



# El algoritmo RSA

## Aula Virtual Crypt4you

Selección del capítulo 14 del Libro Electrónico de Seguridad Informática y Criptografía del mismo autor para **Crypt4you**



Dr. Jorge Ramió Aguirre  
Universidad Politécnica de Madrid

Este archivo forma parte libro electrónico de Seguridad Informática y Criptografía v 4.1 de marzo de 2006. Se autoriza el uso, su reproducción en computador y la impresión sólo con fines docentes o personales, respetando los créditos del autor. Queda prohibida su comercialización.

# Aquí ciframos números, no mensajes

- La operación característica de la cifra asimétrica es mediante un cifrado exponencial. La operación a realizar será  $C = A^B \bmod n$ , en donde  $n$  es el cuerpo de cifra del orden de 1.024 bits,  $B$  es una clave pública 17 bits para el intercambio de clave y cerca de 1.024 bits de la clave privada para firma digital.  $A$  será siempre un número  $N$  (nunca un mensaje  $M$ ) y por lo general del orden de las centenas de bits.
- Esto es así porque este tipo de cifra es muy lenta y sería muy costoso en tiempo cifrar, por ejemplo, mensajes de cientos o miles de bytes.
- Por lo tanto, cuando se cifre con la clave pública de destino para hacer un intercambio de clave, se tratará de un número  $N$  del orden de los 128 bits (la clave de sesión), y cuando se cifre con la clave privada de emisión para una firma digital, se tratará de un número  $N$  de 160 bits, por ejemplo un hash SHA-1 sobre el mensaje  $M$ .

## Otros casos de cifra exponencial

- La cifra con la clave privada de recepción cuando desciframos un número o dato que se nos ha enviado confidencialmente, o bien la cifra con la clave pública del emisor para comprobar así su firma digital, serán casos de descifrado.
- En el primero de ellos, puesto que se recibe un número muy grande dentro del cuerpo de cifra con  $n$  bits y la clave privada será también de esa magnitud, en el caso de RSA se realizará el descifrado usando el Teorema del Resto Chino.
- Si deseamos cifrar mensajes  $M$  con estos algoritmos, se puede hacer formando bloques de cifra, al igual que se hace con los sistemas simétricos, pero recuerde que esto tiene sentido sólo para prácticas de laboratorio y nunca en sistemas reales.

# Cifrado exponencial con clave del receptor

- Al cifrar el número  $N$  y en el descifrado del criptograma  $C$  se usará una exponenciación:  $E_e(N) = C$  y  $E_d(C) = N$ .
- En la operación de cifrado, el subíndice  $e$  significará el uso de la **clave pública del receptor** ( $R$ ) en el extremo emisor y el subíndice  $d$  el uso de la **clave privada del receptor** ( $R$ ) en el extremo receptor.

$$C = E_{eR}(N) = N^{eR} \bmod n_R \Rightarrow N = E_{dR}(C) = C^{dR} \bmod n_R$$

- $N$  deberá ser un elemento del CCR de  $n_R$ .
- Esta operación se usará para realizar el intercambio de una clave de sesión entre un emisor y un receptor.

# Cifrado exponencial con clave del emisor

- En la operación de cifrado el subíndice **d** significa el uso de la **clave privada del emisor** (E) en el extremo emisor, y el subíndice **e** el uso de la **clave pública del emisor** (E) en el extremo receptor.

$$C = E_{dE}(N) = N^{dE} \bmod n_E \Rightarrow N = E_{eE}(C) = C^{eE} \bmod n_E$$

- N deberá ser un elemento del CCR de  $n_E$ .
- Esta operación se usará para autenticar la identidad de un usuario mediante una firma digital, al mismo tiempo que se demuestra la integridad del mensaje mediante un hash.

# Cifrado exponencial genérico tipo RSA

Sea el grupo de trabajo  $n = p * q \Rightarrow \phi(n) = (p-1)(q-1)$

Se eligen una clave pública  $e$  y una privada  $d$  de forma que:  
 $e * d \bmod \phi(n) = 1 \Rightarrow e * d = k(p-1)(q-1) + 1.$

Si  $e * d = k\phi(n) + 1$

Por el Teorema de Euler se tiene que:

$$N^{k\phi(n)} \bmod n = 1$$

para todo  $N$  primo con  $n$

y ...

Por el Teorema del Resto Chino se tiene que:

$$N^{ed} = N \bmod n$$

$$\text{ssi } \begin{cases} N^{ed} = N \bmod p \\ N^{ed} = N \bmod q \end{cases}$$

Luego, el sistema de cifra será válido para cualquier valor de  $N$

# Operación de descifrado exponencial

Al cifrar el número  $N$  con una clave pública  $e$  (en este caso para realizar un intercambio de clave, aunque es igual de válido con una clave  $d$  en caso de firma digital) tenemos:

*Cifrado:*  $C = N^e \bmod n$

*Descifrado:*  $C^d \bmod n = (N^e)^d \bmod n = N^{ed} \bmod n$

$$C^d \bmod n = N^{k\phi(n)+1} \bmod n = N * N^{k\phi(n)} \bmod n$$

$$C^d \bmod n = N * 1 \bmod n = N \bmod n$$

Por lo tanto, la operación  $C^d \bmod n$  recuperará el número  $N$ .

# Comprobación de la recuperación de N

Sea  $n = p * q = 5 * 11 = 55 \Rightarrow \phi(n) = (5-1)(11-1) = 40$

Sea el número  $N = 50 = 2 * 5^2$  (*debe ser un elemento de  $n = 55$* )

Se elige  $e = 3 \Rightarrow d = \text{inv}[e, \phi(n)] = \text{inv}(3, 40) = 27$

$e * d \text{ mod } \phi(n) = 3 * 27 \text{ mod } 40 = 81 \text{ mod } 40 = 1$

$C = N^e \text{ mod } n = 50^3 \text{ mod } 55 = (2 * 5^2)^3 \text{ mod } 55$

$C = [(2)^3 \text{ mod } 55 * (5^2)^3 \text{ mod } 55] \text{ mod } 55$  - *por reducibilidad* ↓ -

$N = C^d \text{ mod } n = \{[(2)^3 \text{ mod } 55 * (5^2)^3 \text{ mod } 55] \text{ mod } 55\}^{27} \text{ mod } 55$

$N = [(2)^{3*27} \text{ mod } 55 * (5^2)^{3*27} \text{ mod } 55] \text{ mod } 55$

$N = [2^{2\phi(n)+1} * 5^{2\phi(n)+1} * 5^{2\phi(n)+1}] \text{ mod } 55$

Por el Teorema de Euler y del Resto Chino  $= 2 * 5 * 5 \text{ mod } 55 = 50$



# Algoritmo de cifra asimétrica RSA

En febrero de 1978 Ron Rivest, Adi Shamir y Leonard Adleman proponen un algoritmo de cifra de clave pública: **RSA**

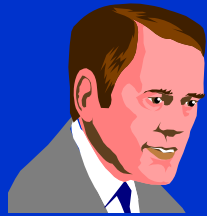
## Pasos del algoritmo

1. Cada usuario elige un grupo  $n = p \cdot q$  (pueden y de hecho son distintos).
2. Los valores  $p$  y  $q$  no se hacen públicos.
3. Cada usuario calcula  $\phi(n) = (p-1)(q-1)$ .
4. Cada usuario elige una clave pública  $e$  de forma que  $1 < e < \phi(n)$  y que cumpla con la condición:  $\text{mcd}[e, \phi(n)] = 1$ .
5. Cada usuario calcula la clave privada  $d = \text{inv}[e, \phi(n)]$ .
6. Se hace público el grupo  $n$  y la clave  $e$ .
7. Se guarda en secreto la clave  $d$ . También guardará  $p$  y  $q$  puesto que en la operación de descifrado usará el Teorema del Resto Chino.

