CRIPTOGRAFIA APLICADA

Ing Germán Bollmann

1

AGENDA

Clave pública / privada

Intercambio de Claves Diffie-Hellman

Esquema General de Cifrado con Clave Pública

RSA

- Cifrado
- Descifrado

Ataques RSA

1) Factorización de N

2) Exponente pequeño

3) Ataque Cíclico

4) Paradoja del Cumpleaños (firma digital)

USOS REALES DE RSA

2

Clave pública / privada

3

Clave Privada

Problemas

- **Confidencialidad**
  - **Administración de las Claves:**
    - gestión
    - almacenamiento
  - **Distribución de las Claves**
- **Integridad**
  - **HASH: NO Impide** la modificación maliciosa
    MAC: mismos problemas de la confidencialidad

4

DIFFIE-HELLMAN
New Directions in Cryptography

New Directions in Cryptography

Invited Paper

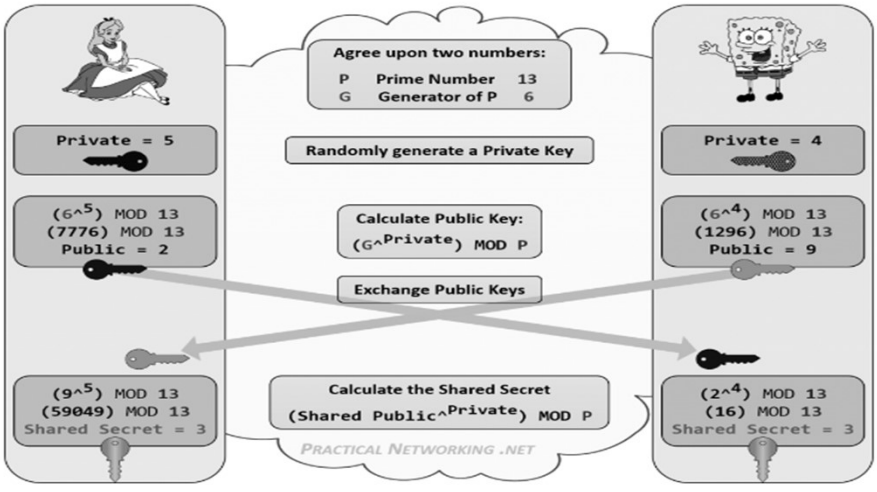
WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

I. INTRODUCTION

https://ee.stanford.edu/~hellman/publications/24.pdf

5

Esquema de Intercambio de Claves Diffie-Hellman

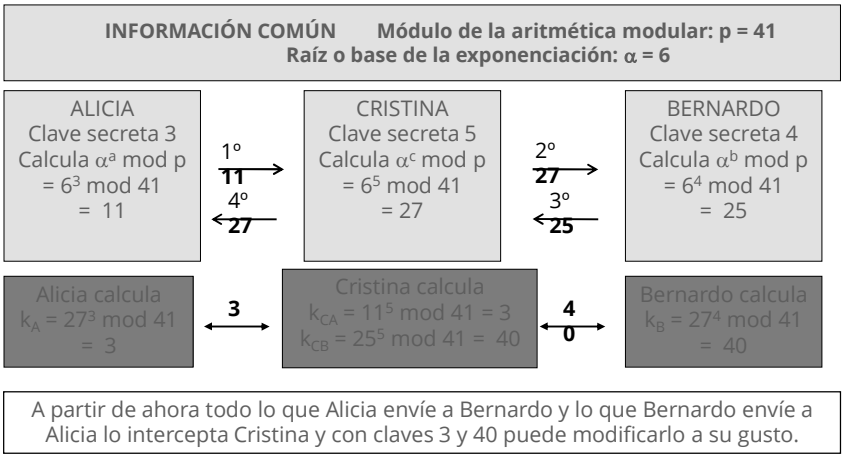


https://ee.stanford.edu/~hellman/publications/24.pdf

Alice y Bob ya comparten una **clave secreta** calculada en cada extremo de la comunicación, imposible de obtener aún habiendo interceptado las comunicaciones intercambiadas **(Problema del Logaritmo Discreto).**

