Objetivo

Comprender el funcionamiento de los algoritmos criptográficos simétricos y asimétricos.

Xifratge simètric

Si queremos enviar información de forma segura se puede proteger el contenido con una contraseña. La misma contraseña sirve tanto para encriptar como para desencriptar.

Problema: los dos participantes en una comunicación han de poseer la contraseña.
Esto lleva a otro problema: cómo hacer llegar la contraseña al otro participante en la comunicación.

Xifratge asimètric

Una solución a lo anterior son los algoritmos con dos contraseñas: uno para cada participante en la comunicación. Las contraseñas son complementarias:
Si genero una pareja de contraseñas, una me la guardo y la otra la publico: son la clave privada y la clave pública.

Tasca

Podemos intentar ponerlas en uso con un ejercicio sencillo sobre papel con algoritmos triviales para ver todos los pasos necesarios y comprender con detalle el funcionamiento.

Algoritmos simétricos (una clave):

- El más antiguo es el algoritmo del César: muy simple, cambiar letras, muy compacto: la clave consiste en un número (el desplazamiento de cada letra), muy poco robusto: sólo hay que probar cerca de 30 veces para encontrar el mensaje original. (la función inversa: probar todos los casos no requiere un gran esfuerzo).

  p.ex. k=3

lletra	a	b	c	d	e	f	g	...	w	x	y	z
Lletra xifrada	d	e	f	g	h	i	j	...	z	a	b	c

- Un codificador mono-alfabético es una mejora de lo anterior, pero muchísimo más robusto: vamos construyendo un diccionario y a cada letra del alfabeto le asignamos otra: por ejemplo: A (elegimos la M entre 28 posibilidades), B (elijo la J entre 27 letras que quedan para elegir), C (elijo la Z entre 26 …): combinaciones: 28! que es muy grande! Del orden de 10^{28} …
  La contraseña es el diccionario.

Symmetric key cryptography

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another

```
plaintext:   abcdefghijklmnopqrstuvwxyz
             |                       |
ciphertext:  mnbvcxzasdfghjklpoiuytrewq
```

E.g.: Plaintext: bob. i love you. alice
ciphertext: nkn. s gktc wky. mgsbc

Q: How hard to break this simple cipher?:

- brute force (how hard?)
- other?

Vemos que además el mono-alfabético es rápido: sólo hay que consultar una tabla (parece que en ensamblador se podría hacer en pocas instrucciones máquina …).

Pero tiene el problema de cómo hacer llegar al destinatario el diccionario para que sólo él pueda leer el mensaje.

Algoritmos asimétricos (dos claves):

Public key encryption algorithms

Requirements:

1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that
   $K_B^-(K_B^+(m)) = m$
2. given public key K_B^+, it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adelson algorithm

Vamos a ver cómo funciona RSA

RSA: Choosing keys

1. Choose two large prime numbers p, q. (e.g., 1024 bits each)
2. Compute n = pq, z = (p-1)(q-1)
3. Choose e (with e<n) that has no common factors with z. (e, z are "relatively prime").
4. Choose d such that ed-1 is exactly divisible by z. (in other words: ed mod z = 1).
5. Public key is (n,e). Private key is (n,d). (K_B^-)

Exemple:

1. Elegir dos números primos p y q (recomiendan > 1024) p=3, q=5

2. Calcular n=p·q; z=(p-1)·(q-1) n=15, z=2·4=8

3. e<n, e primo relativo z e=11

4. d tal que ed-1 divisible por z.

Es decir: e·d-1%z=0 o ed-1=z·c, d=(z·c+1)/e c=1 d=(z+1)/e=9; c=2 d=17/11; c=3 d=25/11; c=4 d=33/11=3 → d=3

Dos claves complementarias (n,e), (n,d)

Por ejemplo:

Clave pública (n,e) (15, 11)

Clave privada (n,d) (15, 3)

RSA: Encryption, decryption

0. Given (n,e) and (n,d) as computed above

1. To encrypt bit pattern, m, compute
   $c = m^e \bmod n$ (i.e., remainder when m^e is divided by n)

2. To decrypt received bit pattern, c, compute
   $m = c^d \bmod n$ (i.e., remainder when c^d is divided by n)

Magic happens! $m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$

RSA example:

Bob chooses p=5, q=7. Then n=35, z=24.
e=5 (so e, z relatively prime).
d=29 (so ed-1 exactly divisible by z.

	letter	m	m^e	$c = m^e \bmod n$
encrypt:	l	12	1524832	17

	c	c^d	$m = c^d \bmod n$	letter
decrypt:	17	48196857210675091509141182522307169 7	12	l

A cada letra le asignamos un código (0 y 1 son malos pues rsa no los codifica bien)

a→2, b→3, …

Texto	M	m^e	$c=m^e \bmod n$	c^d	$m=c^d \bmod n$	
	0	0	0	0	0	
	1	1	1	1	1	
a	2	2048	8	512	2	A
a	1	i^e=1	c=1	c^e=1	m=1	
b	2	i^e=2048	c=8	c^e=8589934592	m=2	
c	3	i^e=177147	c=12	c^e=743008370688	m=3	
d	4	i^e=4194304	c=4	c^e=4194304	m=4	
e	5	i^e=48828125	c=5	c^e=48828125	m=5	
f	6	i^e=362797056	c=6	c^e=362797056	m=6	
g	7	i^e=1977326743	c=13	c^e=1792160394037	m=7	
h	8	i^e=8589934592	c=2	c^e=2048	m=8	
i	9	i^e=31381059609	c=9	c^e=31381059609	m=9	
j	10	i^e=100000000000	c=10	c^e=100000000000	m=10	
k	11	i^e=285311670611	c=11	c^e=285311670611	m=11	
l	12	i^e=743008370688	c=3	c^e=177147	m=12	
m	13	i^e=1792160394037	c=7	c^e=1977326743	m=13	
n	14	i^e=4049565169664	c=14	c^e=4049565169664	m=14	
o	15	i^e=8649755859375	c=0	c^e=0	m=0	

Sólo se pueden enviar las letras de la a – n.