**Cifra:**  $C = N^{eR} \bmod n_R$

**Firma:**  $C = h(M)^{dE} \bmod n_E$

# Intercambio de clave RSA (B → A)



Benito

En el protocolo intercambiaremos una clave K

Sea  $K = DA9F$  (16 bits)



Adela

**Claves Benito**  
 $n_B = 65.669$   
 $e_B = 35, d_B = 53.771$

$2^{16} < 66.331 < 2^{17}$   
 Forzaremos cifrar un  
 bloque de 16 bits

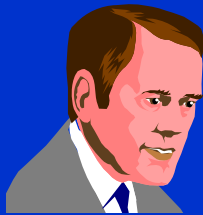
**Claves Adela**  
 $n_A = 66.331$   
 $e_A = 25, d_A = 18.377$

**Cifra**  $K = DA9F_{16} = 55.967_{10}$   
 $C = K^{e_A} \text{ mod } n_A$   
 $C = 55.967^{25} \text{ mod } 66.331 = 16.667$

Benito envía a Adela  $C = 16.667$

En la práctica no habrá que forzar este tamaño ya que la cifra asimétrica se hace en un cuerpo (más de mil bits) mucho mayor que el número que se cifra (cientos de bits).

# Recuperación de la clave K por A



Benito

## Claves Benito

$$n_B = 65.669$$

$$e_B = 35, d_B = 53.771$$

## Claves Adela

$$n_A = 66.331$$

$$e_A = 25, d_A = 18.377$$



Adela

Teníamos que:  $K = DA9F_{16} = 55.967_{10}$

$$C = K^{e_A} \text{ mod } n_A \quad C = 55.967^{25} \text{ mod } 66.331 = 16.667$$

Benito había enviado a Adela  $C = 16.667$

Adela calcula:

- $C^{d_A} \text{ mod } n_A = 16.667^{18.377} \text{ mod } 66.331 = 55.967.$
- El intercambio de clave se ha realizado con confidencialidad porque sólo Adela ha podido realizar ese cálculo con su clave privada  $d_A$ .

Los primos que ha usado Benito son (97, 677) y los de Adela (113, 587)

# Descifrado con números grandes

Grupo  $n = 91 = 7 * 13$ ;  $\phi(n) = \phi(7 * 13) = (7-1)(13-1) = 72$      $N = 48$

Elegimos  $e = 5$  pues  $\text{mcd}(5, 72) = 1$      $\therefore d = \text{inv}(5, 72) = 29$

**CIFRADO:**

$C = N^e \text{ mod } n = 48^5 \text{ mod } 91 = 5245.803.968 \text{ mod } 91 = 55$

**DESCIFRADO:**

$N = C^d \text{ mod } n = 55^{29} \text{ mod } 91 = 48$  ...  $55^{29}$  ya es “*número grande*”

$55^{29}$  es un número con 51 dígitos...

$55^{29} = 295473131755644748809642476009391248226165771484375$

¿Cómo podemos acelerar esta operación?

1ª opción: usar reducibilidad



2ª opción: algoritmo exp. rápida



Opción óptima: usar el Teorema del Resto Chino



# Uso del Teorema del Resto Chino en RSA

- Normalmente la clave pública  $e$  de RSA es un valor bastante bajo, por ejemplo  $2^{16} + 1$  (un valor típico). Luego, en el proceso de cifra (no en la firma) no tendremos problemas con la velocidad de cifra porque el exponente  $e$  será relativamente bajo, en este caso 17 bits.
- Como el cuerpo de trabajo  $n = p \cdot q$  es mucho mayor, del orden de  $2^{1.024}$  si hablamos de claves de 1.024 bits, entonces la clave privada  $d$  será por lo general mucho mayor que el valor de  $e$  y caerá muy cerca de ese valor de 1.024 bits. Por lo tanto, podría ser costoso para el receptor descifrar algo con su clave privada o firmar digitalmente un documento con dicha clave privada.
- La solución está en aplicar el Teorema del Resto Chino: en vez de trabajar en  $n$ , lo haremos en  $p$  y  $q$  por lo que las exponenciaciones modulares se harán en  $p$  y  $q$ , mucho más rápido que hacerlo en  $n$ .

# Descifrado RSA aplicando el TRC

$N = C^d \bmod n$  Aplicando el Teorema del Resto Chino:

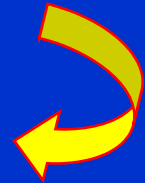
$$N = \{A_p[C_p^{d_p} \pmod{p}] + A_q[C_q^{d_q} \pmod{q}]\} \bmod n$$

con:  $A_p = q [\text{inv}(q, p)] = q^{p-1} \bmod n$

$$A_q = p [\text{inv}(p, q)] = p^{q-1} \bmod n$$

$$d_p = d \bmod (p-1) \qquad d_q = d \bmod (q-1)$$

$$C_p = C \bmod p \qquad C_q = C \bmod q$$



Se hacen más operaciones pero el tiempo de cálculo total es menor dado que los valores  $d_p$ ,  $d_q$ ,  $A_p$  y  $A_q$  están **precalculados**. Las operaciones  $C_p$  y  $C_q$  son sencillas y muy rápidas. El único cálculo que consume tiempo será  $C_p^{d_p}$  y  $C_q^{d_q}$  pero ambos se hacen en cuerpos mucho menores que  $n$ .

# Ejemplo de descifrado RSA usando el TRC

Sea:  $p = 89, q = 31, n = p \cdot q = 89 \cdot 31 = 2.759, \phi(n) = 88 \cdot 30 = 2.640$

Elegimos  $e = 29 \Rightarrow d = \text{inv}[e, \phi(n)] = \text{inv}[29, 2.640] = 2.549$

Si el número a cifrar es  $N = 1.995$ , entonces:

$$C = N^e \bmod n = 1.995^{29} \bmod 2.759 = 141$$

$$N = C^d \bmod n = 141^{2.549} \bmod 2.759 = 1.995$$

$$A_p = q^{p-1} \bmod n = 31^{88} \bmod 2.759 = 713$$

$$A_q = p^{q-1} \bmod n = 89^{30} \bmod 2.759 = 2.047$$

$$d_p = d \bmod (p-1) = 2.549 \bmod 88 = 85$$

$$d_q = d \bmod (q-1) = 2.549 \bmod 30 = 29$$

$$C_p = C \bmod p = 141 \bmod 89 = 52$$

$$C_q = C \bmod q = 141 \bmod 31 = 17$$

Reemplazando en:  $N = \{A_p[C_p^{d_p} \bmod p] + A_q[C_q^{d_q} \bmod q]\} \bmod n$

$$N = \{713[52^{85} \bmod 89] + 2.047[17^{29} \bmod 31]\} \bmod 2.759$$

$$N = \{713 \cdot 37 + 2.047 \cdot 11\} \bmod 2.759 = (26.381 + 22.517) \bmod 2.759 = 1.995$$

# Ataque a la clave por factorización de $n$

¿Qué fortaleza tendrá este algoritmo ante ataques?

- ➡ El intruso que desee conocer la clave secreta  $d$  a partir de los valores  $n$  y  $e$  se enfrentará al Problema de la Factorización de Números Grandes (PFNG), puesto que la solución para conocer esa clave privada es conocer primero el valor del Indicador de Euler  $\phi(n) = (p-1)(q-1)$  para así poder encontrar  $d = \text{inv} [e, \phi(n)]$ , pero para ello deberá saber los valores de los primos  $p$  y  $q$ .
- ➡ La complejidad asociada al PFNG para un número  $n$  viene dada por la ecuación  $e^{\sqrt{\ln(n)} \ln \ln(n)}$ , donde  $\ln$  es logaritmo natural.
- ➡ Le recomiendo se descargue de este sitio el programa **factor.exe** en entorno MS-DOS. No obstante, existirán otros ataques a RSA que no requieren factorizar un número grande.



# Tiempo necesario para afrontar el PFNG

Para un procesador de  $2 \times 10^8$  instrucciones por segundo (años noventa). Fuente: Criptografía Digital, José Pastor. Prensas Univ. de Zaragoza, 1998.