6

Man in the middle MITM



7

Seguridad de DH: elección de parámetros seguros

La fortaleza del sistema radica en dificultad matemática de resolver el ***Logaritmo Discreto PLD***. Dados los avances en **potencia de cómputo**, los **sistemas y computación distribuídos**, como así también en los **algoritmos para bajar la complejidad computaciónal al PLD**, se recomiendan una serie de medidas en la eleccion de ***p*** para mitigar los ataques.

1- Debe tener un tamaño «**muy**» grande.
2- No debe estar **«cerca»** de una potencia de *2*.
3- *(p-1)* tendría que tener un divisor q muy grande ***(p=2q+1)*** ya que los ataques por Logaritmo Discreto tienen una complejidad computacional del orden de $\mathbf{2^{r/2}}$ siendo r el **tamaño del divisor de primo de q** mayor. Si ***q* fuera primo**, entonces el ataque tiene máxima complejidad.
4- Los exponentes aleatorios se deben tomar entre ***1*** y ***(q – 1).***
Si ***q* es primo y mayor que *256* bits**, es suficiente elegir un valor aleatorio de *256* bits para lograr ***128* bits de seguridad.**
Si *q* no es primo, tiene un tamaño ***r*** bits y su mayor divisor primo tiene ***e*** bits, y ***e* ≥ *256***, entonces se eligen exponentes aleatorios de tamaño ***r - (e - 256)*** bits para obtener la "**seguridad de 128 bits**" habitual.

https://www.it-swarm.dev/es/diffie-hellman/cual-es-el-estado-actual-de-seguridad-del-intercambio-de-claves-diffie-hellman/l968274357/

8

Grupos Diffie-Hellman

En el “mundo real” práctico y estandarizado, se usan módulos y generadores establecidos. Como por ejemplo para **IPSEC:**

Value, group Description, Reference, Note

0, Reserved
1, default 768-bit MODP group, [RFC2409], Section 6.1
2, alternate 1024-bit MODP group, [RFC2409], Section 6.2
3, EC2N group on GP[2^155], [RFC2409], Section 6.3
4, EC2N group on GP[2^185], [RFC2409], Section 6.4
5, 536-bit MODP group, [RFC3526], Section 2
6, EC2N group over GF[2^163], draft-ietf-ipsec-ike-ecc-groups], Section 2.1
7, EC2N group over GF[2^163], [draft-ietf-ipsec-ike-ecc-groups], Section 2.2
8, EC2N group over GF[2^283], [draft-ietf-ipsec-ike-ecc-groups], Section 2.3
9, EC2N group over GF[2^283], [draft-ietf-ipsec-ike-ecc-groups], Section 2.4
10, EC2N group over GF[2^409], [draft-ietf-ipsec-ike-ecc-groups], Section 2.5
11, EC2N group over GF[2^409], [draft-ietf-ipsec-ike-ecc-groups], Section 2.6
12, EC2N group over GF[2^571], [draft-ietf-ipsec-ike-ecc-groups], Section 2.7
13, EC2N group over GF[2^571], [draft-ietf-ipsec-ike-ecc-groups], Section 2.8
14, 2048-bit MODP group, [RFC3526], Section 3
15, 3072-bit MODP group, [RFC3526], Section 4
16, 4096-bit MODP group, [RFC3526], Section 5
17, 6144-bit MODP group, [RFC3526], Section 6
18, 8192-bit MODP group, [RFC3526], Section 7
19, 256-bit random ECP group, [RFC5903]
20, 384-bit random ECP group, [RFC5903]
21, 521-bit random ECP group, [RFC5903]
22, 1024-bit MODP group with 160-bit Prime Order Sub group, [RFC5114],
23, 2048-bit MODP group with 224-bit Prime Order Sub group, [RFC5114],
24, 2048-bit MODP group with 256-bit Prime Order Sub group, [RFC5114],
25, 192-bit Random ECP group, [RFC5114],
26, 224-bit Random ECP group, [RFC5114],
27, 224-bit Brainpool ECP group, [RFC6932], Section 2.1. Not for RFC 2409.
28, 256-bit Brainpool ECP group, [RFC6932], Section 2.2. Not for RFC 2409.
29, 384-bit Brainpool ECP group, [RFC6932], Section 2.3. Not for RFC 2409.
30, 512-bit Brainpool ECP group, [RFC6932], Section 2.4. Not for RFC 2409.

