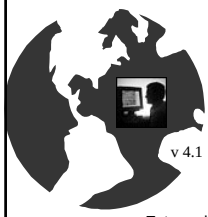
Capítulo 14
Cifrado Asimétrico Exponencial

Seguridad Informática y Criptografía



Material Docente de
Libre Distribución

Ultima actualización del archivo: 01/03/06
Este archivo tiene: 89 diapositivas

Dr. Jorge Ramió Aguirre
Universidad Politécnica de Madrid

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.

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

- 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:

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

ssi $N^{ed} = N \bmod p$

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

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

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

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

$e*d \bmod \phi(n) = 3*27 \bmod 40 = 81 \bmod 40 = 1$

$C = N^e \bmod n = 50^3 \bmod 55 = (2*5^2)^3 \bmod 55$

$C = [(2)^3 \bmod 55 * (5^2)^3 \bmod 55] \bmod 55$ - *por reducibilidad* ↓ -

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

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

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

Por el Teorema de Euler y del Resto Chino $= 2 * 5 * 5 \bmod 55 = 50$

Intercambio de clave de Diffie y Hellman

- El comienzo de los sistemas de clave pública se debe al estudio hecho por Whitfield Diffie y Martin Hellman (1976).

  Protocolo de Intercambio de Claves de Diffie y Hellman

  A y B seleccionan un grupo multiplicativo (con inverso) p y un generador α de dicho primo, ambos valores públicos.

  - A genera un número aleatorio a y envía a B α^a mod p
  - B genera un número aleatorio b y envía a A α^b mod p
  - B calcula $(\alpha^a)^b$ mod p = α^{ab} mod p y luego destruye b
  - A calcula $(\alpha^b)^a$ mod p = α^{ba} mod p y luego destruye a
  - El secreto compartido por A y B es el valor α^{ab} mod p

http://www.cs.purdue.edu/homes/ninghui/courses/Fall04/lectures/diffie-hellman.pdf

Ejemplo de intercambio de clave de DH

Adela (A) y Benito (B) van a intercambiar una clave de sesión dentro del cuerpo primo p = 1.999, con α = 33. El usuario A elegirá a = 47 y el usuario B elegirá b = 117.

Algoritmo:

- A calcula α^a mod p = 33^{47} mod 1.999 = 1.343 y se lo envía a B.
- B calcula α^b mod p = 33^{117} mod 1.999 = 1.991 y se lo envía a A.
- B recibe 1.343 y calcula 1.343^{117} mod 1.999 = 1.506.
- A recibe 1.991 y calcula 1.991^{47} mod 1.999 = 1.506.

La clave secreta compartida por (A) y (B) será K = 1.506

¿Puede un intruso atacar la clave DH?

Un intruso que conozca las claves públicas p y α e intercepte el valor α^a mod p que ha enviado A y el valor α^b mod p que ha enviado B no podrá descubrir los valores de a, de b y ni menos α^{ab} mod p ...

Salvo que se enfrente al Problema del Logaritmo Discreto (PLD) que, como ya hemos visto, se vuelve computacionalmente intratable para valores del primo p grandes.

http://en.wikipedia.org/wiki/Discrete_logarithm

Seguridad del intercambio de clave de DH

- La seguridad del intercambio de clave de Diffie y Hellman radica en la imposibilidad computacional a la que se enfrentará el criptoanalista al tener que resolver el problema del logaritmo discreto para encontrar la clave privada que se encuentra en el exponente de la expresión α^i mod p = C.
- Como p y α serán públicos, al capturar el valor C el atacante deberá resolver $i = \log_\alpha C$ mod p, un problema no polinomial (debido a la operación final dentro del módulo p) que para valores grandes de p (del orden o superior a los 1.000 bits) resulta computacionalmente imposible encontrar su solución.
- El algoritmo propuesto inicialmente es vulnerable ante un ataque del tipo “man in the middle” como veremos a continuación. No obstante, esta vulnerabilidad puede evitarse.

¿Es vulnerable el protocolo de DH?

- A elige un número a con $1 < a < p-1$ y envía a B $\alpha^a \bmod p$
- C intercepta este valor, elige un número c con $1 < c < p-1$ y envía a B $\alpha^c \bmod p$
- B elige un número b con $1 < b < p-1$ y envía a A $\alpha^b \bmod p$
- C intercepta este valor y envía a A $\alpha^c \bmod p$ (valor anterior)
- A y B calculan sus claves $k_A = (\alpha^c)^a \bmod p$, $k_B = (\alpha^c)^b \bmod p$
- C calcula también las claves:
  - $k_{CA} = (\alpha^a)^c \bmod p$
  - $k_{CB} = (\alpha^b)^c \bmod p$

Una solución a este problema es el sellado de tiempo.

Por lo tanto, a partir de ahora C tiene "luz verde" y puede interceptar todos los mensajes que se intercambian A y B.

