UCLM
UNIVERSIDAD DE CASTILLA-LA MANCHA

Consejería de Educación, Cultura y Deportes
Castilla-La Mancha

DSI
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS

escuela superior ingeniería informática

Módulo 1: Criptografía y Aplicaciones Criptográficas

Sesión 1

José Luis Martínez
Universidad de Castilla–La Mancha

Índice

- Píldora 1:
  - Principios de la Criptografía
  - Criptoanálisis
  - Criptografía clásica
- Píldora 2:
  - Comparativa
  - Cifrado Simétrico: DES y AES
  - Cifrado Asimetrico: RSA
  - Diffie & Hellman

La criptografía según la RAE

He aquí una definición no muy afortunada...

La Real Academia Española define criptografía (del griego: oculto + escritura) como:

"*el arte de escribir con clave secreta o de modo enigmático*".

Puede ser interesante y llamativa, pero resulta muy poco ajustada para los tiempos actuales.

Principios de la Criptografía

- Imprecisiones de esta definición:

  el arte de escribir con clave secreta o de modo enigmático

  - **Arte:** la criptografía ha dejado de ser un arte: es una ciencia.
  - **Escritura de documentos:** no sólo se escriben mensajes; se envían o se guardan en un computador diversos tipos de documentos y formatos (TXT, DOC, EXE, DLL, JPG, ...).
  - **Se supone una clave:** los sistemas actuales usan una o dos. En varias aplicaciones de Internet entran en juego 4 claves.
  - **Clave secreta:** existirán sistemas de clave secreta que usan una sola clave y sistemas de clave pública (muy importantes) que usan dos: una clave privada (secreta) y la otra pública.
  - **Representación enigmática:** la representación binaria de la información podría ser enigmática para nosotros los humanos pero no para los computadores ☺ ... es su lenguaje natural.

Principios de la Criptografía

- Una definición más técnica de criptografía 

- Rama inicial de las Matemáticas y en la actualidad también de la Informática y la Telemática, que **hace uso de métodos y técnicas con el objeto principal de cifrar**, y por tanto proteger, un mensaje o archivo por medio de un algoritmo, usando una o más claves.

- Esto dará lugar a **diferentes tipos de sistemas de cifra**, denominados criptosistemas, que nos permiten asegurar al menos tres de los cuatro aspectos básicos de la seguridad informática: **la confidencialidad o secreto del mensaje**, **la integridad del mensaje y autenticidad del emisor**, así como el **no repudio** mutuo entre emisor (cliente) y receptor (servidor).

Principios de la Criptografía

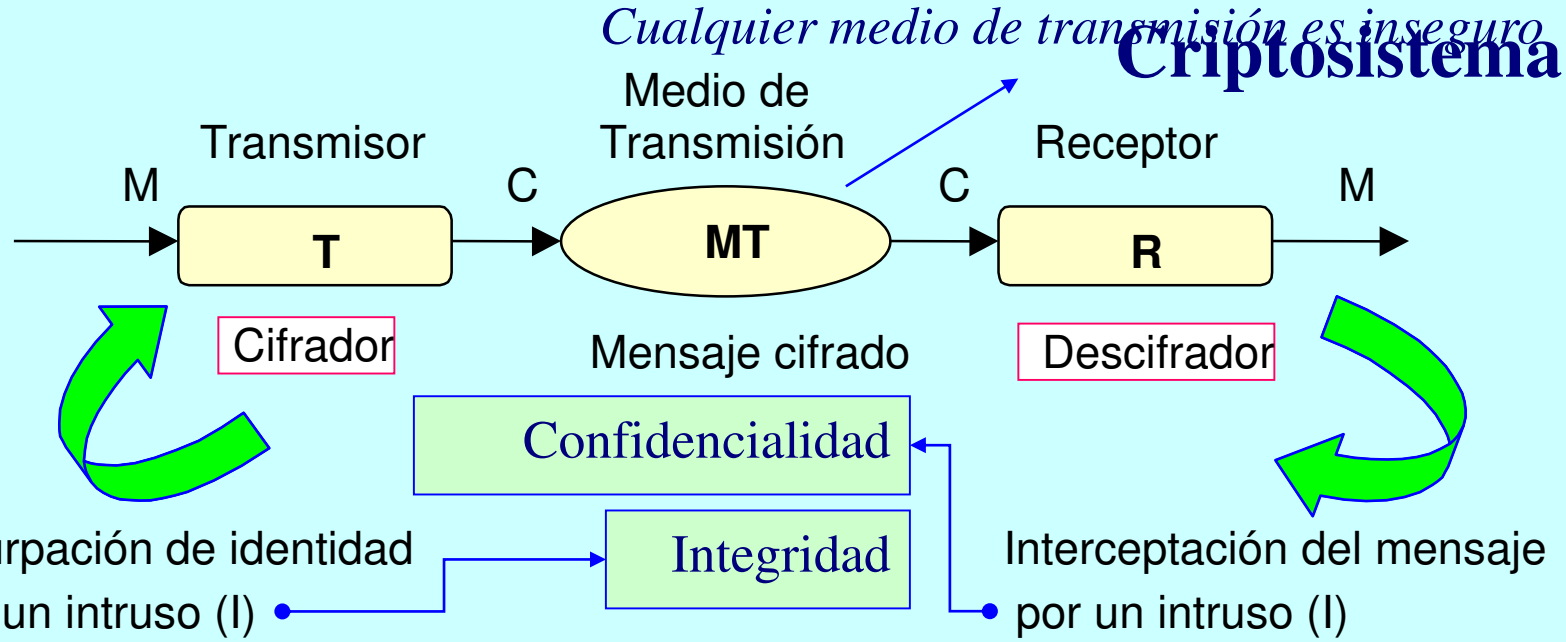
- Mas definiciones:
  - **Criptografía**
    - La ciencia que permite cifrar mensajes
  - **Criptoanálisis**
    - La ciencia de descifrar mensajes cifrados
  - **Criptología**
    - La ciencia que incluye a ambas

Principios de la Criptografía

- Principios de la seguridad
  - **Confidencialidad:** únicamente el emisor y el receptor deseado deben "entender" el contenido del mensaje.
    - Emisor encripta el mensaje.
    - Receptor desencripta el mensaje.
  - **Autenticación:** emisor y receptor quiere confirmar la identidad de cada uno.
  - **Integridad del mensaje:** emisor y receptor quieren estar seguros de que el contenido de sus comunicaciones no es alterado (durante la transmisión o después) sin detección.
  - **Disponibilidad y acceso:** los servicios deben ser accesibles y deben estar disponibles para los usuarios.

Principios de la Criptografía

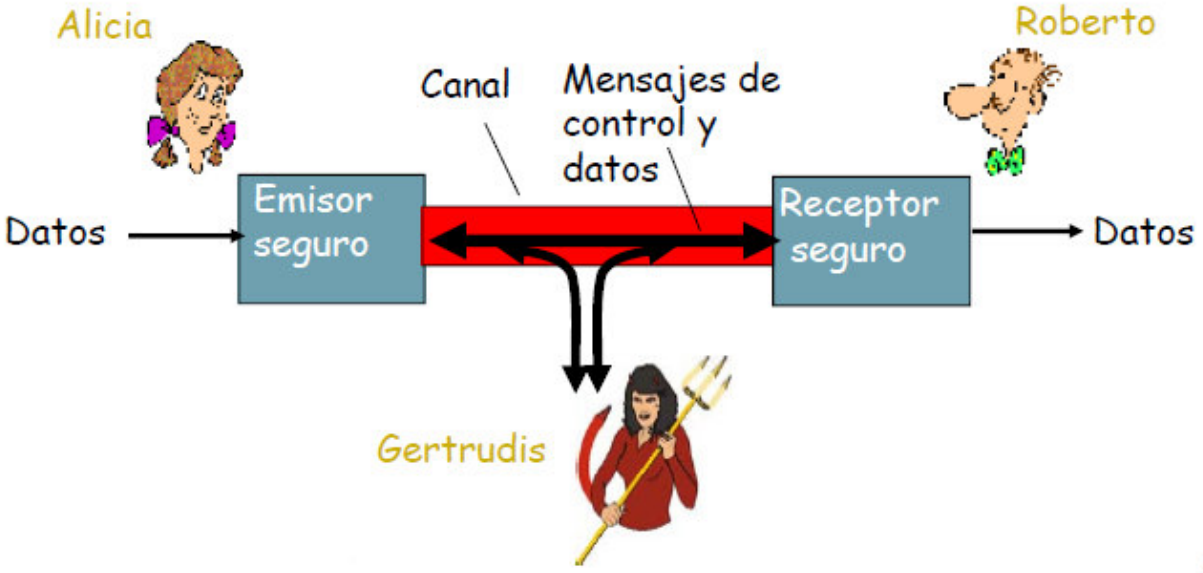
Confidencialidad e integridad



Estos dos aspectos básicos de la seguridad informática, el de la **confidencialidad** y el de **integridad** (además de la disponibilidad del sistema y el no repudio) serán muy importantes en un entorno de intercambio de información segura a través de Internet.

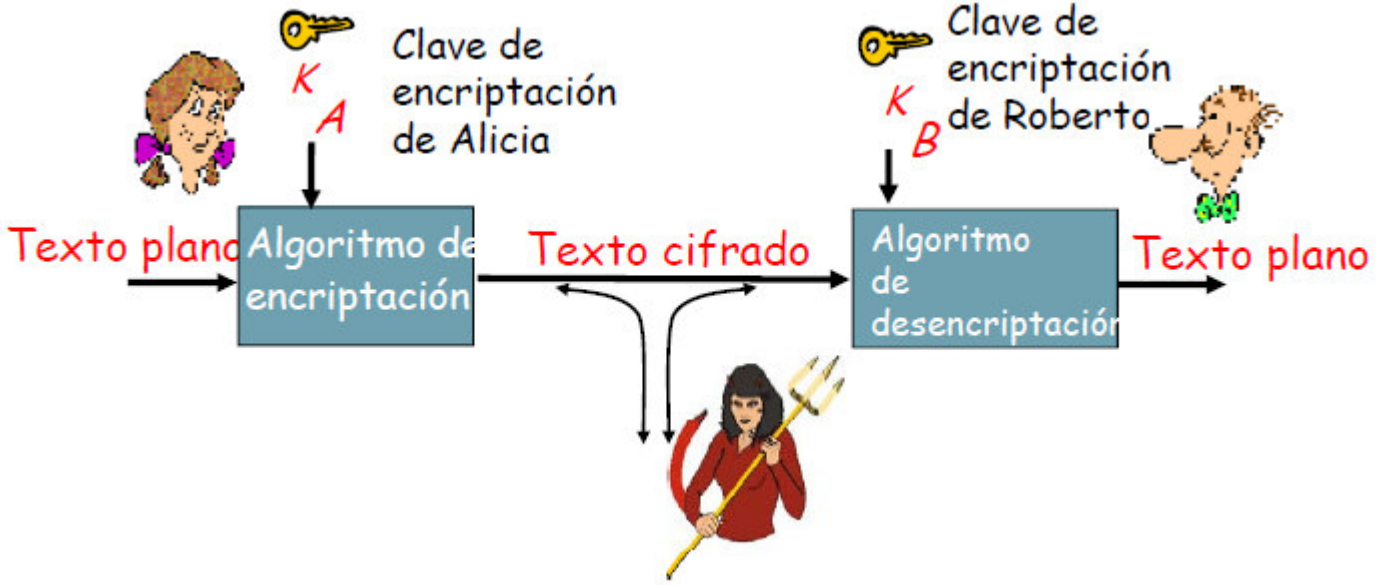
Principios de la Criptografía

- Las **técnicas criptográficas** permiten que un emisor "***disfrace***" los datos de forma disfrace que un intruso no pueda obtener información de los datos interceptados.
- El receptor debe ser capaz de recuperar los datos originales a partir de los datos "disfrazados".



Principios de la Criptografía

- Criptografía de **clave simétrica**: **claves emisor y receptor idénticas y secretas**
- Criptografía de **clave pública**: **encriptación de clave pública, desencriptación de clave secreta privada**.



Principios de la Criptografía

- Criptosistemas **simétricos**:
  - Existirá **una única clave** (secreta) **que deben compartir emisor y receptor**. Con la misma **clave se cifra y se descifra** por lo que la seguridad reside en mantener dicha clave en secreto.