Nº de bits (n)	Nº de dígitos	Días	Años
60	18	$1,7 \times 10^{-8}$	-
120	36	$1,5 \times 10^{-5}$	-
256	77	1,0	-
363	109	$9,0 \times 10^2$	2,5
442	133	$9,4 \times 10^4$	$2,5 \times 10^2$
665	200	$3,8 \times 10^8$	$1,0 \times 10^6$



Desafío RSA640 (193 dígitos) roto en noviembre de 2005 en la Universidad de Bonn. Lo que en 1998 se valoraba en un millón de años, hoy se ha roto en un tiempo equivalente a 30 años con un PC a 2,2 GHz. Y se resolverán nuevos desafíos de números mayores. Debemos ser siempre muy cautos.

# Tamaño de los parámetros en RSA

Toda la seguridad de RSA está basada en sus parámetros: los primos  $p$  y  $q$  y los valores de sus claves pública  $e$  y privada  $d$ .

El cuerpo de trabajo debe ser al menos de 1.024 bits con primos  $p$  y  $q$  de al menos 500 bits y que difieran unos cuantos dígitos.

Aunque la clave pública debe ser pequeña para facilitar así las operaciones, su valor no puede ser excesivamente bajo. Se usará el número 4 de Fermat  $F_4 = 2^{2^4} + 1 = 2^{16} + 1 = 65.537$ .

Como  $ed \bmod \phi(n) = 1$ , esto hace que la clave privada  $d$  sea un número superior a los 1.000 bits, por lo general cerca de 1.024.

Habrà que prestar también especial atención en la generación de dichos primos y la posterior comprobación de su primalidad.

# El problema en la elección del valor de $n$


Si  $p$  y  $q$  son muy cercanos, puede ser fácil factorizar  $n$

- ☞ Si  $p \approx q$  y suponemos que  $p > q$ , entonces  $(p-q)/2$  es un entero muy pequeño y por otra parte  $(p+q)/2$  será un entero ligeramente superior a  $\sqrt{n}$ .
- ☞ Además se cumplirá que:  $n = (p+q)^2/4 - (p-q)^2/4$ . Esto lo podemos escribir como  $n = x^2 - y^2 \Rightarrow y^2 = x^2 - n$
- ☞ Elegimos enteros  $x > \sqrt{n}$  hasta que  $(x^2 - n)$  sea cuadrado perfecto. En este caso  $x = (p+q)/2$ ;  $y = (p-q)/2$ . Por lo tanto rompemos el valor  $n$ :  $p = (x+y)$ ;  $q = (x-y)$ . ☝

## Ejemplo de mala elección del valor de n

- Sea  $p = 181$ ;  $q = 251 \Rightarrow n = 181 * 251 = 45.431$
- Como  $\sqrt{45.431} = 213,14$  buscaremos valores enteros de  $x$  mayores que 213 de forma que  $(x^2 - 45.431)$  sea un cuadrado perfecto  $\downarrow$

1.  $x = 214 \Rightarrow x^2 - 45.431 = 365 \quad \therefore \sqrt{365} = 19,10 \quad \text{☹}$
2.  $x = 215 \Rightarrow x^2 - 45.431 = 794 \quad \therefore \sqrt{794} = 28,17 \quad \text{☹}$
3.  $x = 216 \Rightarrow x^2 - 45.431 = 1.225 \quad \therefore \sqrt{1.225} = 35 \quad \text{☺}$

Entonces:  $p = x - y = 216 - 35 = 181$    
 $q = x + y = 216 + 35 = 251$

Para evitar otros problemas, es recomendable usar los denominados primos seguros.



# Elección de los números primos

Los valores primos deben elegirse apropiadamente:

## Sistema RSA

- $p$  y  $q$  deben diferir en unos pocos dígitos.  
Recuerde que la relación bit/dígito es  $\approx 3,3$ .
- $p$  y  $q$  no deben ser primos muy cercanos.
- Longitud mínima de  $p$  y  $q$ : 500 bits.
- Valores de  $(p-1)$  y  $(q-1)$  del Indicador de Euler con factores primos grandes.
- El mcd entre  $(p-1)$  y  $(q-1)$  debe ser pequeño.



Esto se cumple con los denominados primos seguros  $\longrightarrow$

# Cálculo de números primos $p$ y $q$ seguros

Se elige  $r$  un primo grande de modo que:  $2*r + 1 = p$

Se elige un  $r'$  primo algo mayor que  $r$  de modo que:  $2*r' + 1 = q$

EJEMPLO: Sean  $r = 1.019$  y  $r' = 3.863$

$p = 2*1.019 + 1 = 2.039$  (11 bits)      **Es primo** 👍

$q = 2*3.863 + 1 = 7.727$  (13 bits)      **Es primo** 👍

$n = p*q = 15.755.353$

Luego:  $p-1 = 2.038$ ;       $q-1 = 7.726$

$p-1 = 2*1.019$ ;       $q-1 = 2*3.863 \Rightarrow \text{mcd}(p-1, q-1) = 2$

Los primos  $p$  y  $q$  cumplen la condición de primos seguros

**Nota:** es posible que encuentre algún documento donde proponga elegir un valor  $r$  primo y comprobar luego si  $p = 2r+1$  y  $q = 2p+1$  son primos. En este caso  $p$  y  $q$  seguirán siendo primos seguros pero sólo de forma **independiente**. Aquí será muy fácil atacar el valor  $n$  factorizándolo a través de una ecuación de segundo grado.

# Par de primos seguros pero independientes

Elegimos  $r$  primo. Comprobamos primero que  $p = 2r+1$  es primo y luego que  $q = 2p+1$  también es primo.

Los valores de  $p$  y  $q$  serán primos seguros pero en el sistema RSA basado en  $n = p*q$  no servirán como pareja segura dado que:

$$n = p*q = [2r + 1][2p + 1] = [2r + 1][2(2r + 1) + 1] = [2r + 1][4r + 3]$$

$$n = 8r^2 + 10r + 3 \Rightarrow 8r^2 + 10r + (3 - n) = 0$$

$$\text{Luego: } r = [-10 \pm \sqrt{100 - 32(3-n)}]/16 = [-10 \pm \sqrt{4 + 32n}]/16$$

$$r = [-10 + \sqrt{4 + 32n}]/16$$

Conocido el valor de  $r$  podemos calcular  $p$  y  $q$  😞.

**Ejemplo:**  $r = 41$ ,  $p = 2r+1 = 83$  👍,  $q = 2p+1 = 167$  👍,  $n = 13.861$ .

$$r = [-10 + \sqrt{4 + 32*13.861}]/16 = [-10 + 666]/16 = 41.$$

# Claves privadas parejas en RSA

Una clave privada  $p$  pareja CPP  $d_p$ , permite descifrar el criptograma  $C$  resultado de una cifra con la clave pública  $e$  sin que  $d_p$  sea el inverso de la clave pública  $e$ . En el sistema RSA habrá como mínimo una clave  $d_p$  pareja de la clave privada  $d$ .

Esto se debe a que las claves inversas  $e$  y  $d$  lo serán en  $\phi(n)$  y en cambio la cifra se hace en el cuerpo  $n$ .

## Ejemplo:

Si  $p = 13$ ;  $q = 19$ ;  $n = 247$ ,  $\phi(n) = 216$  y elegimos  $e = 41$ , entonces  $d = \text{inv}(41, 216) = 137$ , que es único. Si ciframos con la clave pública el número  $N = 87$  obtenemos  $C = 87^{41} \bmod 247 = 159$ .

Luego sabemos que  $N = C^d \bmod n = 159^{137} \bmod 247 = 87$  ✌️

Pero también lo desciframos con  $d_p = 29, 65, 101, 173, 209$  y  $245$ .