Intercambio de clave DH entre n usuarios

El protocolo DH se puede generalizar para n usuarios: sea n = 3.

A, B y C seleccionan un grupo p y un generador α

- A genera un número aleatorio a y envía $\alpha^a \bmod p$ a B
- B genera un número aleatorio b y envía $\alpha^b \bmod p$ a C
- C genera un número aleatorio c y envía $\alpha^c \bmod p$ a A
- A recibe $\alpha^c \bmod p$ y calcula $(\alpha^c)^a \bmod p$ y se lo envía a B
- B recibe $\alpha^a \bmod p$ y calcula $(\alpha^a)^b \bmod p$ y se lo envía a C
- C recibe $\alpha^b \bmod p$ y calcula $(\alpha^b)^c \bmod p$ y se lo envía a A
- A recibe $\alpha^{bc} \bmod p$ y calcula $(\alpha^{bc})^a \bmod p = \alpha^{bca} \bmod p$
- B recibe $\alpha^{ca} \bmod p$ y calcula $(\alpha^{ca})^b \bmod p = \alpha^{cab} \bmod p$
- C recibe $\alpha^{ab} \bmod p$ y calcula $(\alpha^{ab})^c \bmod p = \alpha^{abc} \bmod p$
- El secreto compartido por A, B y C es el valor $\alpha^{abc} \bmod p$

Condiciones del intercambio de clave D-H

CONDICIONES DEL PROTOCOLO:

- El módulo p debe ser un primo grande, al menos de 1.024 bits.
- Interesa que el Indicador de Euler $\phi(p) = p-1$, además del valor 2, tenga factores primos grandes.
- El generador α debe ser una raíz primitiva del módulo p.

☹ Si el módulo es un primo pequeño, se puede hacer un ataque por fuerza bruta dentro de un tiempo razonable.

☹ Si el generador no es una raíz primitiva del grupo p, entonces la operación $\alpha^i \bmod p$ $(1 \leq i \leq p-1)$ no genera todos los restos del grupo y esto facilita el ataque por fuerza bruta.

Veamos un ejemplo →

Raíz α incorrecta (falsa)

MALA ELECCIÓN DE LOS PARÁMETROS:

Sean el grupo de trabajo p = 13 y un valor $\alpha = 3$ *... entonces*

$3^1 \bmod 13 = 3$ $3^2 \bmod 13 = 9$ $3^3 \bmod 13 = 1$ $1 \Rightarrow$ *vamos mal*

$3^4 \bmod 13 = 3$ $3^5 \bmod 13 = 9$ $3^6 \bmod 13 = 1$

$3^7 \bmod 13 = 3$ $3^8 \bmod 13 = 9$ $3^9 \bmod 13 = 1$

Sólo debería darse unidad en este caso.

$3^{10} \bmod 13 = 3$ $3^{11} \bmod 13 = 9$ $3^{12} \bmod 13 = 1$ $(\alpha^{p-1} \bmod p = 1)$

Se repiten los restos 3, 9 y 1 porque 3 no es un generador de Z_{13}.
Observe que $3^4 \bmod 13 = (3^3)(3)^1 \bmod 13 = 1*(3)^1 \bmod 13 = 3$.

Un ataque por fuerza bruta deberá buscar sólo en una tercera parte del espacio de claves y, lo que es peor, la probabilidad de éxito de encontrar un valor verdadero b en $\alpha^b \bmod p$ aumenta de 1/12 a 1/3.

Raíz α correcta

¿Y si ahora α = 2 ?

Primero intente calcularlo... y luego para comprobar sus resultados, avance.

$2^1 \bmod 13 = 2$ $2^2 \bmod 13 = 4$ $2^3 \bmod 13 = 8$
$2^4 \bmod 13 = 3$ $2^5 \bmod 13 = 6$ $2^6 \bmod 13 = 12$
$2^7 \bmod 13 = 11$ $2^8 \bmod 13 = 9$ $2^9 \bmod 13 = 5$
$2^{10} \bmod 13 = 10$ $2^{11} \bmod 13 = 7$ $2^{12} \bmod 13 = 1$ ✓

Ahora sí están todos los restos multiplicativos del cuerpo Z_{13} porque el resto 2 es un generador dentro de este cuerpo.
Observe que el valor unidad sólo se obtiene para $\alpha^{p-1} \bmod p$.

Como vimos en el capítulo de Teoría de Números, en p = 13 serán generadores los valores g = 2, 6, 7, 11.