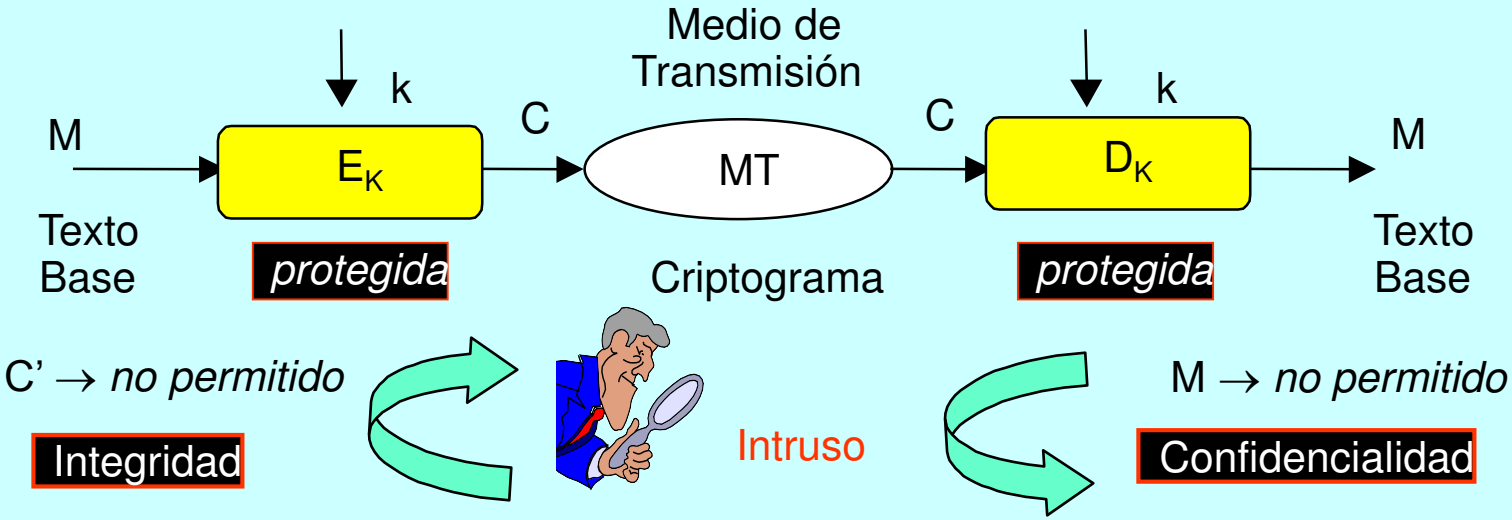
- Criptosistemas **asimétricos**:
  - Cada usuario crea un **par de claves**, **una privada y otra pública**, **inversas** dentro de un cuerpo finito. **Lo que se cifra en emisión con una clave, se descifra en recepción con la clave inversa**. La seguridad del sistema reside en **la dificultad computacional de descubrir la clave privada a partir de la pública**. Para ello, usan funciones matemáticas de un solo sentido o con trampa.

Principios de la Criptografía

- Clasificación de los criptosistemas
- Según el tratamiento del mensaje se dividen en:
- Cifrado en bloque (IDEA, AES, RSA ...) 64 ó 128 bits
- Cifrado en flujo (A5, RC4, SEAL ...) cifrado bit a bit

Principios de la Criptografía
Criptosistemas simétricos

Cifrado con criptosistemas de clave secreta

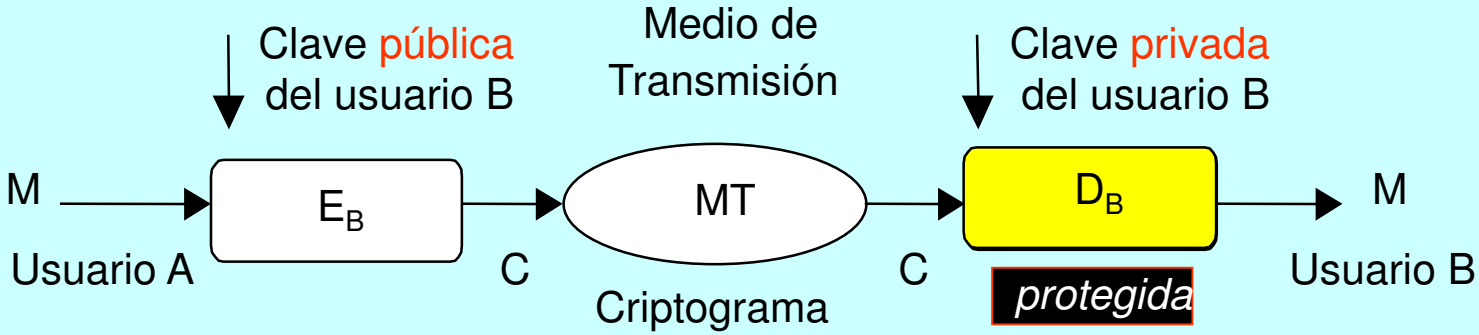


La confidencialidad y la integridad se lograrán si se protegen las claves en el cifrado y en el descifrado. Es decir, se obtienen simultáneamente si se protege la clave secreta.

DES, TDES, IDEA, CAST, RC5, AES, ...

Principios de la Criptografía
Criptosistemas asimétricos (parte 1)

Cifrado con clave pública del receptor (intercambio de claves RSA)



Intruso

M → *no permitido*

Confidencialidad

Observe que se cifra con la clave pública E_B del destinatario B.

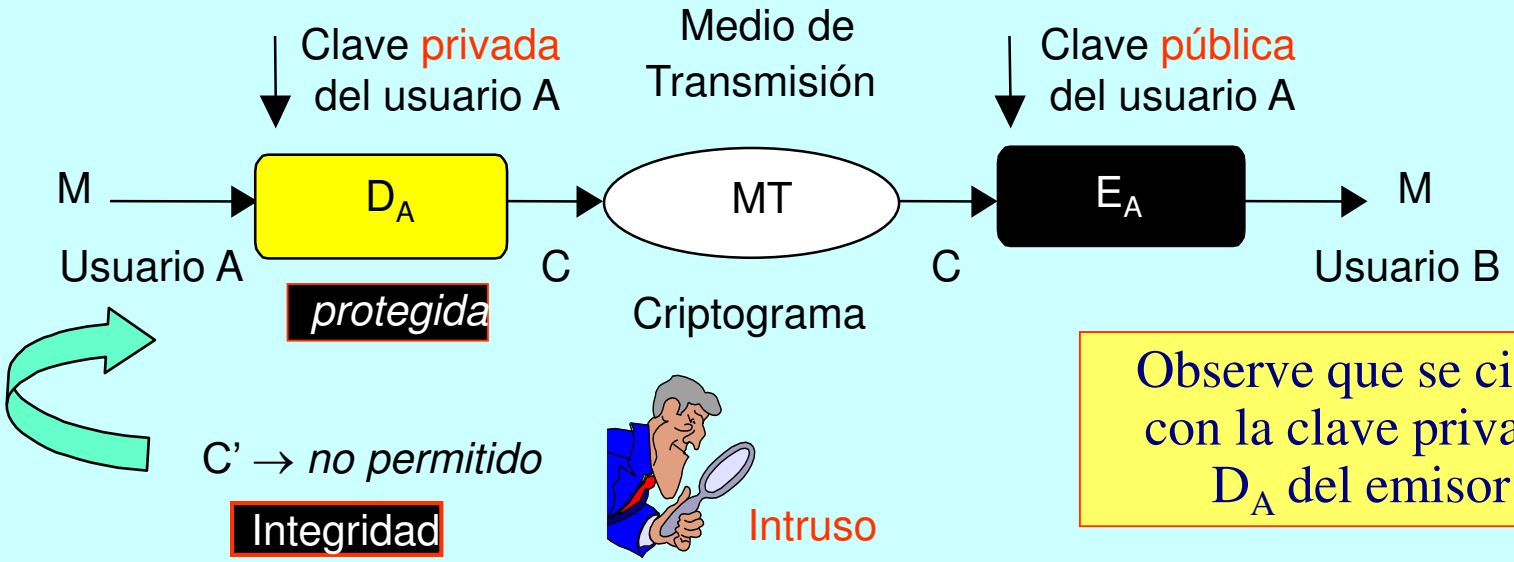
Las cifras E_B y D_B (claves) son inversas dentro de un cuerpo

Un sistema similar es el intercambio de clave de Diffie y Hellman (DH)

Principios de la Criptografía
Criptosistemas asimétricos (parte 2)

Firmas: RSA y DSS

Cifrado con clave privada del emisor (firma digital RSA)



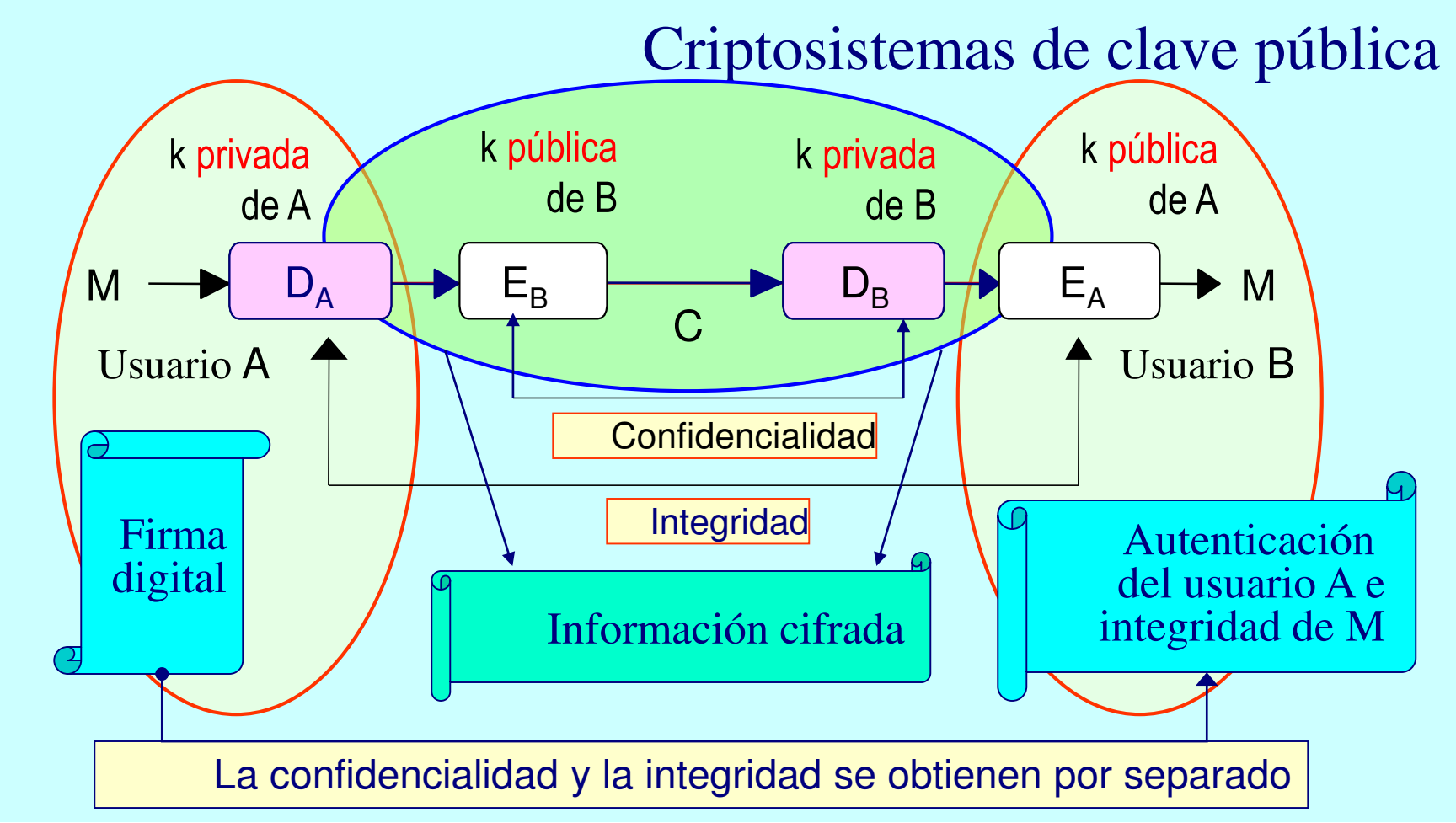
Observe que se cifra con la clave privada D_A del emisor A.

Se firma sobre un hash h(M) del mensaje, por ejemplo SHA-1.

Las cifras D_A y E_A (claves) son inversas dentro de un cuerpo

La firma DSS estará basada en el algoritmo de cifra de ElGamal.