31-32767, Unassigned, ,

32768-65535, Reserved for private use

1- MODP (modular exponentiation group)
2- ECP (elliptic curve group over GF[P])
3- EC2N (elliptic curve group over GF[2^N])

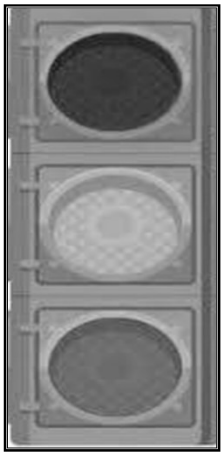
Grupo 1, Exponenciación Modular, (en hexa)
p = FFFFFFFF FFFFFFFF C90FDAA2 2168C234
C4C6628B 80DC1CD1 29024E08 8A67CC74 020BBEA6
3B139B22 514A0879 8E3404DD EF9519B3 CD3A431B
302B0A6D F25F1437 4FE1356D 6D51C245 E485B576
625E7EC6 F44C42E9 A63A3620 FFFFFFFF FFFFFFFF

g = 2

https://www.iana.org/assignments/ipsec-registry/ipsec-registry.xhtm
https://tools.ietf.org/html/rfc2409

9

Seguridad de DH: elección de parámetros



1 - 768 bit modulus - AVOID
2 - 1024 bit modulus - AVOID
5 - 1536 bit modulus - AVOID
24 - modular exponentiation group with a 2048-bit modulus and 256-bit prime order sub group – Next Generation Encryption

14 - 2048 bit modulus – MINIMUM ACCEPTABLE

19 - 256 bit elliptic curve – ACCEPTABLE
20 - 384 bit elliptic curve – Next Generation Encryption
21 - 521 bit elliptic curve – Next Generation Encryption

https://www.it-swarm.dev/es/diffie-hellman/grupo-diffie-hellman-que-coincide-con-el-algoritmo-de-cifrado-ipsec/l968329874/

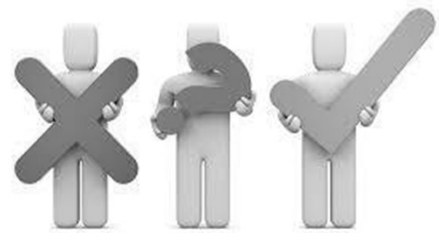
10

Algunas recomendaciones

Cisco
"Cuando sea posible, use … los … grupos ECDH"

CheckPoint
"Los grupos Diffie-Hellman de curva elíptica … proporcionan un mejor rendimiento"
"Los grupos descritos en RFC 5114 (***Grupo 24*** …) NO SE RECOMIENDAN para su uso"

IBM. Directriz
Si está utilizando algoritmos de encriptación o autenticación con una clave de 128 bits, use los grupos Diffie-Hellman 5,14,19 , 20 o **24**.
Si está utilizando algoritmos de encriptación o autenticación con una longitud de clave de 256 bits o mayor, use el grupo 21 de Diffie-Hellman "



https://www.it-swarm.dev/es/diffie-hellman/grupo-diffie-hellman-que-coincide-con-el-algoritmo-de-cifrado-ipsec/l968329874/
https://tools.cisco.com/security/center/resources/next_generation_cryptography

11

Solaris 10 7/07

En mayo de 2007 Diffie ascendió a jefe oficial de seguridad y vicepresidente de Sun Microsystems.