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∗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: mcd [e, $\phi(n)$] = 1.
5. Cada usuario calcula la clave privada d = 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: $C = h(M)^{dE} \bmod n_E$

http://www.di-mgt.com.au/rsa_alg.html

Intercambio de clave RSA (B → A)

Benito

En el protocolo intercambiaremos una clave K

Sea K = DA9F (16 bits)

Adela

Claves Benito	$2^{16} < 66.331 < 2^{17}$	Claves Adela
$n_B = 65.669$	Forzaremos cifrar un bloque de 16 bits	$n_A = 66.331$
$e_B = 35$, $d_B = 53.771$		$e_A = 25$, $d_A = 18.377$

Cifra $K = DA9F_{16} = 55.967_{10}$

$C = K^{eA} \bmod n_A$

$C = 55.967^{25} \bmod 66.331 = 16.667$

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	Claves Adela
$n_B = 65.669$	$n_A = 66.331$
$e_B = 35$, $d_B = 53.771$	$e_A = 25$, $d_A = 18.377$

Adela

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

$C = K^{eA} \bmod n_A$ $C = 55.967^{25} \bmod 66.331 = 16.667$

Benito había enviado a Adela C = 16.667

Adela calcula:

- $C^{dA} \bmod n_A = 16.667^{18.377} \bmod 66.331 = 55.967$.
- 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 mcd (5,72) = 1 ∴ d = inv(5,72) = 29
CIFRADO:
$C = N^e \bmod n = 48^5 \bmod 91 = 5245.803.968 \bmod 91 = 55$
DESCIFRADO:
$N = C^d \bmod n = 55^{29} \bmod 91 = 48$ ... 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∗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:

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

con:
$A_p = q\ [inv\ (q,p)] = q^{p-1} \bmod n$
$A_q = p\ [inv\ (p,q)] = p^{q-1} \bmod n$
$d_p = d \bmod (p-1)$ $d_q = d \bmod (q-1)$
$C_p = C \bmod p$ $C_q = C \bmod q$

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^{dp} y C_q^{dq} pero ambos se hacen en cuerpos mucho menores que n.

Ejemplo de descifrado RSA usando el TRC

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

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

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: $N = \{A_p[C_p^{dp} \pmod p] + A_q[C_q^{dq} \pmod q]\} \bmod n$

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

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

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 $d = inv [e, \phi(n)]$, 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.

http://home.netcom.com/~jrhowell/math/factor.htm

Tiempo necesario para afrontar el PFNG

Para un procesador de $2x10^8$ 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. Por lo tanto, ... deberemos ser siempre muy cautos.

http://www.rsasecurity.com/rsalabs/node.asp?id=2964

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

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

http://www.criptored.upm.es/guiateoria/gt_m117f.htm

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 ⇒ n = 181∗251 = 45.431
- Como √45.431 = 213,14 buscaremos valores enteros de x mayores que 213 de forma que (x^2 - 45.431) sea un cuadrado perfecto ↓

1. $x = 214 \Rightarrow x^2 - 45.431 = 365$ ∴ √365 = 19,10 ☹
2. $x = 215 \Rightarrow x^2 - 45.431 = 794$ ∴ √794 = 28,17 ☹
3. $x = 216 \Rightarrow x^2 - 45.431 = 1.225$ ∴ √1.225 = 35 ☺

   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

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

Esto se cumple con los denominados primos seguros ⟶

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 ⇒ 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:

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

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

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

La clave pública e tendrá λ claves parejas d_i de la forma:

$d_i = d_\gamma + i\,\gamma \qquad 1 < d_i < n$

$i = 0, 1, \ldots \lambda \qquad \lambda = \lfloor (n - d_\gamma)/\gamma \rfloor$

En el ejemplo anterior tenemos que:

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

Luego: $d_\gamma = inv (41, 36) = 29$, 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 extraños 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 λ 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 γ = mcm (10, 46) = 230 y d_γ = 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 = inv [e, $\phi(n)$] = 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 λ sea igual a la unidad, habrá que elegir además un valor adecuado de clave pública:

Tomando el mismo ejemplo anterior:

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

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

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

http://www.openssl.org

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 $e_\gamma = inv(d, \gamma) = inv(137, 36) = 5$, 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:

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

Los números no cifrables serán:

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

con: N_p las soluciones de $N^e \bmod p = N$

N_q las soluciones de $N^e \bmod q = N$

Esto último debido al TRC puesto que $N^e \bmod n = N$

En el ejemplo anterior se da el caso mínimo: ↓ ↓ Valores para un mínimo

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

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

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

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

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

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

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

Ejemplo de números no cifrables (1)

Sea p = 13; q = 17; n = p*q = 221

Elegimos e = 7 por lo que d = inv (7, 192) = 55, luego:

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

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

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:

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

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

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

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

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:

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

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

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

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

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

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

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

Esto se logra usando primos seguros:

$p = 2r + 1$ y $q = 2r' + 1$ con r, r', p y q primos grandes

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

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

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

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

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, mcd (e-1, p-1) = p-1 y mcd (e-1, q-1) = q-1

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

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

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

mcd (e-1, p-1) = mcd (48, 12) = 12

mcd (e-1, q-1) = 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 $\phi(n) = 100*760 = 76.000$

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

$$C_i = C^e_{i-1} \bmod n \quad (i = 1, 2, ...) \quad \text{con} \quad C_0 = C$$

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	$C_5 = 54^{29} \bmod 247 = 123$	en este paso aún no lo sabemos
i = 6	$C_6 = 123^{29} \bmod 247 = 119$	

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

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 mod n y N^j mod 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.