Principios de la Criptografía



Principios de la Criptografía

¿Qué usar, cifra simétrica o asimétrica?

Los sistemas de **clave pública son muy lentos** pero tienen un **fácil intercambio de clave y cuentan con firma digital.**

Los sistemas de clave secreta son muy rápidos pero carecen de lo anterior.

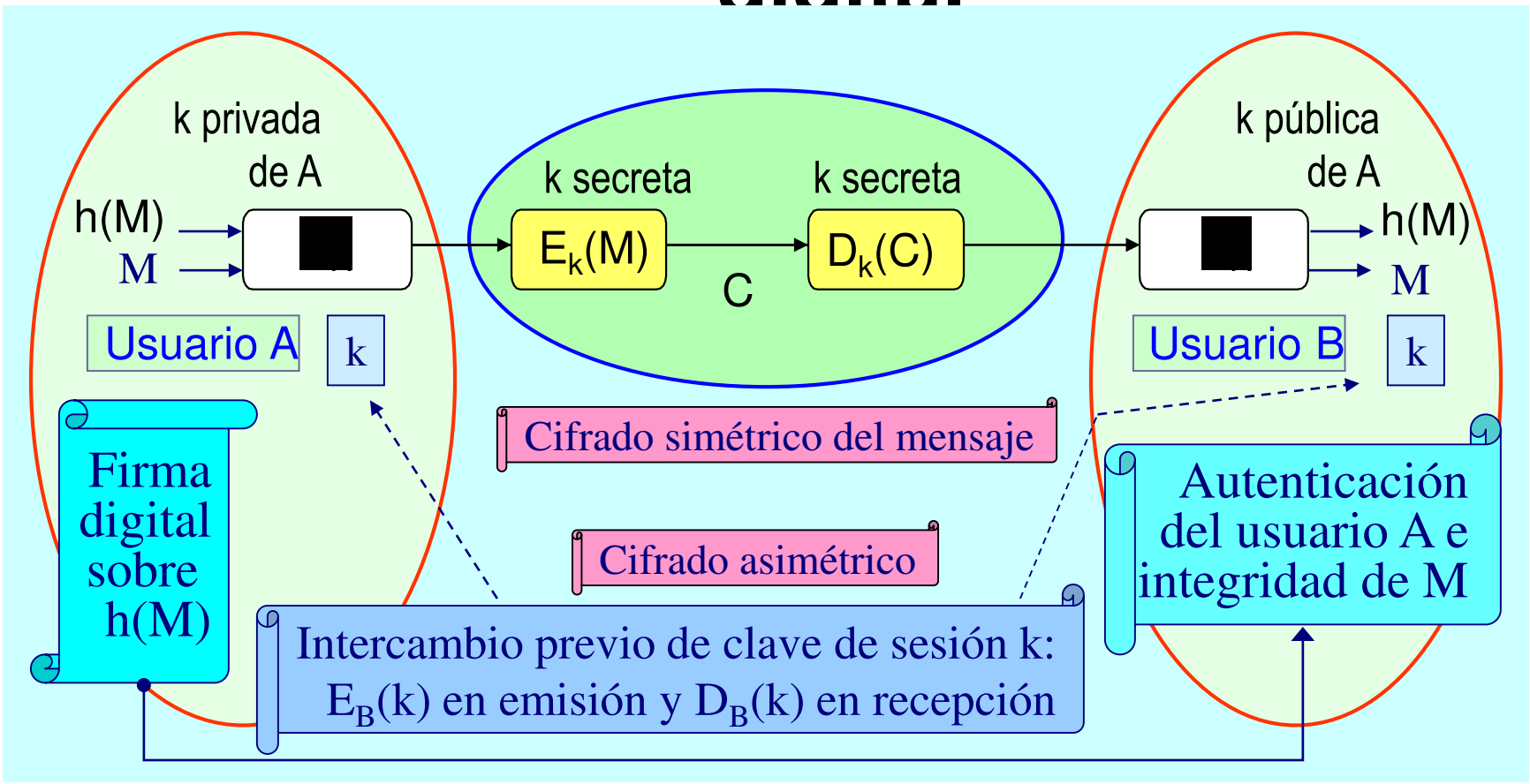
¿Qué hacer?

Cifrado de la información:
- Usaremos sistemas de clave secreta

Firma e intercambio de clave de sesión:
- Usaremos sistemas de clave pública

Principios de la Criptografía

Sistema híbrido de cifra y firma digital



Criptoanálisis

- El **criptoanálisis es el proceso por el que se intenta descubrir un texto plano o una clave de cifrado.**
- Tipos de ataques criptoanáliticos (según la cantidad de información que posee el criptoanalista):
  - **Sólo texto cifrado**: más difícil.
  - **Se conoce el algoritmo de cifrado** y se aplica el ataque de fuerza bruta, probando todas las claves posibles
  - **Texto plano conocido**: se conoce el algoritmo de cifrado y uno o más pares de texto plano a texto cifrado formados con la clave secreta.
  - **Texto plano elegido**: se conoce al algoritmo de cifrado y un mensaje de texto plano elegido con su correspondiente texto cifrado generado con la clave secreta.

Criptoanálisis

- Un algoritmo de cifrado **es computacionalmente seguro** si
  - **El coste de romper el cifrado excede el valor de la información cifrada**;
  - **y/o el tiempo de romper el cifrado excede del tiempo de vida útil de la información**.

Sistemas de Cifrado Clásicos

- La **clasificación** actual de los sistemas de cifra se basa en el tratamiento de la información (**cifrado en bloque vs cifrado en flujo**) o bien en el **tipo de clave utilizada** en la cifra (**sistemas de clave secreta v/s sistemas de clave pública**), pero según su relación con la historia de la criptografía podríamos clasificarlos como:
  - **Sistemas de Cifra Clásicos versus Sistemas de Cifra Modernos**

Sistemas de Cifrado Clásicos

- Esta no es ni mucho menos la mejor clasificación desde el punto de vista de la ingeniería y la informática ... pero **permitirá comprobar el desarrollo de estas técnicas de cifra**, hoy en día rudimentarias y simples, desde una perspectiva histórica y culturalmente interesante para un ingeniero. Además, nos permitirá criptoanalizar con cierta facilidad prácticamente todos estos sistemas y comprobar también las teorías de Shannon sobre las estadísticas del lenguaje.

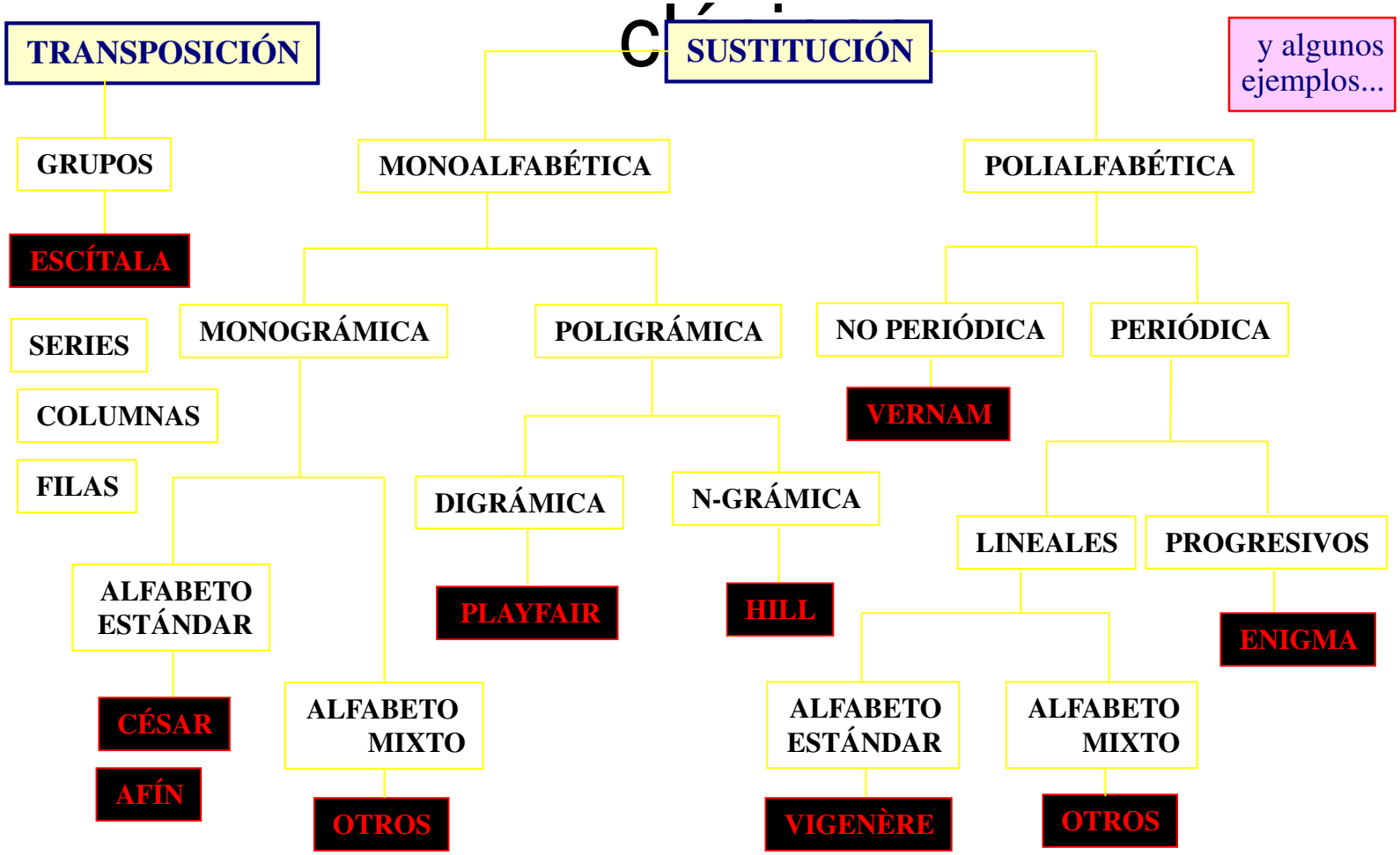
Sistemas de Cifrado Clásicos

Una primera aproximación histórica

- El uso **de técnicas criptográficas es casi tan antiguo** como las culturas de los primeros pueblos de nuestro planeta.
- Ya en el siglo V antes de J.C. un pueblo griego usaba técnicas elementales de cifra para proteger su información.
- Se pretendía garantizar en aquellos días sólo **la confidencialidad y la autenticidad** de los mensajes. A finales del siglo XX se han añadido **la disponibilidad y, últimamente, el no repudio**.
- Los mayores avances se logran en la **Primera y Segunda Guerra Mundiales**, especialmente durante y después de esta última. Los países en conflicto poseían verdaderas empresas con un gran número de matemáticos, cuya función era romper los mensajes cifrados de los teletipos intercambiados por sus enemigos.

Sistemas de Cifrado Clásicos

Clasificación de los criptosistemas clásicos



Sistemas de Cifrado Clásicos

Hitos históricos en la criptografía

- La criptografía clásica abarca desde tiempos inmemoriales hasta la mitad del siglo XX.
- El punto de inflexión en esta clasificación la marcan tres hechos relevantes:
  - En el año 1948 se publica el estudio de Claude Shannon sobre la Teoría de la Información.
  - En 1974 aparece el estándar de cifra DES.
  - Y en el año 1976 se publica el estudio realizado por Whitfield Diffie y Martin Hellman sobre la aplicación de funciones matemáticas de un solo sentido a un modelo de cifra, denominado cifrado con clave pública.