S.O. de Sun Microsystems, ahora en manos de Oracle al comprar Sun.

A partir de la versión Solaris 10 7/07, el contenido de Solaris Encryption Kit se instala mediante el disco de instalación de Solaris. En esta versión se añaden los algoritmos de autenticación SHA2: sha256, sha384 y sha512. Las implementaciones SHA2 cumplen la especificación RFC 4868. Esta versión también agrega grupos Diffie-Hellman más grandes: 2048 bits (grupo 14), 3072 bits (grupo 15) y 4096 bits (grupo 16). Tenga en cuenta que los sistemas de Sun con tecnología CoolThreads sólo aceleran los grupos de 2048 bits.

Antes de la versión Solaris 10 7/07, el disco de instalación de Solaris proporciona algoritmos básicos, además puede añadir algoritmos más complejos desde Solaris Encryption Kit.

De modo predeterminado, están instalados los algoritmos DES-CBC, 3DES-CBC, AES-CBC, y Blowfish-CBC. Los tamaños de claves que admiten los algoritmos AES-CBC y Blowfish-CBC están limitados a 128 bits.

Los algoritmos AES-CBC y Blowfish-CBC que admiten tamaños de claves de más de 128 bits están disponibles para IPsec cuando se instala el Solaris Encryption Kit. Sin embargo, no todos los algoritmos de cifrado están disponibles fuera de Estados Unidos. El kit está disponible en un CD independiente que no forma parte del paquete de instalación de Solaris 10. En la Solaris 10 Encryption Kit Installation Guide se describe cómo instalar el kit. Para obtener más información, visite el sitio web de descargas de Sun.

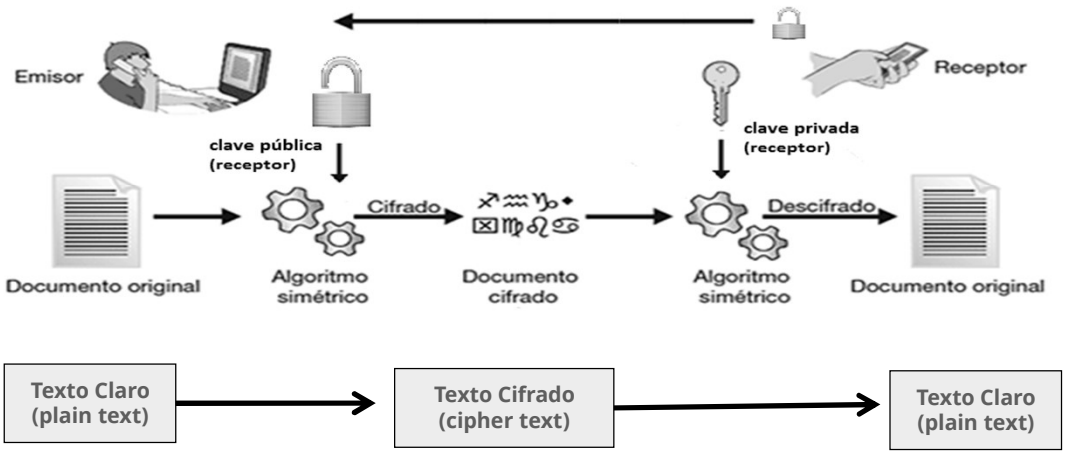
https://docs.oracle.com/cd/E19957-01/820-2981/ipsec-ov-11/index.html