Puede encontrar varios tipos de ataques a RSA en la siguiente página:

http://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf

Ejemplo de ataque paradoja cumpleaños

i	C_i	j	C_i
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:

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

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: $w*s \bmod e = 1$; $e*t \bmod w = 1$

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

$11*t = 1 \bmod 36 \quad \Rightarrow \quad t = inv\ (11, 36) = 23$ ←

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.

http://livinginternet.com/i/is_crypt_pkc_inv.htm

Cifrado Pohlig y Hellman con clave secreta

- Stephen Pohlig y Martin Hellman proponen en enero de 1978 un algoritmo de cifra de clave secreta y que basa su seguridad en el problema del logaritmo discreto. Hablamos de sólo un mes antes que el algoritmo RSA... algo que también llama la atención.
  - Se elige un grupo multiplicativo Z_p^*, p es un primo grande.
  - Cada usuario elige una clave e, que sea primo relativo con el grupo $\phi(p) = p-1$ y calcula $d = inv [(e, \phi(p)]$.
  - La clave secreta serán los valores e y d.
  - Se cifrará $C = M^e \bmod p$ y se descifrará $M = C^d \bmod p$.

Dado que el sistema carece de firma digital en el sentido amplio al ser de clave secreta, estará sólo orientado a la cifra de mensajes o números para la confidencialidad. No puede competir en velocidad con la cifra simétrica.

http://ieeexplore.ieee.org/xpl/abs_free.jsp?arNumber=1055817

Ejemplo de cifrado Pohlig y Hellman

Adela cifrará un mensaje M que desea enviar a Bernardo:

$p = 263 \Rightarrow \phi(p) = 262$; $e = 15 \Rightarrow d = inv(15, 262) = 35$

Sea M = Adiós = 65 100 105 243 115

Como se usa el código ANSI, podremos cifrar en bloques de un carácter pues el módulo p es algo mayor que 256.

Operación Cifrado:

$C = M^e \bmod p = 65^{15} \bmod 263, 100^{15} \bmod 263,$

$105^{15} \bmod 263, 243^{15} \bmod 263, 115^{15} \bmod 263$

C = 245, 143, 179, 86, 101

Ejemplo de descifrado Pohlig y Hellman

B descifra el criptograma C enviado por A:

$p = 263$; $d = inv(15, 262) = 35$

C = 245, 143, 179, 86, 101

Operación Descifrado:

$M = C^d \bmod p = 245^{35} \bmod 263, 143^{35} \bmod 263,$

$179^{35} \bmod 263, 86^{35} \bmod 263, 101^{35} \bmod 263$

M = 065, 100, 105, 243, 115

Convirtiéndolo al código ANSI: M = Adiós

Algoritmo de cifra asimétrica de ElGamal

Taher ElGamal propone en 1985 un algoritmo de cifra que hace uso del problema del logaritmo discreto PLD.

- Se elige un grupo multiplicativo Z_p^*, donde p es un primo grande
- Del grupo p se elige una raíz α, generador del grupo
- Cada usuario elige un número aleatorio λ dentro de p
  - El valor λ será la clave privada
- Cada usuario calcula α^{λ} mod p
  - Los valores (α^{λ} mod p) y p serán la clave pública
- Seguridad del sistema
  - Para descubrir la clave privada, el atacante deberá enfrentarse al problema del logaritmo discreto para p grande

http://web.usna.navy.mil/~wdj/book/node48.html

Operación de cifra con ElGamal

Operación Cifrado: A cifra un número N que envía a B

- El usuario B ha elegido su clave privada *b* dentro del cuerpo del número primo p que es público.
- El usuario B ha hecho pública su clave α^b mod p.
- El emisor A genera un número aleatorio ν de sesión y calcula α^{ν} mod p.
- Con la clave pública de B (α^b) el emisor A calcula:
  - $(\alpha^b)^{\nu}$ mod p y $N*(\alpha^b)^{\nu}$ mod p
- A envía a B el par: $C = [\alpha^{\nu} \text{ mod } p, N*(\alpha^b)^{\nu} \text{ mod } p]$

Operación de descifrado con ElGamal

Operación Descifrado: B descifra el criptograma C que envía A

- El usuario B recibe C = [α^v mod p, N∗(α^b)^v mod p].
- B toma el valor α^v mod p y calcula (α^v)^b mod p.
- B descifra el criptograma C haciendo la siguiente división: [N∗(α^b)^v mod p] / [(α^v)^b mod p] ... porque (α^b)^v = (α^v)^b
- El paso anterior es posible hacerlo porque existirá el inverso de (α^v)^b en el grupo p al ser p un primo. Luego:

$$[N*(\alpha^b)^v * \{inv\ (\alpha^v)^b, p\}] \bmod p = N$$

Ejemplo de cifrado con ElGamal

Adela (A) enviará a Benito (B) el número N = 10 cifrado dentro del cuerpo p = 13 que usa Benito.