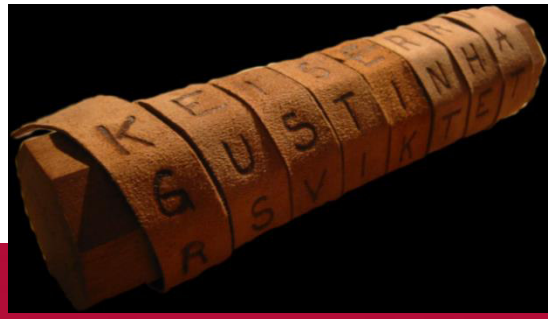
CIFRADO DIGITAL

http://www.cs.jhu.edu/~rubin/courses/sp03/papers/diffie.hellman.pdf

Sistemas de Cifrado Clásicos

Primer cifrador por transposición: escítala

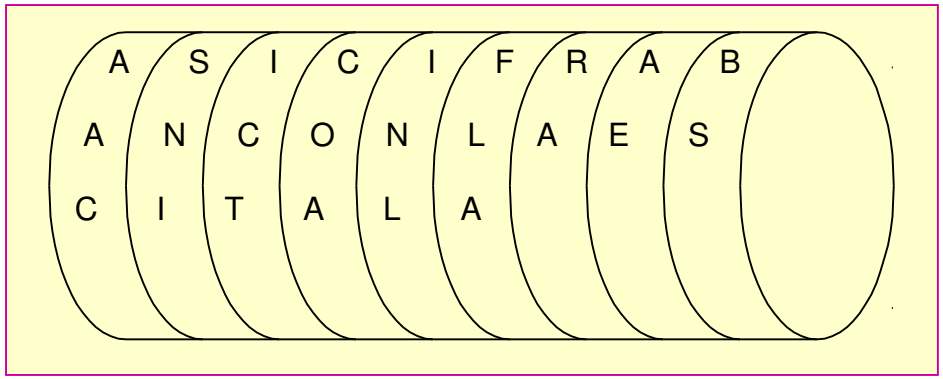
- La escítala era usada en el siglo V a.d.C. por el pueblo griego de los lacedemonios. Consistía en un bastón en el que se enrollaba una cinta de cuero y luego se escribía en ella el mensaje de forma longitudinal.
- Al desenrollar la cinta, las letras aparecerán desordenadas.
- Para descifrar el criptograma y recuperar el mensaje en claro habrá que enrollar dicha cinta en un bastón con el mismo diámetro que el usado en el extremo emisor y leer el mensaje de forma longitudinal. La clave del sistema se encuentra en el **diámetro** del bastón. **Se trata de una cifra por transposición pues los caracteres del criptograma son los mismos que en el texto en claro pero están distribuidos de otra forma dentro del criptograma**.



Sistemas de Cifrado Clásicos

Método de cifra de la escítala

Bastón y cinta para cifrar



En ese bastón residía la fortaleza de un pueblo.

Por ello, y como símbolo de poder, el bastón de mando que se le entrega al alcalde de una ciudad en la ceremonia de su nombramiento, proviene de estos tiempos tan remotos.

El texto en claro es:

M = ASI CIFRABAN CON LA ESCITALA

El texto cifrado o criptograma será:

C = AAC SNI ICT COA INL FLA RA AE BS

Sistemas de Cifrado Clásicos

Primer cifrador por sustitución: **Polybios**

- Es el cifrador por sustitución de caracteres más antiguo que se conoce (siglo II a.d.C.) pero como duplica el tamaño del texto en claro, con letras o números, ... no fue tan buena la idea.

	A	B	C	D	E
A	*A*	*B*	*C*	*D*	*E*
B	*F*	*G*	*H*	*IJ*	*K*
C	*L*	*M*	*N*	*O*	*P*
D	*Q*	*R*	*S*	*T*	*U*
E	*V*	*W*	*X*	*Y*	*Z*

	1	2	3	4	5
1	*A*	*B*	*C*	*D*	*E*
2	*F*	*G*	*H*	*IJ*	*K*
3	*L*	*M*	*N*	*O*	*P*
4	*Q*	*R*	*S*	*T*	*U*
5	*V*	*W*	*X*	*Y*	*Z*

M_1 = QUÉ BUENA IDEA

C_1 = DA DE AE AB DE AE

CC AA BD AD AE EA

M_2 = LA DEL GRIEGO

C_2 = 31 11 14 15 31 22

42 24 15 22 34

Sistemas de Cifrado Clásicos

El cifrador del César

- En el siglo I a.d.C., Julio César usaba este cifrador. El algoritmo consiste en el **desplazamiento de tres espacios hacia la derecha** de los caracteres del texto en claro. Es un cifrador por sustitución monoalfabético en el que las operaciones se realizan módulo n, siendo n el número de elementos del alfabeto (en aquel entonces el latín).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
M_i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
C_i	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Alfabeto de cifrado del César para castellano mod 27

Sistemas de Cifrado Clásicos

Ejemplo de cifra del César en mod 27

M_i	A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z
C_i	D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z A B C

Cifrado: $C_i = M_i + 3 \bmod 27$ Descifrado: $\mathbf{M_i = C_i - 3 \bmod 27}$

M = EL PATIO DE MI CASA ES PARTICULAR

C = HÑ SDWLR GH OL FDVD HV SDUWLFXÑDU

Cada letra se **cifrará siempre igual. Es una gran debilidad** y hace que este sistema sea muy vulnerable y fácil de atacar, simplemente usando las estadísticas del lenguaje.

Sistemas de Cifrado Clásicos

Criptoanálisis del cifrador por sustitución

- **La letra más frecuente del criptograma la hacemos coincidir con la más frecuente del lenguaje**, la letra E, y encontramos así b.

C = LZAHL ZBTHW YBLIH XBLKL ILYOH ZLYCH ROKH

Frecuencias observadas en el criptograma: L (7); H (6); Z (3); B (3); Y (3); I (2); K (2); O (2); A (1); T (1); W (1); X (1); C (1); R (1).

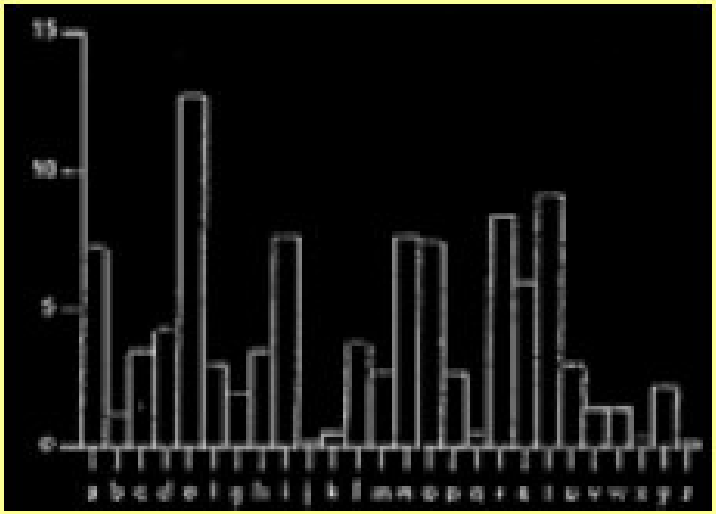
- Es posible que la letra E del lenguaje se cifre como L. Comprobamos además si la letra A (segunda más frecuente) se cifra como H:

$E + b \bmod 27 = L \Rightarrow b = L - E \bmod 27 = 11 - 4 \bmod 27 = 7$ 👌

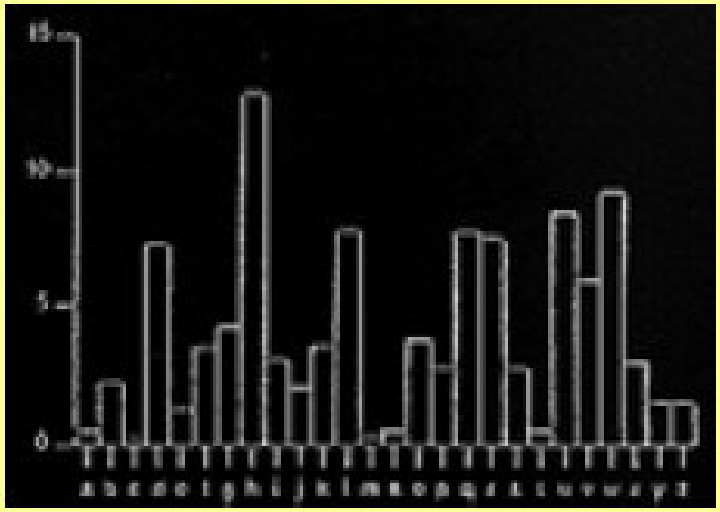
$A + b \bmod 27 = H \Rightarrow b = H - A \bmod 27 = 7 - 0 \bmod 27 = 7$ 👍

M = ESTA ES UNA PRUEBA QUE DEBERIA SER VALIDA

Sistemas de Cifrado Clásicos



Frecuencias en Inglés



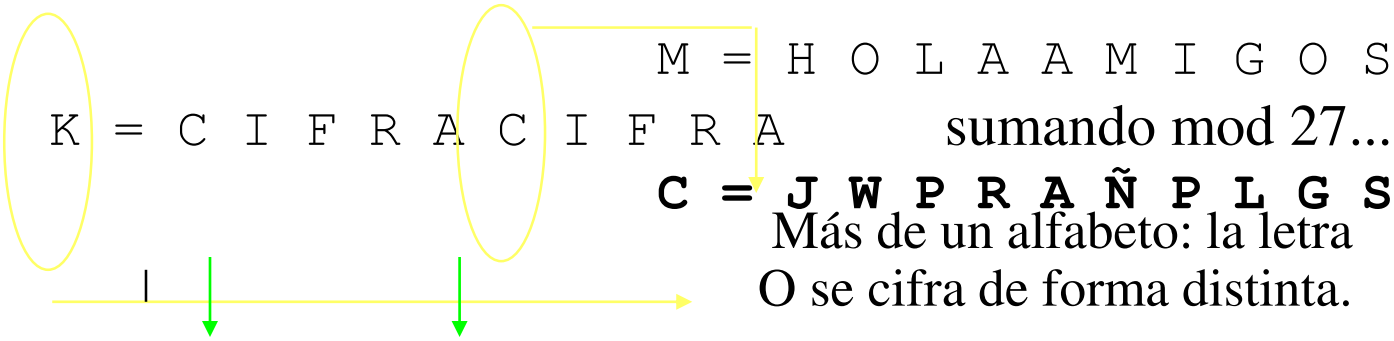
Frecuencias tras César

Sistemas de Cifrado Clásicos
El cifrador de Vigenère

- Este cifrador polialfabético **soluciona la debilidad del cifrado del César en que una letra se cifra siempre igual**. Se usa una clave K de longitud L y se cifra carácter a carácter sumando módulo n el texto en claro con los elementos de esta clave.

$$C_i = M_i + K_i \bmod 27$$

Sea K = CIFRA y el mensaje M = HOLA AMIGOS



Observe que el criptograma P se obtiene de un texto L y de un texto I.

Sistemas de Cifrado Clásicos
¿Es Vigenère un algoritmo seguro?

- Si la clave de Vigenère **tiene mas de 6 caracteres distintos**, se logra una distribución de frecuencias en el criptograma del tipo normal, es decir más o menos plana, por lo que se logra difuminar la redundancia del lenguaje.
- Aunque pudiera parecer que usando una clave larga y de muchos caracteres distintos, y por tanto varios alfabetos de cifrado, **Vigenère es un sistema de cifra seguro, esto es falso.**
- **La redundancia del lenguaje unido a técnicas de criptoanálisis muy sencillas, como los métodos de Kasiski y del Índice de Coincidencia**, permiten romper la cifra y la clave de una manera muy fácil y con mínimos recursos. En la siguiente diapositiva veremos un ataque por el método de **Kasiski**.

Sistemas de Cifrado Clásicos

Ataque por el método de Kasiski

- El método de Kasiski consiste **en buscar repeticiones de cadenas de caracteres en el criptograma**. Si estas **cadenas son mayores o iguales a tres caracteres y se repiten más de una vez**, lo más probable es que esto se deba a cadenas típicas del texto en claro (trigramas, tetragramas, etc., muy comunes) que se han cifrado con una misma porción de la clave.
- Si se detectan estas cadenas, **la distancia entre las mismas será múltiplo de la longitud de la clave**. Luego, el máximo común divisor entre esas cadenas es un candidato a ser la longitud de la clave, digamos L.
- Dividimos el criptograma en L subcriptogramas que entonces han sido cifrados por una misma letra de la clave y en cada subcriptograma hacemos un ataque simple ahora de tipo estadístico monoalfabético.
- La idea es buscar ahora a través de los tres caracteres más frecuentes en cada subcriptograma las posiciones relativas de las letras A, E y O que en castellano están separadas por 4 y 11 espacios. La letra de la posición que ocupe la letra A (A = 0) será entonces la letra clave correspondiente.