## Número de claves privadas parejas

Si  $\gamma = \text{mcm} [(p-1), (q-1)]$  y sea  $d_\gamma = e^{-1} \bmod \gamma = \text{inv}(e, \gamma)$

La clave pública  $e$  tendrá  $\lambda$  claves parejas  $d_i$  de la forma:

$$\begin{aligned} d_i &= d_\gamma + i \gamma & 1 < d_i < n \\ i &= 0, 1, \dots, \lambda & \lambda = \lfloor (n - d_\gamma) / \gamma \rfloor \end{aligned}$$

En el ejemplo anterior tenemos que:

$$\gamma = \text{mcm} [(p-1), (q-1)] = \text{mcm} (12, 18) = 36$$

Luego:  $d_\gamma = \text{inv}(41, 36) = 29$ , así  $d_i = d_\gamma + i \gamma = 29 + i * 36$

Es decir  $d_i = 29, 65, 101, 137, 173, 209, 245$ . Observe que en aparece (137) la clave privada  $d$  y comprobamos que:

$$\lambda = \lfloor (n - d_\gamma) / \gamma \rfloor = \lfloor (247 - 29) / 36 \rfloor = 6,05 = 6$$

## Casos de claves privadas parejas

Sea  $p = 751$ ,  $q = 1.009$ ;  $e = 13$

Clave privada:  $d = 407.077$

Nº de claves privadas parejas: **5**

29.077, 155.077, 281.077, 533.077,  
659.077.

Sea  $p = 751$ ,  $q = 1.009$ ;  $e = 101$

Clave privada: 553.901

Nº de claves privadas parejas: **5**

49.901, 175.901, 301.901, 427.901,  
679.901.

Para otros valores de  $e$ , siempre  
existe una separación entre todas  
las claves privadas igual a **126.000**.

Sea  $p = 379$ ,  $q = 1.783$ ;  $e = 71$

Clave privada:  $d = 531.287$

Nº de claves privadas parejas: **53**

7.379, 19.853, 32.327, 44.801,  
57.275, 69.749, 82.223, 94.697,  
107.171, 119.645, 132.119,  
144.593, 157.067, 169.541, ...  
... 506.339, 518.813, 543.761,  
556.235, 568.709, 581.183,  
593.657, 606.131, 618.605,  
631.079, 643.553, 656.027,  
668.501. ... Y separadas **12.474**.

Sea  $p = 379$ ,  $q = 1.783$ ;  $e = 131$

Ahora las CPP aumentan a **54**.

# Minimizando las claves privadas parejas

Para que  $\lambda$  sea lo más pequeño posible ( $\lambda = 1$ ) un primer paso es elegir los primos  $p$  y  $q$  como primos seguros.

Ejemplo:

Sean  $r' = 5$ ;  $r'' = 23 \Rightarrow p = 2*5 + 1 = 11$  (es primo 👍)

$q = 2*23 + 1 = 47$  (es primo 👍)

En estas condiciones con  $n = 517$  y  $\phi(n) = 460$ , sea  $e = 17$

Luego  $\gamma = \text{mcm}(10, 46) = 230$  y  $d_\gamma = \text{inv}(17, 230) = 203$

Entonces  $\lambda = \lfloor (n - d_\gamma) / \gamma \rfloor = \lfloor (517 - 203) / 230 \rfloor = 1,36 = 1$

Así:  $d_i = d_\gamma + i \gamma = 203 + i*230 = 203, 433 \Rightarrow \lambda = 1$

En efecto,  $d = \text{inv}[e, \phi(n)] = \text{inv}(17, 460) = 433$  y lo cifrado con  $e = 17$  también se descifra con  $d_p = 203$ .

# Minimizando no sólo con primos seguros

Para que  $\lambda$  sea igual a la unidad, habrá que elegir además un valor adecuado de clave pública:

Tomando el mismo ejemplo anterior:

$$p = 11; q = 47; n = 517 \text{ y } \phi(n) = 460$$

Según el valor que elijamos de clave pública  $e$ , podríamos obtener más de una clave privada pareja:

- Sea:  $e = 7, d = 263$   $\gamma = 230$  y  $d_\gamma = \text{inv}(7, 230) = 33$   
 $d_i = d_\gamma + i \gamma = 33 + i*230 = 33, 263, 493 \Rightarrow \lambda = 2$
- Sea:  $e = 77, d = 233$   $\gamma = 230$  y  $d_\gamma = \text{inv}(77, 230) = 3$   
 $d_i = d_\gamma + i \gamma = 3 + i*230 = 3, 233, 463 \Rightarrow \lambda = 2$

Con primos seguros, el número de claves parejas será siempre bajo.

# ¿Preocupado por claves privadas parejas?

Si bien al generar claves RSA con librerías actuales como Crypto++ de Wei Dai (OpenSSL) aparecen claves que no pueden considerarse como óptimas ya que no se controla este hecho, hay que tener en mente que las claves privadas parejas tendrán siempre valores muy cercanos al cuerpo de  $\phi(n)$  es decir un tamaño del orden de  $2^n$  bits.

Por lo tanto, independientemente de la distribución, se trataría de una búsqueda en un cuerpo cercano a  $2^n$  bits, en la actualidad en  $2^{1024}$  bits, es decir un valor inmenso para la capacidad de cómputo actual, incluso suponiendo un ataque similar al del DES Challenge III y un cálculo de claves por segundo varios órdenes de magnitud superior.

No obstante, en todos estos temas siempre hay que estar en alerta pues en cualquier momento puede aparecer algún método óptimo de ataque.

# Claves parejas de la clave pública en RSA

Al trabajar en un cuerpo finito y con iguales opciones de cifra con la clave pública  $e$  y la clave privada  $d$ , tenemos que las ecuaciones vistas en las diapositivas anteriores son válidas en este entorno, cambiando  $d$  por  $e$ .

¿Tiene alguna importancia esto?



No es un problema puesto que todo el mundo conoce la clave pública y el sistema sigue siendo igual de seguro.

Se cumple que los valores de dichas claves parejas son similares y equivalentes en ambos entornos, el de las claves públicas y el de las claves privadas.



## Ejemplo de firma y claves parejas de e

Retomamos el primer ejemplo de claves privadas parejas con:

$$p = 13; q = 19; n = 247, \phi(n) = 216, e = 41, d = 137$$

Si firmamos  $N = 24$ , obtenemos  $C = 24^{137} \bmod 247 = 215$

Luego sabemos que  $N = C^e \bmod n = 215^{41} \bmod 247 = 24$

Como  $e_\gamma = \text{inv}(d, \gamma) = \text{inv}(137, 36) = 5$ , entonces:

$$\lambda = \lfloor (n - e_\gamma) / \gamma \rfloor = \lfloor (247 - 5) / 36 \rfloor = 6,72 = 6$$

$$e_i = e_\gamma + i \gamma = 5 + i * 36 = 5, 41, 77, 113, 149, 185, 221$$

Y se podrá comprobar el criptograma C de la firma con cualquiera de estas claves, parejas de la clave pública e.

# Comprobación de una firma digital

## ¿Problemas con la firma digital y las claves públicas parejas?

- ✓ Un usuario firma un hash de un mensaje con su clave privada, lo que envía al destino junto con el mensaje original. En destino se descifra con la clave pública del emisor y se comparan los dos hash, el de emisión y el recuperado en recepción, para dar validez a dicha firma.
- ✓ En este escenario, esto significa que se podría dar por válida una firma al descifrar el hash recibido con una clave pública pareja  $e'$  distinta a la inversa de la usada en emisión y dar validez a dicha firma; es decir usando alguna de las claves públicas parejas.
- ✓ Esto en sí no es un ataque por lo que, al menos en este contexto y en principio, no debería considerarse como una vulnerabilidad. Esto es así porque, además, como se ha dicho es típico que la clave pública sea el mismo número primo para todos, el valor  $e = 65.537 = 2^{16} + 1$ . Como es obvio, lo que será **distinto** para cada par de claves son  $p$  y  $q$ .