CIFRADO

Claves públicas de Benito: p = 13, α = 6, (α^b) mod p = 2

Adela (A) elige por ejemplo v = 4 y calcula:

(α^v) mod p = 6^4 mod 13 = 9

(α^b)^v mod p = 2^4 mod 13 = 3

N∗(α^b)^v mod p = 10∗3 mod 13 = 4

Y envía a (B): (α^v) mod p, N∗(α^b)^v mod p = [9, 4]

Ejemplo de descifrado con ElGamal

DESCIFRADO

La clave privada de Benito (B) es b = 5

Benito recibe: $[(\alpha^v) \bmod p, N*(\alpha^b)^v \bmod p] = [9, 4]$

Benito calcula:

$(\alpha^v)^b \bmod p = 9^5 \bmod 13 = 3$

$[N*(\alpha^b)^v] * inv[(\alpha^v)^b, p] = 4 * inv(3, 13) = 4 * 9$

$N = 4 * 9 \bmod 13 = 10$ (se recupera el valor)

Recuerde que α debe ser una raíz de p. Como ya hemos visto, si α no es una raíz, aunque sí puede hacerse la cifra, se facilitaría el ataque al problema del logaritmo discreto.

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*q para RSA y p para ElGamal)...

¿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*p para RSA y p para ElGamal.

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.

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

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

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

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

CIFRADO	DESCIFRADO
$C_1 = 791^{25} \bmod 11.303 = 7.853$	$M_1 = 7.853^{10201} \bmod 11.303 = 0791$
$C_2 = 823^{25} \bmod 11.303 = 2.460$	$M_2 = 2.460^{10201} \bmod 11.303 = 0823$
$C_3 = 3^{25} \bmod 11.303 = 6.970$	$M_3 = 6.970^{10201} \bmod 11.303 = 3$

Fortaleza de la cifra exponencial

El problema del Logaritmo Discreto PLD será similar al de la Factorización de Números Grandes PFNG ya que ambos van a suponer un tiempo de ejecución de tipo no polinomial.

Recuerde que el número de pasos para resolver el PFNG era de $e^{\sqrt{\{\ln(n)*\ln[\ln(n)]\}}}$.

Si suponemos un sistema que consuma 1 μseg por paso:

n = 60 dígitos $\Rightarrow$ $2,7*10^{11}$ pasos	3 días
n = 100 dígitos $\Rightarrow$ $2,3*10^{15}$ pasos	74 años
n = 200 dígitos $\Rightarrow$ $1,2*10^{23}$ pasos	$3,8*10^{9}$ años

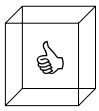
El PLD es matemáticamente similar pues el número de pasos será ahora aproximadamente igual a $e^{\sqrt{\{\ln(p)*\ln[\ln(p)]\}}}$.

http://en.wikipedia.org/wiki/Discrete_logarithm

Resumen de los sistemas de clave pública

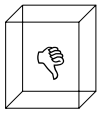
Pros y contras de los Sistemas de Clave Pública

- Emisor y receptor generan un par de claves, pública y privada, relacionadas por una función con trampa.
- Emisor y receptor de un mensaje usan claves diferentes para las operaciones de cifrado, descifrado y firma.
- La seguridad del sistema va asociada a la resolución de un problema matemático de difícil solución en el tiempo.
- Firma digital completa: autentican al mensaje y al emisor.



... pero

- Es necesario contar con mecanismos de certificación para asegurar la veracidad de las claves públicas.
- Son sistemas de cifra muy lentos.



Fin del capítulo

Cuestiones y ejercicios (1 de 4)

1. A partir de la ecuación $e*d = k\phi(n) + 1$, compruebe que las claves RSA e = 133 y d = 38.797 son inversas en el cuerpo n = 40.501.
2. En el ejemplo de intercambio de clave DH del libro con p = 1.999, ¿podrían haber elegido Adela y Benito α = 34, 35, 36 ó 37?
3. Carmela (C) intercepta la clave de sesión DH que se intercambian Adela (A) y Benito (B) dentro del cuerpo p = 127. Si se usa como generador α = 19 y a = 3, b = 12 y c = 7, desarrolle el algoritmo que permite a C interceptar la comunicación y engañar a A y B.
4. Los usuarios A, B, C y D desean intercambiar una clave usando el método DH. Proponga un protocolo genérico que solucione este problema y presente un ejemplo en el cuerpo p = 23 y elija α.
5. Diseñe un sistema RSA en el que p = 53, q = 113. Elija como clave pública el mínimo valor posible de 2 dígitos.