Sistemas de Cifrado Clásicos

Cadenas repetidas en ataque de Kasiski

Sea el criptograma C de 404 caracteres que vamos a criptoanalizar el siguiente:

```
PBVRQ VICAD SKAÑS DETSJ PSIED BGGMP SLRPW RÑPWY EDSDE ÑDRDP CRCPQ MNPWK
UBZVS FNVRD MTIPW UEQVV CBOVN UEDIF QLONM WNUVR SEIKA ZYEAC EYEDS ETFPH
LBHGU ÑESOM EHLBX VAEEP UÑELI SEVEF WHUNM CLPQP MBRRN BPVIÑ MTIBV VEÑID
ANSJA MTJOK MDODS ELPWI UFOZM QMVNF OHASE SRJWR SFQCO TWVMB JGRPW VSUEX
INQRS JEUEM GGRBD GNNIL AGSJI DSVSU EEINT GRUEE TFGGM PORDF OGTSS TOSEQ
OÑTGR RYVLP WJIFW XOTGG RPQRR JSKET XRNBL ZETGG NEMUO TXJAT ORVJH RSFHV
NUEJI BCHAS EHEUE UOTIE FFGYA TGGMP IKTBW UEÑEN IEEU.
```

Entre otras, se observan las siguientes cadenas (subrayadas) en el criptograma:

- 3 cadenas GGMP, separadas por 256 y 104 posiciones.
- 2 cadenas YEDS, separadas por 72 espacios.
- 2 cadenas HASE, separadas por 156 espacios.
- 2 cadenas VSUE, separadas por 32 espacios.

Luego el período de la clave puede ser mcd (256, 104, 72, 156, 32) = 4. La clave tendrá cuatro caracteres, por lo tanto tomaremos del criptograma el carácter 1º, el 5º, el 9º, etc. para formar el primer subcriptograma C_A; luego el 2º, el 6º, el 10º, etc. para formar el subcriptograma C_B, y lo mismo para subcriptogramas C_C y C_D.

Criptool: https://www.cryptool.org/en/

- **Ejercicio 1:**
  - Cifra un mensaje cualquiera usando el método César y pásaselo a tu compañero.
  - Coge el mensaje cifrado de tu compañero e intenta descifrarlo usando el método de frecuencias
- **Ejercicio 2:**
  - Cifra un mensaje cualquier usando el método Vigenère y pásaselo a tu compañero
  - Intentar romperlo usando el métido Kasiski
  - Descífralo con la clave que te ha pasado él

Bibliografía:

- http://www.criptored.upm.es/crypt4you/temas/criptografiaclasica/leccion1.html

Criptografía moderna

Conceptos elementales

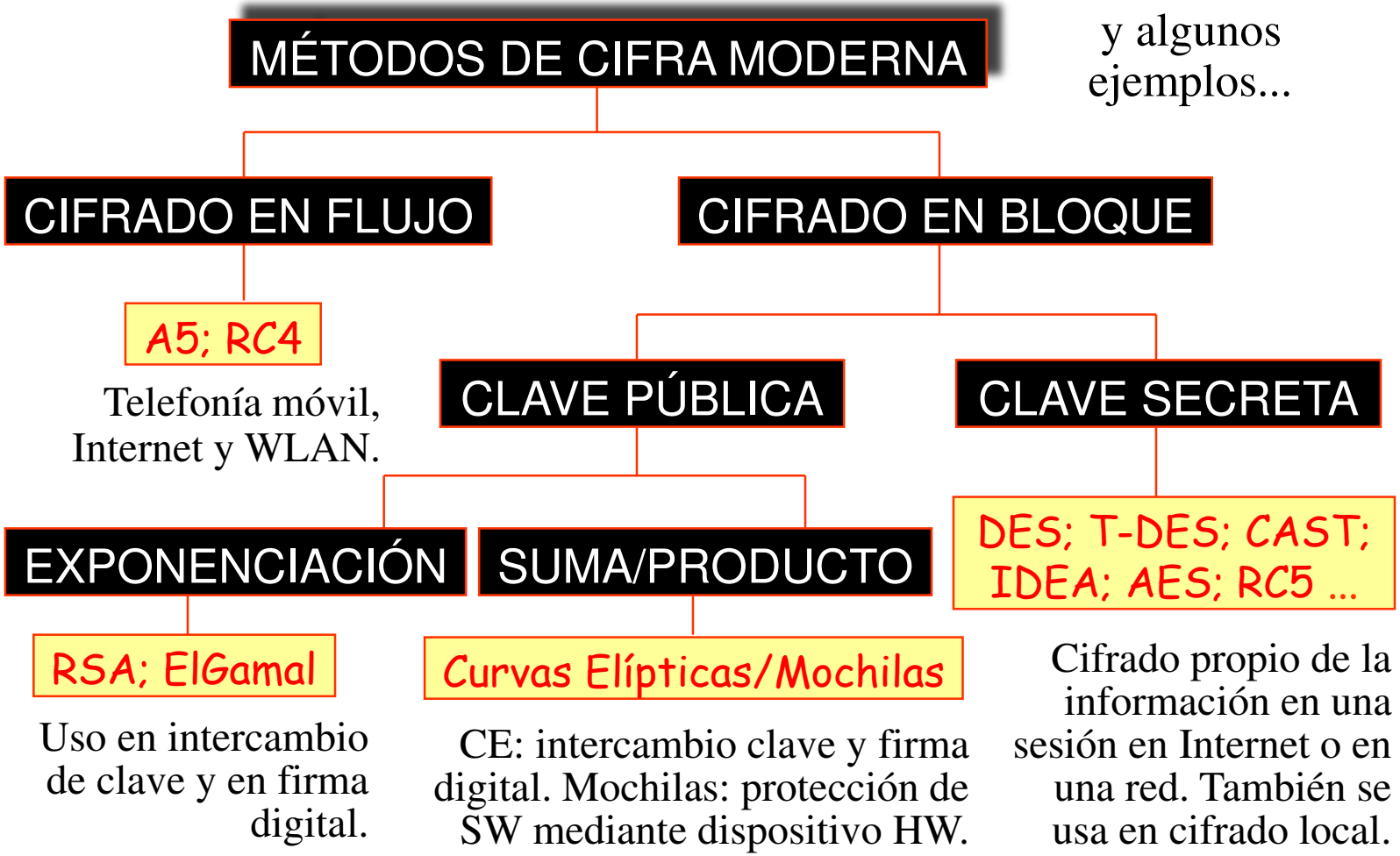
Un par de ideas básicas

- **Los criptosistemas <u>modernos</u>, cuya cifra en bits está orientada a todos los caracteres ASCII o ANSI**, usan por lo general una operación algebraica en Z_n, un cuerpo finito, **sin que necesariamente este módulo deba corresponder con el número de elementos del alfabeto o código utilizado**. Es más, **<u>nunca</u>** coinciden: **siempre será mucho mayor el cuerpo de trabajo que el alfabeto usado**.
- **Su <u>fortaleza</u>** se debe basar en **la imposibilidad computacional de descubrir una clave secreta única**, en tanto que **el algoritmo de cifra es** (**o al menos debería serlo**) **público**.
- En la siguiente dirección web, encontrará un amplio compendio de sistemas de cifra y criptografía.

http://en.wikipedia.org/wiki/Category:Cryptography

Comparativa

Clasificación de los criptosistemas



Comparativa

Técnica de cifra en flujo

- El mensaje en claro se **leerá bit a bit**.
- Se realizará **una operación de cifra, normalmente la función XOR,** con una secuencia cifrante de bits Si que debe cumplir ciertas condiciones:
  - Tener un período muy alto (ya no infinito)
  - Tener propiedades pseudoaleatorias (ya no aleatorias)
  - Ejemplo: A5 en sus diferentes versiones (A5/1, A5/2),…)

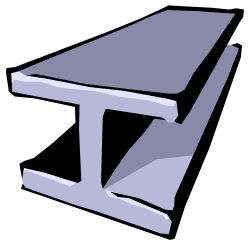
Comparativa

Técnica de cifra en flujo

- RC4
- Se usa en algunos de los protocolos más populares como **TLS/SSL y Wired Equivalent Privacy (WEP)**
- Algunos modos de usar el algoritmo de criptografía RC4 lo han llevado a ser un sistema de criptografía **muy inseguro, incluyendo su uso WEP**
- Entre los factores principales que han ayudado a que RC4 esté en un rango tan amplio de aplicaciones son su **increíble velocidad y simplicidad**. La implementación tanto en software como en hardware es muy sencilla de desarrollar
- RC4 es un un algoritmo sorprendentemente simple. Este consiste en 2 algoritmos: 1-Key Scheduling Algorithm (KSA) y 2- Pseudo-Random Generation Algorithm (PRGA).

Comparativa

Introducción a la cifra en bloque



El mensaje **se agrupa en bloques**, por lo general de 8 ó 16 bytes (64 ó 128 bits) antes de aplicar el algoritmo **de cifra a cada bloque de forma independiente con la misma clave**.

Cifrado con Clave Secreta

Hay algunos algoritmos muy conocidos por su uso en aplicaciones bancarias (**DES**), correo electrónico (IDEA, CAST), comercio electrónico (Triple DES) y el nuevo estándar (**AES Rijndael**).

Comparativa

¿Qué tamaño de bloque usar?

Si el bloque fuese muy pequeño, por ejemplo uno o dos bytes, **esto facilitaría un ataque por estadísticas del lenguaje**. Se trataría de un cifrado por monogramas o diagramas muy débil.

Pero si el bloque fuese muy grande, por ejemplo cientos de bytes, **el sistema sería lento en el tratamiento del texto en claro y no sería bueno su rendimiento.**

Los valores indicados **de 64 y 128 bits** son un término medio que satisface ambas condicionantes: es la típica situación de compromiso que tanto vemos en ingeniería.

Comparativa

Tres debilidades en la cifra simétrica

a) **Mala gestión de claves.** Crece el número de claves secretas en una proporción igual a n^2 para un valor n grande de usuarios lo que imposibilita usarlo 👎.

b) **Mala distribución de claves.** No existe posibilidad de enviar, de forma segura y eficiente, una clave a través de un medio o canal inseguro 👎.

c) **No tiene firma digital.** Aunque sí será posible autenticar el mensaje mediante una marca, no es posible firmar digitalmente el mensaje, al menos en un sentido amplio y sencillo 👎.

Comparativa

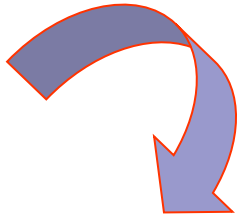
¿Por qué usamos entonces clave secreta?

a) Mala gestión de claves 👎

b) Mala distribución de claves 👎

c) No permite firma digital 👎

¿Tiene algo de bueno la cifra en bloque con clave secreta?



Sí: la velocidad de cifra es muy alta 👍 y por ello se usará para realizar la función de cifra de la información. **Además, con claves de sólo unas centenas de bits obtendremos una alta seguridad pues la no linealidad del algoritmo hace que en la práctica el único ataque factible sea por fuerza bruta.**

Comparativa

Cifrado asimétrico

- Comienza a ser ampliamente conocido a través de su aplicación en los sistemas de correo electrónico seguro (**PGP y PEM**) **permitiendo cifrar e incluir una firma digital adjunta al documento** o e-mail enviado y también en los navegadores Web.
- **Cada usuario tendrá dos claves**, una secreta o privada y otra pública, inversas entre sí dentro de un cuerpo.
- **Usan las funciones unidireccionales con trampa**.

Comparativa

Funciones unidireccionales con trampa

- **Son funciones matemáticas de un solo sentido** (one-way functions) y que **nos permiten usar la función en sentido directo o de cálculo fácil para cifrar y descifrar** (usuarios legítimos**) y fuerza el sentido inverso o de cálculo difícil para aquellos impostores**, hackers, etc. que lo que desean es atacar o criptoanalizar la cifra.

$f(M) = C$ es *siempre fácil.*

$f^{-1}(C) = M$ e*s difícil salvo que se tenga la trampa.*

Comparativa

Funciones con trampa más usadas

Problema de la factorización

Cálculo directo: producto de dos primos grandes $p*q = n$
Cálculo inverso: factorización de número grande $n = p*q$

Problema del logaritmo discreto

Cálculo directo: exponenciación discreta $\beta = \alpha^x \bmod n$
Cálculo inverso: logaritmo discreto $x = \log_\alpha \beta \bmod n$

Comparativa

Otras funciones con trampa

Problema de la mochila

Cálculo directo: sumar elementos de mochila con trampa
Cálculo inverso: sumar elementos de mochila sin trampa

Problema de la raíz discreta

Cálculo directo: cuadrado discreto $x = a*a \bmod n$
Cálculo inverso: raíz cuadrada discreta $a = \sqrt{x} \bmod n$

Comparativa

Uso de la criptografía asimétrica

- Estas dos operaciones de cifra son posibles debido a la característica intrínseca de los sistemas de clave pública: el uso de una clave privada (secreta) inversa de una pública.

¿Qué aplicación tendrán entonces los sistemas de criptografía de clave pública o asimétrica?

