622682720636555061054518094237179607049171652323924305445296061319932856661784341835911415119741125200568297979457173603610127821884789274156609048002350719071527718591497518846588863210114835410336165789846796838676373376577746562507928052114814184404814184430812773059004692874248559166462108656 |