Es decir, al tomar módulo n, el valor de n determina el tamaño del bloque (el número de mensajes posibles). Normalmente se trabaja con bloques mucho más grandes, si no sería muy predecible.

RSA: Why is that $m = (m^e \bmod n)^d \bmod n$

Useful number theory result: If p,q prime and n = pq, then:

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

(using number theory result above)

$$= m^1 \bmod n$$

(since we chose ed to be divisible by (p-1)(q-1) with remainder 1)

$$= m$$

RSA: another important property

The following property will be very useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key}}$$

Result is the same!

Altres exemples

Exemple 1
p=q=3; n=9; e=7; d=3
1 i^e=1 c=1 c^e=1 m=1
2 i^e=128 c=2 c^e=128 m=2
3 i^e=2187 c=0 c^e=0 m=0
4 i^e=16384 c=4 c^e=16384 m=4
5 i^e=78125 c=5 c^e=78125 m=5
6 i^e=279936 c=0 c^e=0 m=0
7 i^e=823543 c=7 c^e=823543 m=7
8 i^e=2097152 c=8 c^e=2097152 m=8
9 i^e=4782969 c=0 c^e=0 m=0
10 i^e=10000000 c=1 c^e=1 m=1
11 i^e=19487171 c=2 c^e=128 m=2
12 i^e=35831808 c=0 c^e=0 m=0
13 i^e=62748517 c=4 c^e=16384 m=4
14 i^e=105413504 c=5 c^e=78125 m=5
i^e=170859375 c=0 c^e=0 m=0
El anterior es un mal código … tiene ceros (i=3, i=6) como n es 9 sólo caben unos pocos casos.

Exemple 2
p=5, q=7
n=35; z=24
e=5
d=29
i=1 i^e=1 c=1 c^e=1 m=1
i=2 i^e=32 c=32 c^e=33554432 m=2
i=3 i^e=243 c=33 c^e=39135393 m=3
i=4 i^e=1024 c=9 c^e=59049 m=4
i=5 i^e=3125 c=10 c^e=100000 m=5
i=6 i^e=7776 c=6 c^e=7776 m=6
i=7 i^e=16807 c=7 c^e=16807 m=7
i=8 i^e=32768 c=8 c^e=32768 m=8
i=9 i^e=59049 c=4 c^e=1024 m=9
i=10 i^e=100000 c=5 c^e=3125 m=10
i=11 i^e=161051 c=16 c^e=1048576 m=11
i=12 i^e=248832 c=17 c^e=1419857 m=12
i=13 i^e=371293 c=13 c^e=371293 m=13
i=14 i^e=537824 c=14 c^e=537824 m=14
i=15 i^e=759375 c=15 c^e=759375 m=15

Qué hay que hacer:

Usando las versiones más sencillas de los algoritmos (simétrico César, RSA con claves mínimas) intentar enviar un secreto desde A a C sin que B pueda verlo.

Puede hacerse primero diciéndole A a C el número en la oreja (sin que lo oiga B) y usar el algoritmo del César.

Después puede enviarse ese mismo número de A a C usando RSA (cada participante elige una pareja de claves de las vistas en los ejemplos anteriores) y luego un mensaje codificado con ese número (ahora se puede hacer todo a distancia, sin necesidad de acercarse físicamente A y C.

También se podría haber hecho todo con RSA pero sería en la realidad más costoso en cálculo.

Qué pasaría si B fuera malicioso y cambiara los mensajes? Por ejemplo hiciera creer a A que él es C y le diera B su clave pública a A en lugar de la clave pública de C?
Se puede detectar? Se puede evitar?

B está en el medio de comunicación: puede ser bueno y hacer bien su trabajo: copiar de la columna izquierda a la derecha y viceversa, puede copiar y leer lo que viaja, puede transformar el contenido de una columna a otra.

B es "malicioso", podría sólo intentar leer lo que pasa por el medio, pero también podría cambiar el mensaje: Mensaje1' ≠ Mensaje1 o Mensaje2' ≠ Mensaje2.

Cómo hacerlo?

En grupos de 3 personas se reparten los roles de A, B y C.
Cada participante hace su trabajo: A y C pueden publicar su clave pública en Apub y Cpub. Los secretos los apuntan en su columna privada: Apriv y Cpriv.
B puede simplemente copiar de la columna izquierda a derecha y viceversa, pero también puede mirar: apuntar lo que ve en su columna privada, o copiar algo distinto de un lado a otro de la comunicación.

Entregar la siguiente hoja con vuestra solución

(Enviar a aad@ac.upc.es. Subject: **Sessió Algorismes simètrics i asimètrics)**

Nombres:_____, _____, _____

Aquí A escribe cosas que **sólo** A puede ver	Aquí A escribe cosas que **otros** pueden ver	Aquí está lo que viene de C o va hacia C. Mensaje1→ ←Mensaje 2'	Aquí B escribe cosas que **sólo** B puede ver	Aquí está lo que viene de C o va hacia C. Mensaje1'→ ←Mensaje2	Aquí C escribe cosas que **otros** pueden ver	Aquí C escribe cosas que **sólo** A puede ver
Apriv	Apub	B	B	B	Cpriv	Cpub