Cuestiones y ejercicios (2 de 4)

6. Para los datos de diseño del ejercicio 5, cifre el número M = 121 y luego descífrelo. Use el algoritmo de exponenciación rápida.
7. Vuelva a descifrar el criptograma usando ahora el Teorema del Resto Chino. Aunque en este caso haya hecho más cálculos ¿por qué es interesante usar aquí el Teorema del Resto Chino?
8. Si p = 353 y q = 1.103, cifre el mensaje ASCII de cuatro caracteres M = HOLA en bloques de tamaño eficiente (mínimo) para e = 17.
9. Para los datos del ejercicio anterior, encuentre la clave privada d usando el algoritmo extendido de Euclides.
10. ¿Por qué se usa una clave pública e de un valor relativamente bajo y, por contrapartida, la clave privada d es alta? ¿Qué utilidad tiene?
11. ¿Cómo atacaría un sistema RSA mal diseñado con primos cercanos y cuyo módulo es n = 205.027? Encuentre los valores de p y q.

Cuestiones y ejercicios (3 de 4)

12. En el sistema RSA con p = 11 y q = 19, se elige como clave pública e = 31. ¿Cuántas y cuáles son las claves privadas parejas?
13. Para los mismos datos del ejercicio anterior, ¿cuántos y cuáles son los mensajes no cifrables? ¿Qué sucede si ahora e = 33?
14. ¿Cómo puede minimizarse el número de claves privadas parejas y el de mensajes no cifrables? Dé un ejemplo con primos de dos dígitos.
15. Atacamos un sistema RSA con un cifrado cíclico. ¿Qué es lo que vulneramos, el secreto de la clave privada o secreto del mensaje?
16. Se ataca por cifrado cíclico el sistema RSA: p = 23, q = 41 y e = 17. ¿Cuántos cifrados hay que hacer para romper N = 200 y N = 185?
17. Cifre M = “La Puerta de Alcalá” (de 19 caracteres ANSI) con un sistema de Pohlig y Hellman. Use el primer primo que le permita cifrar bloques de 2 bytes. Descifre ahora el criptograma C.

Cuestiones y ejercicios (4 de 4)

18. Vamos a cifrar con ElGamal el mensaje en ASCII decimal de la letra A. ¿Cuáles son los valores mínimos del cuerpo de cifra, del generador α, de la clave pública y del valor local v?
19. Descifre el criptograma obtenido en el ejercicio anterior.
20. Se va a cifrar con RSA el mensaje M = Hola. Si p = 53 y q = 97, ¿de cuántos dígitos será el bloque óptimo de cifra?
21. En un sistema real, ¿tiene sentido hablar de bloques con un número de dígitos óptimos? ¿Por qué? Justifique su respuesta.
22. ¿Cuántos pasos debe realizar una factorización de un número n de 120 dígitos? Si tenemos un sistema que consume 1 μseg por paso, ¿cuántos años tardaríamos más o menos en factorizar ese número?
23. El valor encontrado en el ejercicio anterior, ¿es siempre fijo o es sólo una aproximación? Justifique su respuesta.

Use el portapapeles Prácticas del tema 14 (1/10)

Software genRSA: http://www.criptored.upm.es/software/sw_m001d.htm

1. Genere una clave de forma manual con p = 5, q = 13 y e = 11. Observe el número de claves parejas y el de mensajes no cifrables.
2. ¿Cuáles son las claves privadas parejas para p = 13, q = 19, e = 41?
3. Compruebe las claves parejas del ejemplo de los apuntes usando primos seguros: p = 11, q = 47, e = 17. Repita el ejemplo con e = 7, e = 77.
4. Genere al menos una docena de claves de forma automática para un valor n de 24 bits, con p y q diferentes, y observe el número de claves parejas y de mensajes no cifrables. Repita el ejemplo con p y q de igual tamaño y luego repita todo el ejercicio para claves de 30 y 32 bits.
5. Para la clave (p, q, e) que se indica (7187, 107791, 6293) observe cómo se distribuyen las claves privadas parejas en incrementos de tamaño 107790.
6. Cambie las unidades a hexadecimal y repita el ejercicio 4 para claves de 512, 1024 y 2048 bits. En cada caso compruebe la primalidad de p y q.