# Números no cifrables en RSA

- ❖ Si  $N^e \bmod n = N$  se dice que  $N$  es un número no cifrable, NNC. Aunque la clave  $e$  sea válida, el número  $N$  se enviará en claro ☹.
- ❖ En RSA habrá como mínimo 9 números no cifrables.
- ❖ En el caso más crítico, todos los números del cuerpo  $n$  pueden ser no cifrables como veremos más adelante.
- ❖ Para conocer estos valores no cifrables, habrá que hacer un ataque de cifrado por fuerza bruta en  $p$  y  $q$ , es decir deberemos comprobar que  $X^e \bmod p = X$  y  $X^e \bmod q = X$  con  $1 < X < n-1$  ☹.

Ejemplo:

Sea el cuerpo  $n = 35$  ( $p = 5$ ,  $q = 7$ ), con  $\phi(n) = 24$  y  $e = 11$ .

Dentro de los números posibles  $\{0, 34\}$  serán no cifrables:  $\{6, 14, 15, 20, 21, 29, 34\}$  además de los obvios  $\{0, 1\}$ . El valor  $n-1$  (en este caso 34) será también siempre no cifrable.

# Cantidad de números no cifrables

La cantidad de números no cifrables dentro de un cuerpo  $n$  será:

$$\sigma_n = [1 + \text{mcd}(e-1, p-1)][1 + \text{mcd}(e-1, q-1)]$$

Los números no cifrables serán:

$$N = [q\{\text{inv}(q, p)\}N_p + p\{\text{inv}(p, q)\}N_q] \text{ mod } n$$

con:  $N_p$  las soluciones de  $N^e \text{ mod } p = N$

$N_q$  las soluciones de  $N^e \text{ mod } q = N$

Esto último debido al TRC puesto que  $N^e \text{ mod } n = N$

En el ejemplo anterior se da el caso mínimo:  $\downarrow \quad \downarrow$  Valores para un mínimo

$$\sigma_n = [1 + \text{mcd}(10, 4)][1 + \text{mcd}(10, 6)] = (1+2)(1+2) = 9$$

$$N^{11} \text{ mod } 5 = N \Rightarrow N_5 = \{0, 1, 4\} \quad N^{11} \text{ mod } 7 = N \Rightarrow N_7 = \{0, 1, 6\}$$

$$N = [7\{\text{inv}(7, 5)\}N_p + 5\{\text{inv}(5,7)\}N_q] \text{ mod } 35$$

$$N = [7*3 N_p + 5*3 N_q] \text{ mod } 35 = [21\{0, 1, 4\} + 15\{0, 1, 6\}] \text{ mod } 35$$

$$N = \{(0, 21, 84) + (0, 15, 90)\} \text{ mod } 35 \quad \text{sumando todos los términos...}$$

$$N = \{0, 15, 90, 21, 36, 111, 84, 99, 175\} \text{ mod } 35 \quad \text{ordenando...}$$

$$N = \{0, 1, 6, 14, 15, 20, 21, 29, 34\}$$

## Ejemplo de números no cifrables (1)

Sea  $p = 13$ ;  $q = 17$ ;  $n = p \cdot q = 221$

Elegimos  $e = 7$  por lo que  $d = \text{inv}(7, 192) = 55$ , luego:

$$\sigma_n = [1 + \text{mcd}(e-1, p-1)][1 + \text{mcd}(e-1, q-1)]$$

$$\sigma_{221} = [1 + \text{mcd}(6, 12)][1 + \text{mcd}(6, 16)] = (1+6)(1+2) = 21$$

Soluciones de  $N^7 \bmod 13 = N \Rightarrow N_p = \{0, 1, 3, 4, 9, 10, 12\}$

Soluciones de  $N^7 \bmod 17 = N \Rightarrow N_q = \{0, 1, 16\}$

Los números no cifrables serán:

$$N = [q\{\text{inv}(q, p)\}N_p + p\{\text{inv}(p, q)\}N_q] \bmod n$$

$$N = [17\{\text{inv}(17, 13)\}N_p + 13\{\text{inv}(13, 17)\}N_q] \bmod 221$$

$$N = [\{17 \cdot 10\}N_p + \{13 \cdot 4\}N_q] \bmod 221$$

$$N = [170 \cdot N_p + 52 \cdot N_q] \bmod 221$$

## Ejemplo de números no cifrables (2)

Teníamos  $N_p = \{0, 1, 3, 4, 9, 10, 12\}$

$N_q = \{0, 1, 16\}$

$N = [170*N_p + 52*N_q] \bmod 221$  luego:

$N = [170*\{0, 1, 3, 4, 9, 10, 12\} + 52*\{0, 1, 16\}] \bmod 221$

$N = [\{0, 170, 510, 680, 1.530, 1.700, 2.040\} + \{0, 52, 832\}] \bmod 221$

$N = [0+0, 0+52, 0+832, 170+0, 170+52, 170+832, \dots] \bmod 221$

$N = [0, 52, 832, 170, 222, 1.002, 510, 562, 1.342, 680, 732, 1.512, 1.530, 1.582, 2.362, 1.700, 1.752, 2.531, 2.040, 2.092, 2.872] \bmod 221$

$N = [0, 52, 169, 170, 1, 118, 68, 120, 16, 17, 69, 186, 204, 35, 152, 153, 205, 101, 51, 103, 220]$  ordenando...

$N = [0, 1, 16, 17, 35, 51, 52, 68, 69, 101, 103, 118, 120, 152, 153, 169, 170, 186, 204, 205, 220]$  estos son los 21 mensajes de  $\sigma_{221}$ .

## Distribución de números no cifrables

Dado que  $N = \{0, 1, 16, 17, 35, 51, 52, 68, 69, 101, 103, 118, 120, 152, 153, 169, 170, 186, 204, 205, 220\}$ , observe que excepto el valor 0, los valores de los extremos siempre sumarán el valor del módulo:  $1+220 = 16+205 = 17+204 = 35+186 \dots = 221 = n$ .

No obstante, esto no es una debilidad porque el siguiente valor no cifrable posterior al 1 es aleatorio y también la distribución entre los demás. Es más, en la mayoría de las claves no se aprecia una secuencia de valores muy clara, aunque sí se observa un comportamiento y distribución bastante curiosos.

Si no fuera así, el sistema sería muy débil porque podríamos conocer de antemano qué valores muy pequeños serían no cifrables (además del 0 y el 1) y con esa información poder deducir si un valor  $x$  de centenas de bits (clave) es o no cifrable.

# Dos casos de números no cifrables

Sean  $p = 409, q = 499$   
 Con  $e = 31, d = 19.663$

Total números no cifrables: 49

0, 1, 1.636, 1.637, 23.313, 23.314,  
 24.949, 24.950, 26.586, 48.263,  
 49.899, 56.388, 58.024, 72.855,  
 74.491, 79.701, 81.337, 81.338,  
 82.973, 82.974, 96.168, 97.804,  
 97.805, 99.440, 99.441, 104.650,  
 104.651, 106.286, 106.287, 107.923,  
 121.117, 121.118, 122.753, 122.754,  
 124.390, 129.600, 131.236, 146.067,  
 147.703, 154.192, 155.828, 177.505,  
 179.141, 179.142, 180.777, 180.778,  
 202.454, 202.455, 204.090.

Sean  $p = 241, q = 251$   
 Con  $e = 61, d = 26.281$

Total números no cifrables: 671

0, 1, 231, 250, 251, 364, 400, 482,  
 522, 604, 640, 733, 866, 1.004,  
 1.024, 1.287, 1.486, 1.506, 1.777,  
 1.870, 1.988, 2.009, 2.028, 2.227,  
 2.259, 2.260, 2.291, 2.510, ...  
 ... 57.981, 58.200, 58.231, 58.232,  
 58.264, 58.463, 58.482, 58.503,  
 58.621, 58.714, 58.985, 59.005,  
 59.204, 59.467, 59.487, 59.625,  
 59.758, 59.851, 59.887, 59.969,  
 60.009, 60.091, 60.127, 60.240,  
 60.241, 60.260, 60.490.