12

RSA

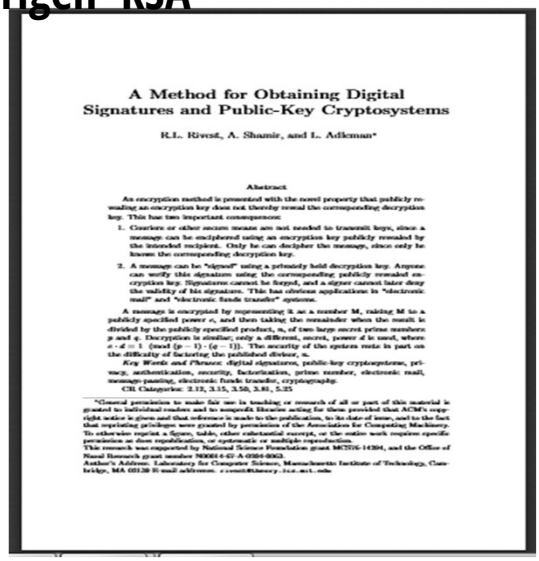
13

Esquema General de Cifrado con Clave Pública



14

Origen RSA



A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

R.L. Rivest, A. Shamir, and L. Adleman*

Abstract

https://people.csail.mit.edu/rivest/Rsapaper.pdf

El algoritmo fue descrito en 1977 por Ron Rivest, Adi Shamir y Leonard Adleman, del Instituto Tecnológico de Massachusetts (MIT); las letras RSA son las iniciales de sus apellidos. https://es.wikipedia.org/wiki/RSA

15

Algoritmo re-inventado

*Clifford Cocks, un matemático británico que trabajaba para la agencia de inteligencia británica **GCHQ**, había descrito un sistema equivalente en un documento interno en **1973**. Debido al elevado coste de las computadoras necesarias para implementarlo en la época su idea no trascendió. Su descubrimiento, sin embargo, no fue revelado hasta 1997 ya que era confidencial, por lo que Rivest, Shamir y Adleman desarrollaron RSA de forma independiente.* https://es.wikipedia.org/wiki/RSA

*Clifford Christopher Cocks CB FRS (born 28 December 1950) is a British mathematician and cryptographer. In 1973, while working at the United Kingdom **Government Communications Headquarters (GCHQ), he invented a public key cryptography algorithm equivalent to what would become (in 1978) the RSA algorithm.***

https://en.wikipedia.org/wiki/Clifford_Cocks

16

RSA Generando los Parámetros del Criptosistema

1) Se eligen **p y q primos grandes (hoy mínimo 1024 bits en adelante)**
Se recomienda que sean "primos fuertes. Aunque papers posteriores indicaron que no eran necesario.

2) Se calcula N=p*q que se llama **"Módulo Público"**

3) Se calcula **φ(N)** (Función Phi de Euler).

4) En el grupo multiplicativo (φ(N), X) se eligen ***e*** y ***d*** tales que sean inversos o primos relativos.

(p-1)*(q-1) (recordar que primos relativos significa que *e* no tiene factores comunes con (p-1) y (q-1)).
Finalmente se selecciona *d* (operación inversa de *e*) de modo tal que
e *d = 1 mod ((p-1)*(q-1)) (para hallar d se utiliza un algoritmo)
Clave Pública: (e, N)
Clave Privada (d, N) que se debe conservar secreta y segura

1) p=3, q=11

2) **N = 3*11**
N = 33

3) φ(N) =(p-1)*(q-1)
2*10
φ(N)=20
4) e =3 d=7
Pues 3*7≡ 1(mód 20)
e=3 y d=7

Clave Pública: (3, 33)
Clave Privada (7, 33)

17

"Cifrando" con RSA

1) Sea m el mensaje a cifrar
RSA se basa en el Teorema de Euler-Fermat se requiere que mcd(m, N)=1
La mayoría de las explicaciones, se "olvidan" de mencionarlo o lo dan por supuesto.

2) El emisor consigue el par (e, N) y es por eso que puede calcular c≡ m^e mód (N). O sea que Cifrar(m)= m^e mód (N).

3) El emisor envía c.