Use el portapapeles Prácticas del tema 14 (2/10)

7. Observe las claves parejas para la clave de 1.024 bits con valores p; q; e: 02E68E02BE7400FE11E8A45B60017F988251AEED1CF5A9820A6BC9 DE01A408A2725A0977B1A4584556C8F2A6E450089AA860007CE446 BD342D5320AF12E9CE8D41; 3560B5461224C1DD6332CA632A13C9C13D9072AB1297092332E0773 FEACF9D2D449DE87436B3267C185698515B8948A4792F9C1328712C AAD35FD1A94ADAFE9D; 10001. ¿Le llama algo la atención?
8. Repita el ejercicio anterior con e = 41B3, e = 41B7, e = 41BD, e = 41BF.
9. Observe las claves parejas para la clave de 512 bits con valores p; q; e: D2BF44D863DC579E5192EDF83744EBFF5A72E2D5E8DE9FE330EDF D65114FFF0F; D4C1CE77FE15374D8C7E0CE2CEF52E283D237521262E3D4E51D321 6BE1B50665; 10001. Analice la distribución de estas claves parejas.
10. Si aunque alto, el porcentaje de claves débiles en el caso anterior sigue siendo insignificante, ¿estaría Ud. seguro del secreto de su clave privada?

Use el portapapeles Prácticas del tema 14 (3/10)

11. Genere manualmente las claves que se indican (p, q, e) de 16, 17, 18, 19 y 20 bits y genere el log de los mensajes no cifrables: (199, 251, 26803); (211, 463, 21727); (419, 499, 30713); (409, 907, 2995); (769, 929, 8413). Guarde los archivos como 16b_mnc, 17b_mnc, 18b_mnc, 19b_mnc, 20b_mnc y en cada caso anote el tiempo empleado en hacer todos los cifrados. En este último su computador podría tardar cerca de 3 minutos.
12. De acuerdo a los tiempos observados en el ejercicio anterior, encuentre la tasa media de cifrados del programa genRSA en este entorno. Imagine y comente cómo funcionaría un hipotético ataque multiusuario en red.
13. Con la clave (p, q, e) del ejemplo de los apuntes (89, 31, 29) cifre el mensaje numérico N = 1995. Compruebe los pasos del descifrado usando el Teorema del Resto Chino.
14. Cree la clave (p, q, e) de 24 bits (8123, 1523, 25219) y cifre a continuación el número N = 3571. Descífrelo primero de forma normal y luego con el Teorema del Resto Chino.

Use el portapapeles Prácticas del tema 14 (4/10)

15. Para la clave (p, q, e) de 24 bits (8123, 1523, 25219) cifre y descifre el valor decimal N = 65, la letra A. Repita la cifra para el mensaje M = A, indicando que es texto. Observe la representación hexadecimal del ANSI. Compare los valores de ambos criptogramas en una sola base.
16. Con esa clave, cifre M = ABC, observe el criptograma y el descifrado. Repita la cifra para M = ABCD, luego M = abc1234. Compruebe que, dada la longitud de la clave, en este caso se cifra por bloques de 24 bits, 3 bytes.
17. Para la clave de 20 bits con primos de igual tamaño (p, q, e) = (853, 983, 3671) ataque el módulo n = 838499 por factorización de primos cercanos con 300 vueltas. ¿En cuántas vueltas se factoriza? ¿Qué puede comentar?
18. Repita ahora para una clave similar de 20 bits n = 863407 pero con primos de un tamaño distinto (443, 1949, 25881). ¿Qué sucede ahora?
19. Para la clave (p, q, e) de 48 bits (C7812B, EA6935, 566B) con primos de igual tamaño, factorice solicitando en este caso 50.000 vueltas.

Use el portapapeles Prácticas del tema 14 (5/10)

20. Para la clave (p, q, e) de los apuntes (13, 19, 29) haga un ataque por cifrado cíclico del número N = 123. Compruebe los valores que aparecen en los apuntes.
21. Para la clave (p, q, e) de los apuntes con primos seguros (11, 23, 17) haga un ataque por cifrado cíclico del número N = 123; compruebe los valores que aparecen en los apuntes. ¿Qué pasa si N = 45? ¿Y si N = 46? Limpie la pantalla, genere otros valores de N y repita el ataque.
22. Para la clave (p, q, e) de 24 bits (15217, 907, 635) realice un ataque por cifrado cíclico para los valores N = 7978282 y N = 11537541, con 1000 vueltas. Limpie la pantalla, genere otros valores de N y repita el ataque.
23. Para la clave (p, q, e) de 32 bits (110581, 30851, 20999) realice un ataque por cifrado cíclico para el valor N = 1820641683, con 750 vueltas. Limpie la pantalla, genere otros valores de N y repita el ataque.
24. Para la clave (p, q, e) de 40 bits (0E22A9, 0E65F1, 6B65) realice un ataque por cifrado cíclico para N = C799A8A19F, con 10000 vueltas.