- Usando la **clave pública del destino** se hará el **intercambio de claves de sesión de una cifra con sistemas simétricos** (decenas a centenas de bits). Sistema Híbrido
- Usando la **clave privada de origen**, se firmará digitalmente un resumen (centenas de bits) del mensaje obtenido con una función hash.
- Observe que se hace hincapié en las “centenas de bits” dado que estos sistemas son muy lentos comparados con los simétricos.

Comparativa

Comparativa: la gestión de claves

Gestión de claves	
<u>Clave Secreta</u>	<u>Clave Pública</u>
Hay que memorizar un número muy alto de claves: $\rightarrow n^2$.	Sólo es necesario memorizar la clave privada del emisor.

En cuanto a la gestión de claves, serán mucho más eficientes los sistemas de cifra asimétricos pues los simétricos no permiten una gestión lógica y eficiente de estas claves: en los asimétricos sólo es necesario memorizar la frase o palabra de paso para acceder a la clave privada.

Comparativa

Comparativa: el espacio de claves

Longitud y espacio de claves

Clave Secreta

Debido al tipo de cifrador usado, la clave será del orden de centenas de bits.

≥ 128

Clave Pública

Por el algoritmo usado en la cifra, la clave será del orden de miles de bits.

≥ 1.024

En cuanto al espacio de claves, no son comparables los sistemas simétricos con los asimétricos. Para atacar un sistema asimétrico no se buscará en todo el espacio de claves como debería hacerse en los sistemas simétricos.

Comparativa

Comparativa: la vida de las claves

Vida de una clave	
<u>Clave Secreta</u>	<u>Clave Pública</u>
La duración es muy corta pues casi siempre se usa como clave de una sesión.	La duración de la clave pública, que la entrega y gestiona un tercero, suele ser larga.
Segundos o minutos	Meses o un año

En cuanto a la vida de una clave, en los sistemas simétricos ésta es muchísimo menor que las usadas en los asimétricos. La clave de sesión es aleatoria, en cambio la asimétrica es propia del usuario.

Comparativa

Comparativa: la autenticación de emisor

Autenticación	
Clave Secreta	Clave Pública
Se puede autenticar el mensaje pero no al emisor de forma sencilla y eficiente.	Al haber una clave pública y otra privada, se podrá autenticar el mensaje y al emisor.

En cuanto a la autenticación, los sistemas simétricos tienen una autenticación más pesada y con una tercera parte de confianza. Los asimétricos permiten una firma digital verdadera, eficiente y sencilla, en donde la tercera parte de confianza es sólo presencial.

Comparativa

Comparativa: la velocidad de cifra

Velocidad de cifra	
Clave Secreta	Clave Pública
La velocidad de cifra es muy alta. Es el algoritmo de cifra del mensaje.	La velocidad de cifra es muy baja. Se usa para el intercambio de clave y la firma digital.

Cientos de M Bytes/seg en HW

En cuanto a la velocidad de cifra, los sistemas simétricos son de 100 a 1.000 veces más rápidos que los asimétricos. En SW la velocidad de cifra es más baja.

Cientos de K Bytes/seg en HW

Comparativa

Resumen comparativo de estas cifras

Cifrado Simétrico

- Confidencialidad
- Autenticación parcial
- Sin firma digital
- Claves:
  - Longitud pequeña
  - Vida corta (sesión)
  - Número elevado
- Velocidad alta

Cifrado Asimétrico

- Confidencialidad
- Autenticación total
- Con firma digital
- Claves:
  - Longitud grande
  - Vida larga
  - Número reducido
- Velocidad baja

Comparativa

Seguridad en la cifra simétrica y asimétrica

- La criptografía simétrica o de clave secreta usa una única clave para cifrar en emisión y descifrar en destino.
  - **La seguridad del sistema reside entonces en cuán segura sea dicha clave.**
- En la criptografía asimétrica cada usuario se crea un par de claves llamadas pública y privada, inversas entre sí dentro de un cuerpo finito, de forma que lo que hace una la otra lo deshace. Para cifrar se usa, por ejemplo, la clave pública de destino y para descifrar el destinatario hará uso de su clave privada.
  - **La seguridad del sistema reside ahora en la dificultad computacional de encontrar la clave privada a partir de la clave pública.**

DES

Especificaciones técnicas finales del DES.

- Bloque a cifrar: 64 bits
- Clave: 8 bytes (con paridad, no caracteres ASCII)
- Normas ANSI:
- X3.92: Descripción del algoritmo.
- X3.108: Descripción de los modos de operación (ECB, CBC, OFB).
- Fácil implementación en un circuito integrado.

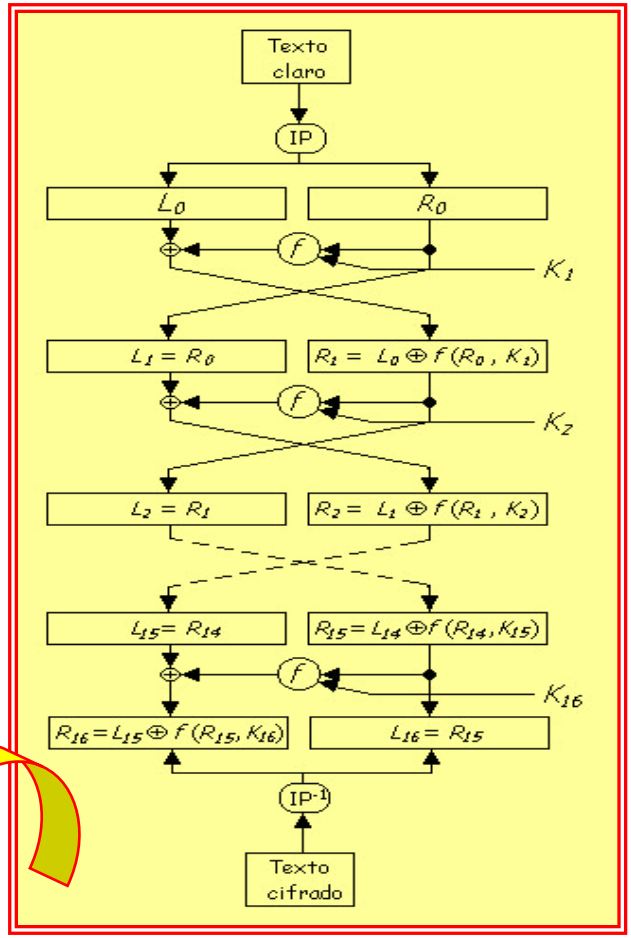
Veremos su descripción y modos de operación. En la página que se indica encontrará las especificaciones del DES.

http://www.itl.nist.gov/fipspubs/fip46-2.htm

DES

Visión general del DES

- Cifrador de bloque
  - Tipo Feistel
- Longitud de clave de 56 bits
  - Realiza 16 vueltas.
- La cifra del bloque central usa técnicas de **sustituciones y permutaciones**.
- Para poder realizar las sumas or exclusivo, usará permutaciones con expansión y compresión para igualar el número de bits.



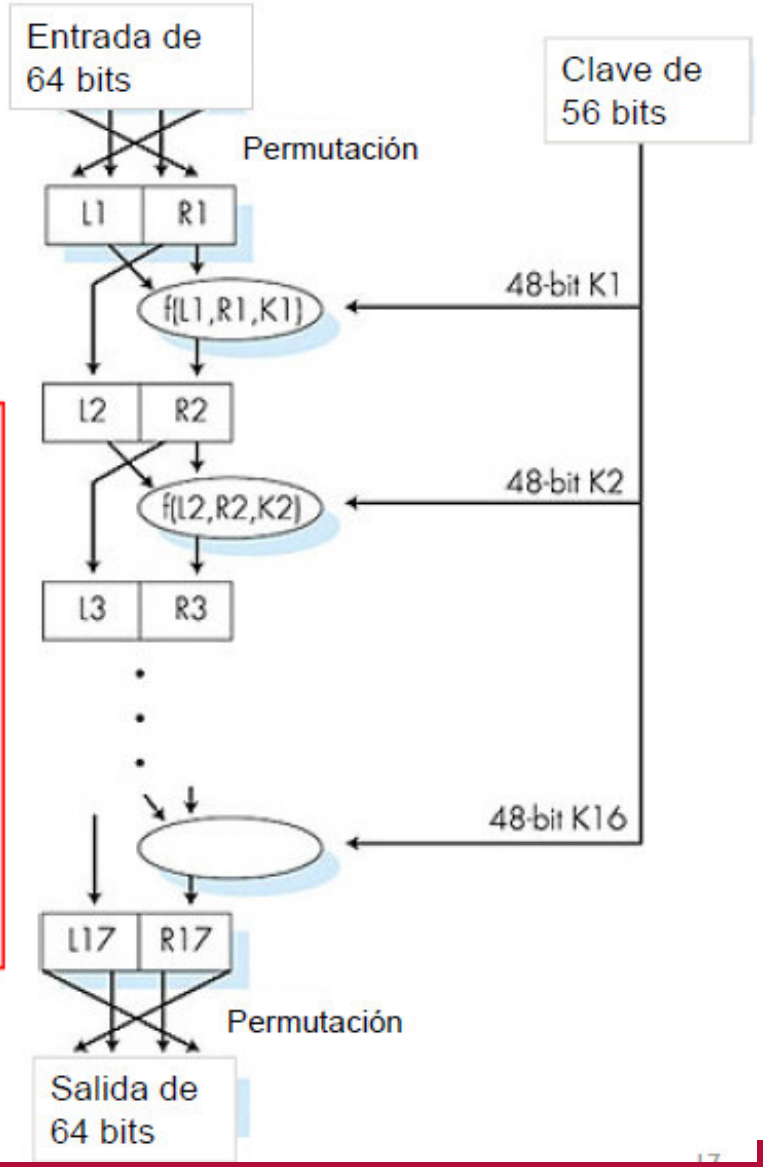
En el descifrado se aplican claves y desplazamientos en sentido inverso

DES

Funcionamiento DES

Permutación inicial y final

16 "rondas" idénticas de aplicación de función, cada una utiliza 48 bits distintos de permutación final de clave.



AES
El nuevo estándar en cifra AES

- AES: Advanced Encryption Standard

- El DES, estándar desde 1976, pasa la certificación de la NBS National Bureaux of Standards en 1987 y en 1993.
- **En 1997 el NIST** National Institute of Standards and Technology (antigua NBS) **no certifica al DES** y **llama a concurso público para un nuevo algoritmo estándar, el AES**.

http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

http://www.iaik.tu-graz.ac.at/research/krypto/AES/

- En **octubre del año 2000 el NIST** elige el algoritmo belga **Rijndael** como nuevo estándar para cifrado del siglo XXI.

http://www.criptored.upm.es/guiateoria/gt_m480a.htm

AES
Características del algoritmo AES

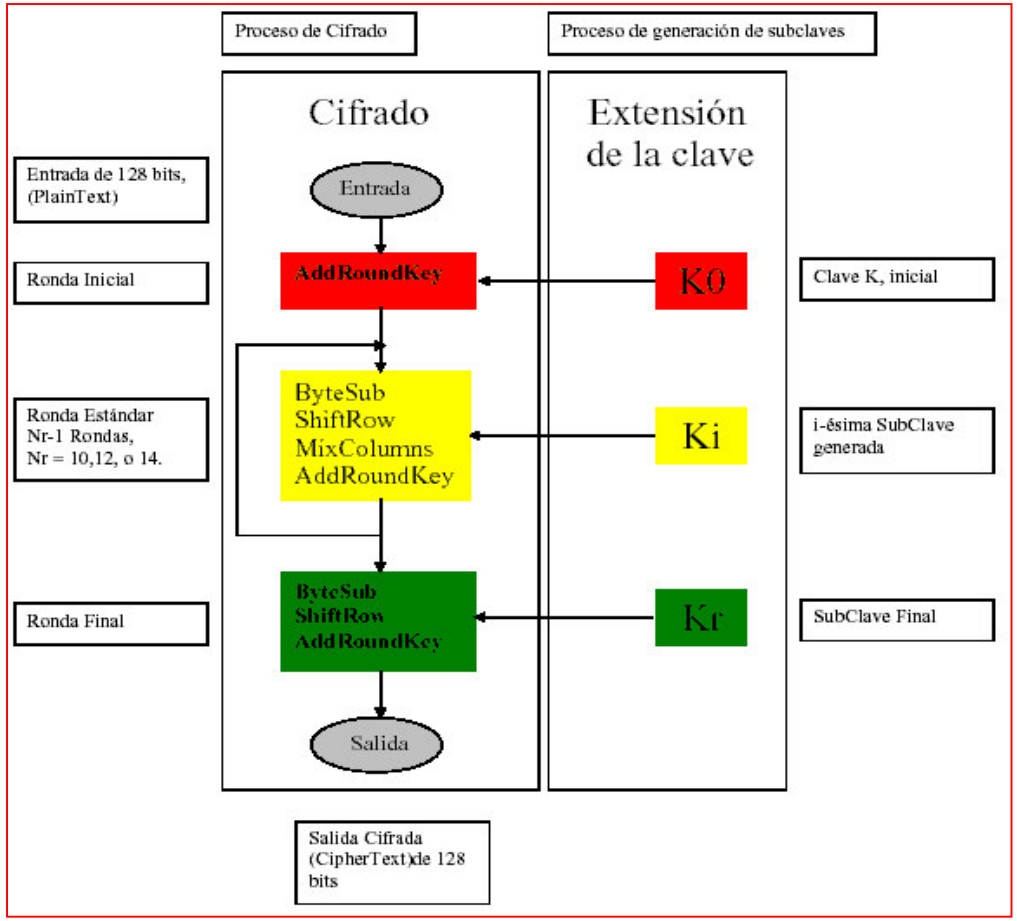
Rijndael: autores Vincent Rijmen & Joan Daemen