# Cantidad mínima de números no cifrables

Para que la cantidad de números no cifrables sea la mínima posible, es decir 9, deberemos elegir la clave pública  $e$  de forma que:

$$\text{mcd}(e-1, p-1) = 2 \quad \text{y} \quad \text{mcd}(e-1, q-1) = 2$$

Entonces: 
$$\sigma_n = [1 + 2][1 + 2] = 9$$

Esto puede lograrse usando por ejemplo primos seguros:

$$p = 2r + 1 \quad \text{y} \quad q = 2r' + 1 \quad \text{con } r, r', p \text{ y } q \text{ primos grandes}$$

ya que:  $\text{mcd}(e-1, p-1) = \text{mcd}(e-1, (2r+1)-1) \Rightarrow \text{mcd} = 2$  o bien  $2r$

$$\text{mcd}(e-1, q-1) = \text{mcd}(e-1, (2r'+1)-1) \Rightarrow \text{mcd} = 2 \text{ o bien } 2r'$$

Luego: 
$$\sigma_n = \{9, 3(2r+1), 3(2r'+1), (2r+1)(2r'+1)\}$$

Hay que comprobar en diseño que no se den valores del mcd igual a  $2r$  o  $2r'$  pues tendríamos un número alto de mensajes no cifrables.

Además, observe que si  $e = p \Rightarrow \sigma_n = 3p$  y si  $e = q \Rightarrow \sigma_n = 3q$ .

# Cantidad máxima de números no cifrables

En el peor de los casos,  $\text{mcd}(e-1, p-1) = p-1$  y  $\text{mcd}(e-1, q-1) = q-1$

Entonces:  $\sigma_n = [1 + \text{mcd}(e-1, p-1)][1 + \text{mcd}(e-1, q-1)]$

$\sigma_n = p*q = n$  ... ¡todas las cifras irán en claro!

Si en el ejemplo anterior con  $p = 13$ ,  $q = 17$ , hubiésemos elegido como clave  $e = 49$ , con  $d = \text{inv}(49, 192) = 145$ , observamos que:

$$\text{mcd}(e-1, p-1) = \text{mcd}(48, 12) = 12$$

$$\text{mcd}(e-1, q-1) = \text{mcd}(48, 16) = 16$$

$$\sigma_n = [1 + 12][1 + 16] = 13*17 = 221 = p*q = n$$

Por lo tanto, cualquier número en el cuerpo  $n = 221$  será no cifrable para la clave pública  $e = 49$ . Compruebe que en este caso esto se cumple si  $e = \phi(n)/k + 1$  ( $k = 2$  y  $4$ ), es decir  $e = 97$  y  $49$ .

Nunca podrá usarse  $e = \phi(n)/2 + 1$  ya que la clave de descifrado será igual a 1 y por lo tanto no será cifrable ningún número de  $n$ .



# NNC por mala elección de la clave $e$

Sea  $p = 101$ ,  $q = 761$ ,  $n = 76.861$ . Luego  $\phi(n) = 100 \cdot 760 = 76.000$

Algunos valores de  $e$  válidos como clave pero relacionados con  $\phi(n)$ :

$e = \phi(n)/2 + 1 = 38.001$	$\Rightarrow$	76.861 NNC	(100 %)
$e = \phi(n)/4 + 1 = 19.001$	$\Rightarrow$	76.861 NNC	(100 %)
$e = \phi(n)/5 + 1 = 15.201$	$\Rightarrow$	76.861 NNC	(100 %)
$e = \phi(n)/8 + 1 = 9.501$	$\Rightarrow$	38.481 NNC	(50 % aprox.)
$e = \phi(n)/10 + 1 = 7.601$	$\Rightarrow$	76.861 NNC	(100 %)
$e = \phi(n)/16 + 1 = 4.751$	$\Rightarrow$	9.741 NNC	(12,5 % aprox.)
$e = \phi(n)/19 + 1 = 4.001$	$\Rightarrow$	4.141 NNC	(5 % aprox.)
$e = \phi(n)/20 + 1 = 3.801$	$\Rightarrow$	76.861 NNC	(100 %)
$e = \phi(n)/50 + 1 = 1.521$	$\Rightarrow$	15.981 NNC	(20 % aprox.)
$e = \phi(n)/100 + 1 = 761$	$\Rightarrow$	15.981 NNC	(20 % aprox.)
$e = \phi(n)/1.000 + 1 = 77$	$\Rightarrow$	385 NNC	(0,5 % aprox.)

# Confidencialidad en intercambio de clave

- A diferencia del número de claves privadas parejas, por lo general un número relativamente bajo y distribución generalmente en torno a  $2^n$  bits, la cantidad de números no cifrables es mucho mayor y en ciertos casos puede llegar a ser todo el cuerpo de cifra.
- No obstante en este nuevo escenario debemos ser menos paranoicos: la utilización actual de este tipo de cifra con clave pública de destino está en el intercambio de una clave de sesión de corta duración, por lo que la confidencialidad de dicha clave no está en compromiso en tanto es computacionalmente imposible un ataque por fuerza bruta a ella durante el corto tiempo de validez de la misma.
- El único problema es que sería fácilmente detectable pues si la cifra de  $K^e \bmod n$  se envía en claro, el resultado será un número  $K$  de 128 bits en un cuerpo de cifra de 1.024 bits... habrá centenas de ceros 😊.

# Firmas digitales no cifrables

¿Hay algún problema con la firma digital no cifrable?

- ✓ Si la cantidad de números no cifrables con la clave pública (tema confidencialidad) es alto, también lo será en igual proporción el de números no cifrables con la clave privada (tema autenticidad).
- ✓ En este caso, significa que el hash de la firma del mensaje dentro del cuerpo de cifra  $n$  iría en claro. Aunque el hash sea de 128 bits (MD5) ó 160 bits (SHA1) y se cifre con la clave privada del emisor y luego se reduzca al cuerpo de cifra de 1.024 bits, la cifra irá en claro por lo que se podría apreciar claramente al tener esa cifra tener sólo una centena de bits significativos... y muchos ceros a la izquierda.
- ✓ Como mucho esto puede dar pistas al criptoanalista en cuanto a que la clave del emisor podría no ser óptima y, por lo tanto, animarle a intentar otros tipos de ataques por la cifra de otros números en claro.

# Ataque al secreto de N por cifrado cíclico

**Un nuevo problema:** se puede encontrar el número en claro N sin necesidad de conocer d, la clave privada del receptor.

Como  $C = N^e \pmod n$ , realizaremos cifrados sucesivos de los criptogramas  $C_i$  resultantes con la misma clave pública hasta obtener nuevamente el cifrado C original.

$$C_i = C_{i-1}^e \pmod n \quad (i = 1, 2, \dots) \quad \text{con } C_0 = C$$

Si en el cifrado iésimo se encuentra el criptograma C inicial, entonces es obvio que el cifrado anterior (i-1) será el número buscado. Esto se debe a que RSA es un grupo mutiplicativo. Para evitarlo hay que usar primos seguros de forma que los subgrupos de trabajo sean lo suficientemente altos.

# Ejemplo de ataque por cifrado cíclico

Sea  $p = 13$ ,  $q = 19$ ,  $n = 247$ ,  $\phi(n) = 216$ ,  $e = 29$  ( $d = 149$ , no conocido)

El número a cifrar será  $M = 123 \Rightarrow C = 123^{29} \bmod 247 = 119$

$i$	$C_i$
$i = 0$	$C_0 = 119$
$i = 1$	$C_1 = 119^{29} \bmod 247 = 6$
$i = 2$	$C_2 = 6^{29} \bmod 247 = 93$
$i = 3$	$C_3 = 93^{29} \bmod 247 = 175$
$i = 4$	$C_4 = 175^{29} \bmod 247 = 54$
$i = 5$	$C_5 = 54^{29} \bmod 247 = 123$
$i = 6$	$C_6 = 123^{29} \bmod 247 = 119$

en este paso aún no lo sabemos

El ataque ha prosperado muy rápidamente: como hemos obtenido otra vez el criptograma  $C = 119$ , es obvio que el paso anterior con  $C = 123$  se correspondía con el texto en claro. ¿Y si usamos primos seguros?