1) m=5
mcd(5;33)=1

2) Conozco (3;33)
c ≡ 5^3 mód (33)
≡ 125 mód (33)
≡ 26 mód (33)

3) Envía 26

18

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.

19

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.

20

"Descifrando" RSA

1) Sea c el mensaje recibido

2) El receptor recurre a su clave privada d.
Calcula $m \equiv c^d$ mód (N).
O sea que Descifrar(c)= c^d mód (N).

3) El receptor recupera m

1) c=26

2) Conozco (7;33)
$m \equiv 26^7$mód (33)
$\equiv$ 8031810176 mód (33)
$\equiv$ 5 mód (33)

OBSERVACIÓN:
8031810176=33* 243388187+5

3) Descifra m=5

21

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...
$55^{29} = 29547313175564474880964247600939124822616577148437 5$
¿Cómo podemos acelerar esta operación?

1ª opción: usar reducibilidad

2ª opción: algoritmo exp. rápida

Opción óptima: usar el Teorema del Resto Chino

22

Ataques RSA


Twenty Years of Attacks on the RSA Cryptosystem

Dan Boneh

1 Introduction

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

- 1) Factorización de N
- 2) Exponente pequeño
- 3) Ataque Cíclico
- 4) Paradoja del Cumpleaños (firma digital)

23

1)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

24

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:

$p = (x+y)$; $q = (x-y)$.

25

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 ↓

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

   Entonces: $p = x - y = 216 - 35 = 181$

   $q = x + y = 216 + 35 = 251$ 👍

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

26

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.

27

¿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

28

2) Exponente pequeño
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.

29

3) 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.

30

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?

31

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	C_i
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).

32

4) 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.

33

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

34

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.

35

¿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 ☺.**

36

USOS REALES DE RSA

RSA

COBERTURA RSA
Enviar una clave privada de forma segura y luego enviar la información cifrada mediante criptografía simétrica

FIRMA DIGITAL
Dotar de Integridad y Autenticación a un documento

37

Ataques teóricos al algoritmo RSA

a) Ataque basado en la factorización del módulo n
Prácticas con software genRSA V2.1

b) Ataque por cifrado cíclico con la clave pública
Prácticas con software genRSA v2.1 y RingRSA

c) Ataque basado en la paradoja del cumpleaños
Prácticas con software genRSA v2.1 y LegionRSA

Si prosperan:
a) Se encuentra p y q, con ello $\phi(n)$ y luego fácilmente se calcula la clave privada d.
b) Se rompe el secreto sin necesidad de usar la clave privada d.
c) Se encuentra la clave privada d o una clave privada pareja.

38

UNDEF Universidad de la Defensa Nacional

FIRMA

39

Problemas a resolver

Autenticidad del emisor
Integridad del mensaje
Actualidad (no replay)
No repudio (del emisor y receptor)
Detección de usurpación de identidad
Deben ser verificables por terceros

40

FIRMA

En muchos sistemas prácticos, se utiliza criptografía asimétrica

Si ciframos mensaje con la clave privada, alcanza con verificar que es descifrado correctamente con la clave pública

Los algoritmos asimétricos son lentos y se generarían firmas tan largas como el mensaje

-> Se cifra un hash del mensaje con la clave privada

41

Ejemplo de Procedimiento Usando RSA

- Sea d_a y e_a las claves pública y privada de A
- A envía a B el mensaje M (cifrado o no), y $E_{e_a}(H(M))$
- El receptor calcula $H' = D_{d_a}(E_{e_a}(H(M)))$, calcula H(M'), y los compara
- Si coinciden, tiene la "seguridad" de que solo A pudo generar el mensaje
  - No se puede generar otro mensaje con el mismo hash
  - Solo quien conoce e_a pudo generar $E_{e_a}(H(M))$

42

Problemas y soluciones

¿Cómo distribuir la clave pública?