Use el portapapeles Prácticas del tema 14 (6/10)

25. Para la clave (p, q, e) de los apuntes (7, 13, 11) haga un ataque a la clave por paradoja del cumpleaños. Compruebe que obtiene alguna de las claves parejas si N = 33, N = 9, y otros valores.
26. Nota: en algunos casos es posible que se encuentre una "clave pareja" no genérica y que sólo sirve para descifrar ese criptograma. Compruebe que para el mensaje N = 83, se obtiene una "clave" d' = 3 que no es pareja pero que sí descifra el criptograma 51, resultado de la cifra 83^{11} mod 91. Va a suceder lo mismo para N = 8, N = 21, N = 34.
27. Para la clave (p, q, e) de 16 bits (239, 191, 25499) realice un ataque por paradoja del cumpleaños para N = 40872. Limpie la pantalla, genere otros valores de N y repita el ataque.
28. Para la clave (p, q, e) de 20 bits (977, 997, 17405) realice un ataque por paradoja del cumpleaños para N = 516055. Esta operación puede tardar algunos minutos. Active la opción obtener todos los valores. Observe que aquí sí se obtiene la clave privada. Abra y vea el archivo html generado.

Use el portapapeles Prácticas del tema 14 (7/10)

Software ExpoCrip: http://www.criptored.upm.es/software/sw_m001l.htm

1. Para la clave (p, q, e) que se indica (7, 13, 5) observe las claves privadas parejas. Observe otra vez las claves parejas para e = 7, 11, 13, 17.
2. En el caso anterior, ¿qué sucede con los mensajes no cifrables si e = 37?
3. Para la clave (p, q, e) que se indica (127, 557, 337) obtenga los mensajes no cifrables en un archivo de nombre 215mnc.txt. Ordénelos con algún procesador de texto (Word) y observe cómo se distribuyen los valores.
4. Cifre N = 1001. Descífrelo con la clave privada d y la clave pareja 47293. Repítalo con el número hexadecimal N = AB3.
5. Para esta clave, cifre el mensaje M = 123 ABC. Use el portapapeles sobre el criptograma en ASCII y descífrelo. Usando la clave pareja d_p = 47293, descifre nuevamente el criptograma. ¿Y si ahora usa d_p= 47294?
6. Si (p, q, e) = (61, 103, 41) cifre y descifre el mensaje M = Vamos a ver cómo funciona ahora esto. Vea la ayuda del programa para cifra en bloque.

Use el portapapeles Prácticas del tema 14 (8/10)

7. Para cada una de las claves (p, q, e) dadas (59141, 51593, 29571); (59141, 51593, 381393861); (59141, 51593, 1525575441), observe el número de mensajes no cifrables y la relación el valor de e y la función $\phi(n)$. Observe la situación del último caso en el que una de las claves privadas es 1.
8. Para la clave (p, q, e) dada (41, 137, 19) realizar un ataque cíclico introduciendo el mensaje M = A. Repítalo con M = b, M = X.
9. Conociendo sólo n = 29740913, e = 101, realice un ataque por cifrado cíclico al criptograma en ASCII C = ?÷ñ. Observe el texto en claro en el cifrado n-1. Con el programa factorice n y genere la clave RSA. Para obtener el texto en claro, descifre ahora el criptograma C = ?÷ñ.
10. Factorice el módulo n = 29740913. Duplicando el número de dígitos de p y q, obtenemos el módulo de 16 dígitos n = 1479905711715731. Factorice este valor y observe que ahora tardará casi un minuto.
11. Para la clave (p, q, e) dada (41, 137, 19) realizar un ataque por paradoja de cumpleaños introduciendo el mensaje M = Vale. Repítalo varias veces.

Use el portapapeles Prácticas del tema 14 (9/10)

12. Genere un sistema de ElGamal con p = 43, α = 5, x = 13, k = 18. ¿Qué sucede si introduce como generador α = 6 y da al tabulador?
13. Con la misma aplicación, observe todos los generadores del cuerpo 43.
14. Con la clave anterior, cifre el número N = 25. Ahora con la ayuda del portapapeles, descifre el criptograma obtenido. Compruebe con la calculadora de Windows todos los valores encontrados.
15. Con la misma clave cifre el mensaje M = Hola qué tal! Con la ayuda del portapapeles, descifre el criptograma. Modifique en el criptograma el octavo carácter B por una C y observe lo que sucede con el descifrado.
16. Observe que en el cifrado hexadecimal, al igual que con números, se cifra sólo dentro del cuerpo de cifra, en este caso p = 43 que corresponde a un CCR = {0, 1, ... 42} es decir desde el valor 00 al valor 2A. Si ciframos 2B y los siguientes valores, volvemos a obtener el mismo cifrado módulo 43.
17. Con la clave p = 564387560286133, α = 128, x = 3217, k = 8752, cifre distintos valores numéricos y observe el criptograma.

Use el portapapeles Prácticas del tema 14 (10/10)

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\bin> y desde el símbolo del sistema MSDOS genere una clave RSA de 1.024 bits mediante el comando: openssl genrsa 1024. Observe que una vez 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".