# Ataque por cifrado cíclico y primos seguros

Sea  $p = 11$  y  $q = 23$ , aunque esto no sea recomendable. Luego  $n = 253$ ,  $\phi(n) = 220$ , y si  $e = 17$ , la clave privada es  $d = 134$ , **no conocida**.

Sea el número confidencial  $N = 123 \Rightarrow C = 123^{17} \bmod 253 = 128$ .

$i$	$C_i$	$i$	$C_i$
$i = 0$	$C_0 = 128$	$i = 12$	$C_{12} = 167^{17} \bmod 253 = 150$
$i = 1$	$C_1 = 128^{17} \bmod 253 = 6$	$i = 13$	$C_{13} = 150^{17} \bmod 253 = 193$
$i = 2$	$C_2 = 6^{17} \bmod 253 = 173$	$i = 14$	$C_{14} = 193^{17} \bmod 253 = 118$
$i = 3$	$C_3 = 173^{17} \bmod 253 = 101$	$i = 15$	$C_{15} = 118^{17} \bmod 253 = 200$
$i = 4$	$C_4 = 101^{17} \bmod 253 = 95$	$i = 16$	$C_{16} = 200^{17} \bmod 253 = 73$
$i = 5$	$C_5 = 95^{17} \bmod 253 = 39$	$i = 17$	$C_{17} = 73^{17} \bmod 253 = 94$
$i = 6$	$C_6 = 39^{17} \bmod 253 = 96$	$i = 18$	$C_{18} = 94^{17} \bmod 253 = 41$
$i = 7$	$C_7 = 96^{17} \bmod 253 = 2$	$i = 19$	$C_{19} = 41^{17} \bmod 253 = 123 \checkmark$
$i = 8$	$C_8 = 2^{17} \bmod 253 = 18$	$i = 20$	$C_{20} = 123^{17} \bmod 253 = 128$
$i = 9$	$C_9 = 18^{17} \bmod 253 = 215$		
$i = 10$	$C_{10} = 215^{17} \bmod 253 = 151$		
$i = 11$	$C_{11} = 151^{17} \bmod 253 = 167$		

Para  $n = 253$ , hemos tenido que recorrer un espacio mucho mayor dentro de un cuerpo de cifra muy similar al anterior ( $n = 247$ ).

# La paradoja del cumpleaños

☞ El próximo ataque a la clave privada estará basado en este problema.

**Pregunta:** ¿Cuál será la confianza (probabilidad  $> 50\%$ ) de que en un aula con 365 personas -no se tiene en cuenta el día 29/02 de los años bisiestos- dos de ellas al azar estén de cumpleaños en la misma fecha?

**Solución:** Se escribe en la pizarra los 365 días del año y las personas entran al aula de uno en uno, borrando el día de su cumpleaños de la pizarra. Para alcanzar esa confianza del 50%, basta que entren sólo 23 personas al aula. Este es un valor muy bajo, en principio inimaginable y de allí el nombre de paradoja, aunque matemáticamente no lo sea.

**Explicación:** El primero en entrar tendrá una probabilidad de que su número no esté borrado igual a  $n/n = 1$ , el segundo de  $(n-1)/n$ , etc. De esta manera, la probabilidad de no coincidencia será  $p_{NC} = n!/(n-k)!n^k$ . Para  $k = 23$  se tiene  $p_{NC} = 0,493$  y así la probabilidad de coincidencia será igual a  $p_C = (1 - p_{NC}) = 0,507$ , que es mayor que 0,5.

# Ataque a la clave por paradoja cumpleaños

Algoritmo propuesto por Merkle y Hellman en 1981:

- El atacante elige dos números aleatorios distintos  $i, j$  dentro del cuerpo de cifra  $n$ . Lo interesante es que elige, además, un mensaje o número  $N$  cualquiera.
- Para  $i = i+1$  y para  $j = j+1$  calcula  $N^i \bmod n$  y  $N^j \bmod n$ .
- Cuando encuentra una coincidencia de igual resultado de cifra para una pareja  $(i, j)$ , será capaz de encontrar  $d$ .

Un ejemplo para resolver en siguientes diapositivas: sea  $p = 7$ ;  $q = 13$ ,  $n = 91$ ,  $e = 11$ ,  $d = 59$ . El atacante sólo conoce  $n = 91$  y  $e = 11$ . Partirá con el número  $N = 20$  y elegirá los valores  $i = 10$  y  $j = 50$ .



# Ejemplo de ataque paradoja cumpleaños

$i$	$C_i$	$j$	$C_j$
$i = 10$	$C_{10} = 20^{10} \bmod 91 = 43$	$j = 50$	$C_{50} = 20^{50} \bmod 91 = 36$
$i = 11$	$C_{11} = 20^{11} \bmod 91 = 41$	$j = 51$	$C_{51} = 20^{51} \bmod 91 = 83$
$i = 12$	$C_{12} = 20^{12} \bmod 91 = 1$	$j = 52$	$C_{52} = 20^{52} \bmod 91 = 22$
$i = 13$	$C_{13} = 20^{13} \bmod 91 = 20$	$j = 53$	$C_{53} = 20^{53} \bmod 91 = 76$
$i = 14$	$C_{14} = 20^{14} \bmod 91 = 36$	$j = 54$	$C_{54} = 20^{54} \bmod 91 = 64$
$i = 15$	$C_{15} = 20^{15} \bmod 91 = 83$	$j = 55$	$C_{55} = 20^{55} \bmod 91 = 6$
$i = 16$	$C_{16} = 20^{16} \bmod 91 = 22$	$j = 56$	$C_{56} = 20^{56} \bmod 91 = 29$
$i = 17$	$C_{17} = 20^{17} \bmod 91 = 76$	$j = 57$	$C_{57} = 20^{57} \bmod 91 = 34$

Hay una colisión en el paso quinto al coincidir el valor  $C = 36$  en contador  $i$  que ya había aparecido en contador  $j$ . Observe los valores repetidos.

Con los valores de  $i$ ,  $j$  y el desplazamiento observado en uno de ellos cuando se detecta la colisión ( $i = 14$ ), se establece un conjunto de ecuaciones y, si el ataque prospera, obtenemos la clave privada, una clave privada pareja, o bien un valor de clave privada particular que sólo sirve para descifrar el número elegido (aquí el 20) y no un número genérico. En este caso se hablará de un falso positivo.

# Resultado del ataque paradoja cumpleaños

La primera coincidencia se encuentra para  $i = 14$ ;  $j = 50$ . Así, el atacante conociendo la clave pública  $e = 11$ , calcula:

$$w = (14-50) / \text{mcd}(11, |14-50|) = -36 / \text{mcd}(11, 36) = -36.$$

Entonces deberán existir valores  $s, t$  de forma que se cumpla lo siguiente:

$$w*s + e*t = 1 \quad \Rightarrow \quad -36*s + 11*t = 1$$

Las posibles soluciones a la ecuación son:  $w*s \bmod e = 1$ ;  $e*t \bmod w = 1$

$$-36*s = 1 \bmod 11 \quad \Rightarrow \quad s = \text{inv}(-36, 11) = \text{inv}(8, 11) = 7$$

$$11*t = 1 \bmod 36 \quad \Rightarrow \quad t = \text{inv}(11, 36) = 23 \quad \longleftarrow$$

El valor  $t = 23$  será una clave privada pareja de  $d = 59$ . Compruebe que se verifica  $w*s + e*t = 1$  y que las claves parejas son 11, 23, 35, 47, 71 y 83.