¿Cómo estoy seguro que la clave pública que tengo es realmente la de A, y que es actual?

Repudio....¿Qué pasa si A alega que el no firmó? ¿y si alega que le robaron la clave?

Soluciónes

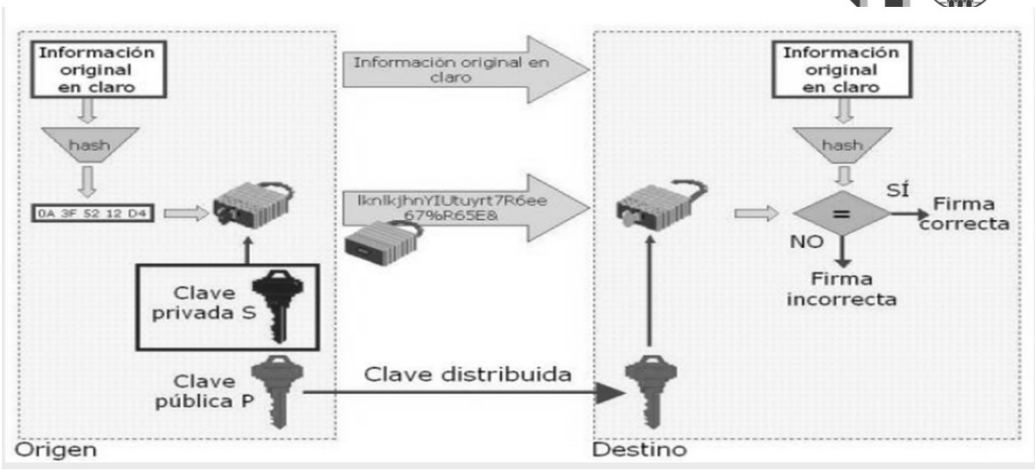
Exigir que cada mensaje lleve un timestamp, y exigir reporte “inmediato” de claves comprometidas a una autoridad central

Utilizar un “árbitro”, una entidad confiable que certifique el origen y contenido del mensaje

- Se puede hacer con algoritmos simétricos o asimétricos
- Dependen de encontrar la entidad confiable

43

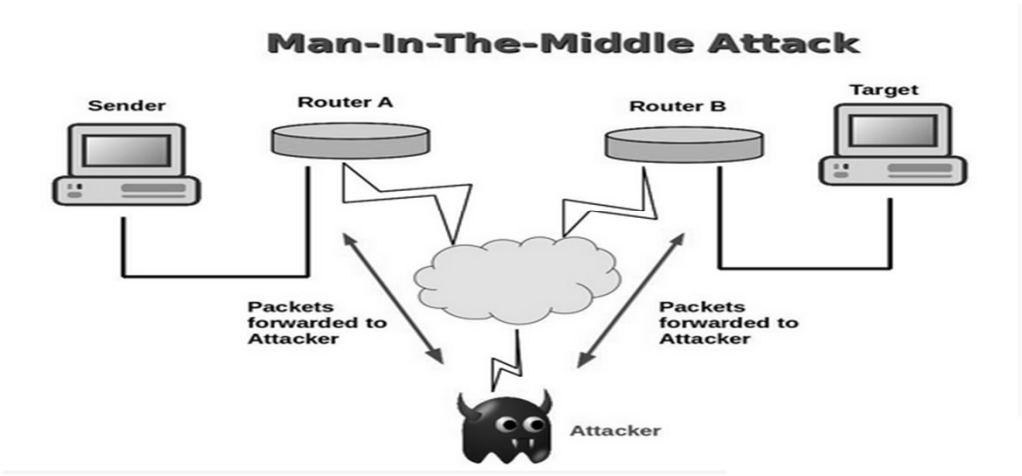
Esquema de Firma Digital RSA



https://ee.stanford.edu/~hellman/publications/24.pdf

44

Ataque Man in the Middle (MITM)



45

¡Gracias!

46