- **No es de tipo Feistel**.
- Implementado para trabajar en los procesadores de 8 bits usados en tarjetas inteligentes y en CPUs de 32 bits.
- **Tamaño de clave variable**: 128, 192 y 256 bits (estándar) o bien múltiplo de 4 bytes.
- **Tamaño del bloque de texto**: 128 bits o múltiplo de 4 bytes.
- **Operaciones modulares a nivel de byte** (representación en forma de polinomios) y de palabra de 4 bytes: 32 bits.
- **Número de etapas flexible** según necesidades del usuario.
- Usa un conjunto de Cajas S similares a las del DES.

http://www.iaik.tu-graz.ac.at/research/krypto/AES/old/%7Erijmen/rijndael/

AES

Esquema general del AES



Funciones en cifrado:

- *AddRoundKey*
- *ByteSub*
- *ShiftRow*
- *MixColumns*

Funciones en descifrado:

- *InvAddRoundKey*
- *InvByteSub*
- *InvShiftRow*
- *InvMixColumns*

Se realizará además una expansión de la clave K para generar desde K0 hasta Kr.

Figura y tablas tomadas de: http://www.criptored.upm.es/guiateoria/gt_m117i.htm

Criptool: https://www.cryptool.org/en/

- **Ejercicio 3:**
  - Utiliza la herramienta “procedimientos individuales” -> Visualización de Algoritmos para visualizar cómo funciona el algoritmo DES y responde:
    - En DES, ¿se realiza una permutación previa antes de dividir el bloque en dos mitades?
    - ¿Qué longitud tiene la clave de DES?
    - ¿Tiene bits de paridad? ¿Cuántos?
    - ¿Se permuta también la clave de cifrado?
    - ¿En cada iteración, cual es la longitud de la clave?
    - ¿En que consiste la expansión?¿Por qué se debe hacer?
    - ¿En qué consiste el algoritmo de generación de subclave?. La clave que se aplica en cada iteración, ¿es siempre la misma?
    - En qué consiste el algoritmo Triple-DES

- **Ejercicio 4 (opcional):**
  - Realiza una visualización del algoritmo AES:
  - https://www.nayuki.io/page/aes-cipher-internals-in-excel

    - ¿Qué longitud tiene el bloque de entrada para cifrar en AES?
    - ¿y la clave de cifrado?
    - En AES, ¿Cuáles son los 4 tipos de transformaciones?
    - En el proceso de cifrado, ¿qué se hace en la ronda incial AddRoundKey?
    - ¿En que consiste la ronda subBytes?
    - ¿En qué consiste la ronda ShiftRows?
    - ¿En qué consiste la ronda MixColumns?
    - En AES, cual es la unidad mínima con la que se realizan todas las operaciones
    - ¿Cuantas iteraciones realiza AES?
    - La ronda final tiene alguna particularidad
    - El texto cifrado resultante, que tamaño de bloque tiene
    - En el algoritmo del cálculo de subclaves, ¿qué se perdigue?
    - ¿Qué tres operaciones se realizan en la primera iteración del cálculo de subclaves?
    - En las restantes iteraciones, ¿se hacen las mismas operaciones?

Cifrado Asimétrico: RSA

- Problema de **clave simétrica**: las dos partes deben conocer la clave compartida lo que necesita una **¡comunicación segura!**
- Los sistemas criptográficos de **clave pública** también **son útiles para la autentificación**.

- **En la encriptación de clave pública, el receptor tiene dos claves: una pública conocida por todos, y una privada, conocida sólo por el propio receptor**.

Cifrado Asimétrico: RSA

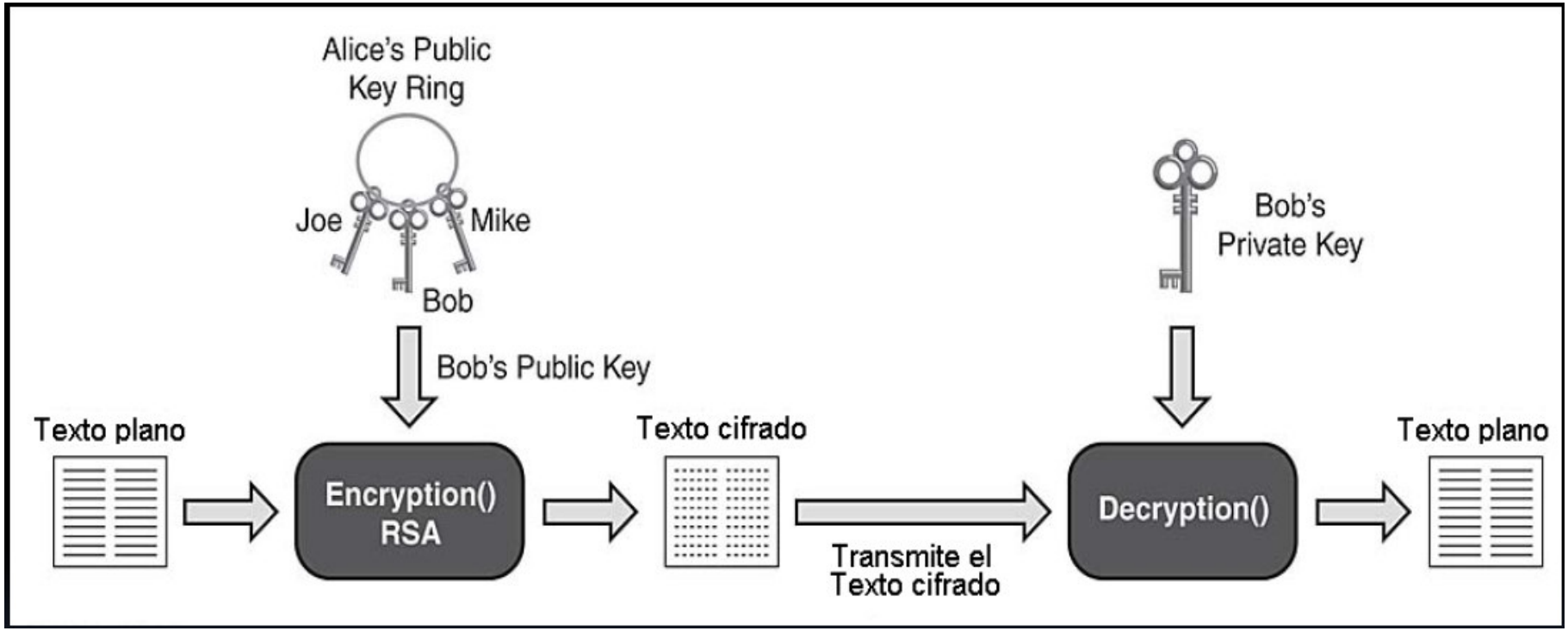
- Aquí ciframos números, no mensajes
- La operación característica de la cifra asimétrica **es mediante un <u>funciones con trampa</u>**
  - Ej: Cifrado exponencial: La operación a realizar será **$C = A^B \bmod n$**, en donde **n** es el cuerpo de cifra del orden de 1.024 bits, **B es una clave pública 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 (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**.

Cifrado Asimétrico: RSA

- **Algoritmo RSA:**
- Hasta ese momento la criptografía sólo abarcaba cifrados simétricos, al menos a nivel de aplicación práctica
- Creado en 1977 por los profesores del MIT Ron Rivest, Adi Shamir y Leonard Adleman
- Surge como **alternativa a los sistemas de cifrado simétrico.** Resuelve la problemática del intercambio de claves
- **Desde su aparición se convirtió en el estándar de los cifrados de clave pública**

Cifrado Asimétrico: RSA

- ¿Cómo funciona RSA?



Cifrado Asimétrico: RSA

- ¿Cómo funciona RSA?
- Bob elige **dos enteros primos** de en torno a 800-1000 bits **P y Q**, que serán secretos, y obtiene N= P*Q, **que podrá ser público** y tendrá una longitud en torno a 2000 bits
- Bob **elige el exponente de cifrado E**, que es un entero que debe cumplir que:
  - *MáximoComúnDivisor*(E, ø(N)) = 1
  - Siendo ø(N) = (P-1)*(Q-1)
  - *MáximoComúnDivisor* => El mayor número que los divide
- Por ejemplo si N= 15 = 3* 5, entonces tenemos que:
  - ø(15) = (3-1)*(5-1) = 8 y por lo tanto el número E que escoja Bob no debe ser múltiplo de 2.

Cifrado Asimétrico: RSA

- ¿Cómo funciona RSA?
- Ahora que tenemos el exponente de cifrado E, calculamos el exponente de descifrado D, que se obtiene con la operación:
  - D*E ≈ 1 módulo (ø(N)); *la operación módulo debe dar 1*
- Ya tenemos todo lo que necesitamos:
  - **Información secreta de Bob: P, Q y D**
  - **Información pública que poseen todos: N, E y el mensaje cifrado por Alice C**

Cifrado Asimétrico: RSA

- ¿Cómo funciona RSA?
- Alice ahora **quiere enviar un mensaje M** a Bob, entonces utiliza la clave pública de cifrado E y el número N para obtener C
- **Suponiendo que M≤N, y si no se trocea en partes de tamaño N o menor**:
  - **$C = M^E$ módulo (N)**

Cifrado Asimétrico: RSA

- ¿Cómo funciona RSA?
- Bob recibe el mensaje cifrado C y lo descifra con su exponente de descifrado o clave privada D
  - M= C^D módulo (N)

¡Magia!

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

- Alice puede estar segura de que Bob ha descifrado el mensaje y de que nadie más lo puede hacer

Cifrado Asimétrico: RSA

RSA: ¿por qué es $m = (m^e \bmod n)^d \bmod n$

Resultado útil de la teoría de los números:

si p,q primos y $n = pq$, entonces:

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

Cifrado Asimétrico: RSA

- ¿Por qué es seguro RSA?
- **Pregunta**: Si todo el mundo conoce N y E (*C= M^E módulo (N))*, y sabemos que D*E ≈ 1 módulo (ø(N)), ¿por qué no despejamos D de esa operación, y así podemos descifrar todos los mensajes de Bob?
- La respuesta es que ES MUY (MUY ES MUY) DIFÍCIL obtener ø(N) a partir de N, y la única manera factible es conocer P y Q y calcular (P-1)*(Q-1)

Cifrado Asimétrico: RSA

- ¿Por qué es seguro RSA?
- **Pregunta**: Si necesito conocer P y Q, pero yo sé que N= P*Q, y además que son primos ¿por qué no intento averiguar cuánto valen ambos?
- La respuesta es que es MUY (MUY ES MUY) DIFÍCIL hallar P y Q a partir de N, cuando P y Qson MUY (MUY ES MUY) GRANDES (2^{900} aproximadamente) ya que requiere una capacidad computacional muy elevada.