Nota: como este algoritmo parte con valores aleatorios de los contadores  $i, j$  ( $i$  deberá ser el menor posible y  $j$  la mitad del cuerpo de cifra) y del número  $N$  (en el ejemplo 20) no siempre prospera el ataque, es decir será no determinista. En ese caso, es posible que cambiando el valor de  $N$  sí se logre el objetivo buscado.

# Ataque que entrega alguna clave pareja

- Normalmente el ataque rompe la clave privada o una clave privada pareja; sin embargo, se darán situaciones especiales.
- Sea  $p = 11$ ,  $q = 31$ ,  $e = 13$ . Entonces  $d = 277$  y las claves privadas parejas son: 7, 37, 67, 97, 127, 157, 187, 217, 247, 307, 337.
- Observe la diferencia constante igual a  $\phi(n)/10 = 30$ .
- Se realiza el ataque con el software genRSA tomando valores de  $N$  desde 2 hasta 50, partiendo el contador  $i$  en 3 y  $j$  en  $n/2$ .
- Para  $N = 2$  encuentra  $d' = 157$ ; para  $N = 3$  encuentra  $d' = 97$ ; para  $N = 4$  encuentra  $d' = 127$ ; para  $N = 5$  encuentra  $d' = 127$ ; para  $N = 6$  encuentra  $d' = 97$ ; ... etc.
- Para  $N = 32$  encuentra  $d' = 13$ , una solución falsa.
- Para  $d' = 13$  el programa realiza 2 iteraciones, para  $d' = 127$  realiza 3, para  $d' = 157$  realiza 4 y para  $d' = 97$  realiza 14.

# Ataque que entrega la clave privada

- En función de los parámetros de la clave, a veces se encuentra para muchos valores de  $N$  casi siempre la clave privada  $d$ .
- Sea  $p = 191$ ,  $q = 211$ ,  $e = 31$ . Entonces  $d = 12.871$  y hay 9 claves privadas parejas.
- Se realiza el ataque con el software genRSA tomando valores de  $N$  desde 2 hasta 50, partiendo el contador  $i$  en 3 y  $j$  en  $n/2$ .
- Para casi todos los valores de  $N$  encuentra la clave privada  $d$ .
- Para  $N = 7, 39, 49$  encuentra  $d' = 1.951$ , para  $N = 14$  encuentra  $d' = 13.668$  y para  $N = 23$  encuentra  $d' = 18.191$ , todas falsas.
- Sea ahora  $p = 241$ ,  $q = 251$ ,  $e = 11$ . Entonces  $d = 49.091$  y hay 9 claves privadas parejas.
- Aunque aquí casi siempre encuentra el valor  $d' = 19.091$  como clave privada pareja válida, si usamos  $N = 36$  el programa nos da como solución el valor  $d' = 0$ , obviamente un falso positivo.

# ¿Podría darse un ataque distribuido?

- ➡ El ataque basado en la paradoja del cumpleaños no sería factible realizarlo en un solo PC por la alta complejidad computacional.
- ➡ ... pero bien podría pensarse en un algoritmo distribuido, de forma que un computador hiciera las veces de servidor y todos los demás (... tal vez varios cientos de miles) actuaran como clientes.
- ➡ El servidor tendría como función distribuir trozos de cifra entre los clientes en diferentes intervalos de valores  $i, j$  como los del ejemplo anterior y, además, recibir los resultados de los clientes para detectar colisiones. Esta última función será la más crítica.
- ➡ Supuestamente este ataque llevaría un tiempo menor que el de factorizar el valor de  $n$ , para así encontrar la clave privada.
- ➡ Si bien no está demostrado la factibilidad real en tiempo de cómputo de esta opción, el hecho de que un certificado digital, y por ende la clave privada, tenga una validez de un año podría ser un motivo de preocupación ... siempre sin caer en paranoias 😊.

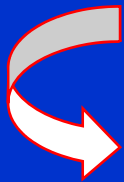
# La otra historia del algoritmo RSA

- ➡ Rivest, Shamir y Adleman son los autores de RSA pero un algoritmo de cifra asimétrico basado en la dificultad de factorizar números grandes como función unidireccional fue descubierto mucho antes...
- ➡ En el año 1969 el Government Communications Headquarters (GCHQ) en Gran Bretaña comienza a trabajar en la idea de poder distribuir claves a través de una cifra no simétrica. En 1973, el matemático Clifford Cocks llegará a la misma conclusión que los creadores de RSA.
- ➡ Desgraciadamente este trabajo fue considerado como alto secreto por el gobierno británico por lo que su contenido no se hace público ni se patenta como invento, algo que sí hacen Diffie y Hellman en 1976 con su intercambio de claves y en 1978 otro tanto los creadores del algoritmo RSA.

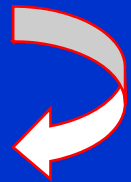
# Consideraciones sobre el bloque de cifra

Si queremos cifrar mensajes en vez de números y ese mensaje fuese mayor que el módulo de trabajo del sistema  $n = p \cdot q$  para RSA ...

¿cómo se generarían los bloques del mensaje a cifrar?

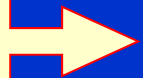


El mensaje  $M$  puede transformarse en números y éstos se dividen en bloques de  $g-1$  dígitos, siendo  $g$  el número de dígitos del módulo de trabajo:  
el valor  $n = n \cdot p$  para RSA.



Ya se ha dicho que la práctica esto no ocurrirá puesto que el cuerpo de cifra es como mínimo de 1.024 bits y el “mensaje” a cifrar tendrá sólo una centena de bits.

Ejemplo



# Ejemplo de elección del bloque con RSA

Se representará el mensaje en su valor ANSI decimal.

$$n = p * q = 89 * 127 = 11.303 \Rightarrow \text{bloques de cuatro dígitos}$$

$$\phi(n) = 11.088; e = 25; d = \text{inv}(25, 11.088) = 10.201$$

$$M = \text{Olé} = 079\ 108\ 233 \Rightarrow M = 0791\ 0823\ 3$$

*Se recupera el mensaje agrupando en bloques de 4 dígitos excepto el último*

## CIFRADO

$$C_1 = 791^{25} \text{ mod } 11.303 = 7.853$$

$$C_2 = 823^{25} \text{ mod } 11.303 = 2.460$$

$$C_3 = 3^{25} \text{ mod } 11.303 = 6.970$$

## DESCIFRADO

$$M_1 = 7.853^{10201} \text{ mod } 11.303 = 0791$$

$$M_2 = 2.460^{10201} \text{ mod } 11.303 = 0823$$

$$M_3 = 6.970^{10201} \text{ mod } 11.303 = 3$$



Use el portapapeles

## Prácticas del tema

Software OpenSSL:

<http://www.slproweb.com/products/Win32OpenSSL.html>

1. Una vez instalado Win32OpenSSL para Windows en el disco duro, en la carpeta C:\OpenSSL-Win32\bin> y desde el símbolo del sistema MSDOS genere una clave RSA de 1024 bits mediante el comando: `openssl genrsa 1024`. Observe que creada la clave, nos indica “e is 65537 (0x10001)”.
2. Cree una nueva clave RSA guardando la clave en un archivo de nombre `rsakey` con el comando: `openssl genrsa -out rsakey 1024`.
3. Recupere ahora esa clave y guárdela en un archivo en formato hexadecimal de nombre `claveRSA1`: `openssl rsa -in rsakey -text -out claveRSA1`. Si no desea incluir en el archivo la clave privada, agregue la opción `-noout`.
4. Edite `claveRSA1` con Word o WordPad, elimine todos los “:” y elimine luego los “cuatro espacios en blanco” de forma que los valores de p y de q en hexadecimal estén cada uno en una sola cadena. Pegue esos valores de p y de q en el programa `genRSA`, con e = 10001, genere y observe la clave.
5. Repita los puntos 2, 3 y 4 hasta encontrar una clave de “baja calidad”.