Cifrado Asimétrico: RSA

- **¿Cuál es la debilidad de RSA?**
- Como hemos visto, la fortaleza y a su vez la debilidad del cifrado es que actualmente **la capacidad computacional** que se requiere para factorizar un número N en dos primos enteros P y Q no está disponible, al menos para descifrar un simple mensaje. *En el momento en que se elijan mal los números P y Q, o se consiga la capacidad computacional necesaria, el cifrado estará vulnerado*
- **Solución: Elegir primos más grandes**: 1500, 1800, 2000 bits… el problema es encontrar esos primos, ya que cada vez están más distanciados entre si

Cifrado Asimétrico: RSA

- **Ventajas:**
  - No hay que acordar ninguna clave secreta
  - Actualmente se puede lograr un cifrado altamente seguro si se eligen bien las claves
- **Desventajas:**
  - El proceso de cifrado y descifrado **es más costoso computacionalmente** que al aplicar un algoritmo simétrico
    - Solución: utilizar un algoritmo **asimétrico para acordar una clave simétrica**
  - La dificultad de encontrar un par de primos P, Q que dificulten los ataques al cifrado

Cifrado Asimétrico: Diffie-Hellman

- ¿Qué es el algoritmo de Diffie-Hellman?

- El algoritmo Diffie-Hellman **fue el primer algoritmo asimétrico**. Se describía en el famoso articulo "*New directions in Cryptography*" publicado en noviembre de 1976, se utilizaba para ilustrar un ejemplo de la criptografía que Diffie y Hellman acababan de descubrir, la criptografía de clave pública.

- Este algoritmo **permite que dos partes, comunicándose mediante un canal no cifrado, se pongan de acuerdo en un valor numérico sin que un tercero, que tiene acceso completo a la conversación, pueda conocerlo o calcularlo, al menos en un tiempo práctico.**

- **Solamente se puede utilizar para intercambiar claves simétricas**, pero ésta es una de las principales funciones de los algoritmos asimétricos, así está muy extendido **en sistemas de Internet con confidencialidad de clave simétrica** (VPNs, SSL, etc...).

Cifrado Asimétrico: Diffie-Hellman

Descripción

- Es un **algoritmo de intercambio seguro de claves**.

- La seguridad del algoritmo depende de la dificultad **del cálculo de un logaritmo discreto (diferente del otro algoritmo asimétrico que hemos visto: RSA)**. Esta función es la inversa de la potencia discreta, o sea, de calcular una potencia y aplicar una función mod

Cifrado Asimétrico: Diffie-Hellman

- La generación de claves públicas es el siguiente:
  - Se busca un número **grande y primo** llamado ***p***.
  - Un número ***a*** tal que ***a < p***, y a es raíz primitiva de ***p***
  - Se dice que ***a,*** es raíz primitiva de ***p*** si las potencias de ***a*** generan todos los enteros desde **1** hasta ***p–1 .*** Fijada b una raíz primitiva módulo p, cualquier entero a que no sea divisible entre p: $a = b^r \pmod{p}$
  - ***a*** y ***p*** son claves públicas.

Cifrado Asimétrico: Diffie-Hellman

Generación de claves de usuario

- La generación de claves de usuarios es el siguiente:

USUARIO A

- Selecciona clave privada $X_A < p$
- Calcula clave pública $Y_A = a^{X_A} \bmod p$

USUARIO B

- Selecciona clave privada $X_B < p$
- Calcula clave pública $Y_B = a^{X_B} \bmod p$

Cifrado Asimétrico: Diffie-Hellman

Cálculo de clave compartida

Intercambio

- El usuario A transmite al usuario B su clave pública Y_A. El usuario B le envía su clave pública Y_B.

Usuario A

Calcula $K = (Y_B)^{X_A} \bmod p$

Usuario B

Calcula $K = (Y_A)^{X_B} \bmod p$

Demostración

- Se verifica que:

$(Y_B)^{X_A} \bmod p = (a^{X_B})^{X_A} \bmod p = (a^{X_B * X_A}) = (a^{X_A})^{X_B} \bmod p = (Y_A)^{X_B} \bmod p$

Y por lo tanto ambas claves K resultan idénticas.

Cifrado Asimétrico: Diffie-Hellman

1. Both nodes agree on two values (g and n)

2. Generate a random value (x)

3. $A = G^x \bmod n$

2. Generate a random value (y)

3. $B = G^y \bmod n$

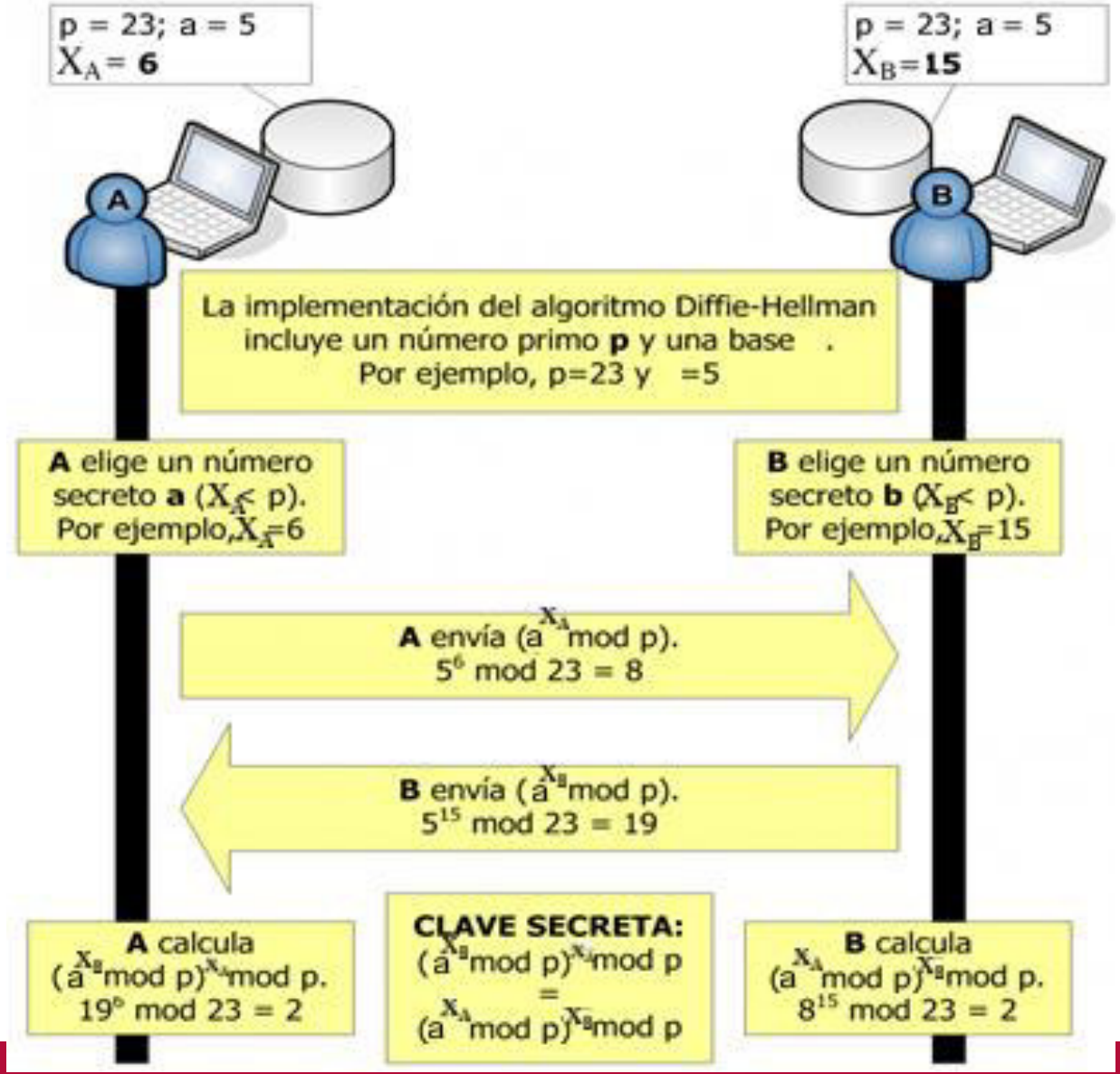
4. A and B Values are exchanged

5. $K1 = B^x \bmod n$

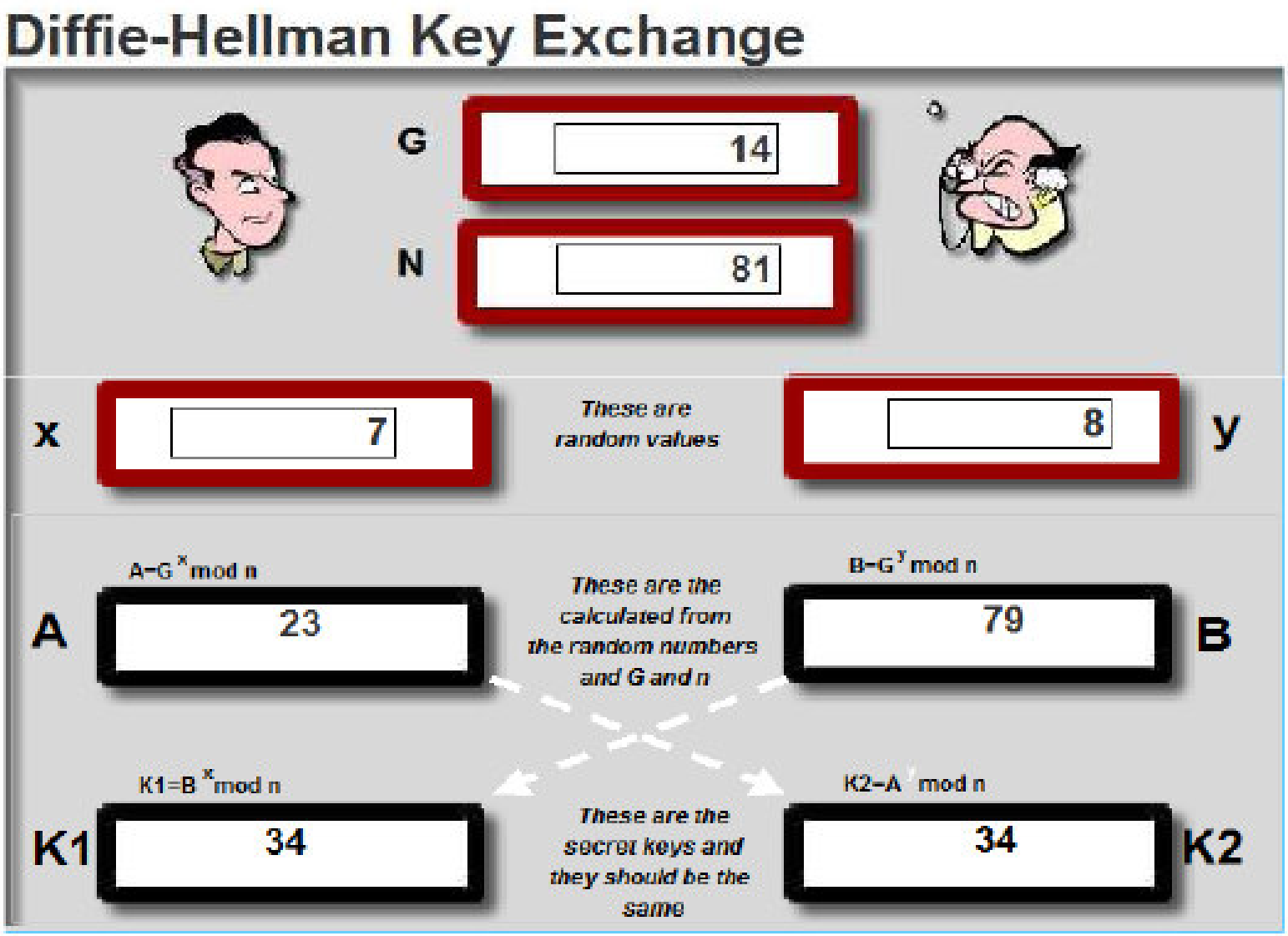
5. $K2 = A^y \bmod n$

$K1$ and $K2$ should be the same and are the secret key

Cifrado Asimétrico: Diffie-Hellman



Cifrado Asimétrico: Diffie-Hellman



Cifrado Asimétrico: Diffie-Hellman

- Ataque pasivo:
  - Los valores de “p” y “a” **son públicos y cualquier atacante puede conocerlos**, pero esto no supone una vulnerabilidad. Aunque un atacante conociese sus valores y capturara los dos mensajes enviados entre las máquinas A y B, no sería capaz de averiguar la clave secreta. La información capturada por un atacante sería la siguiente:

$$(a\ X_A \bmod p) = 8 \rightarrow (5\ X_A \bmod 23) = 8$$

$$(a\ X_B \bmod p) = 19 \rightarrow (5\ X_B \bmod 23) = 19$$

Cifrado Asimétrico: Diffie-Hellman

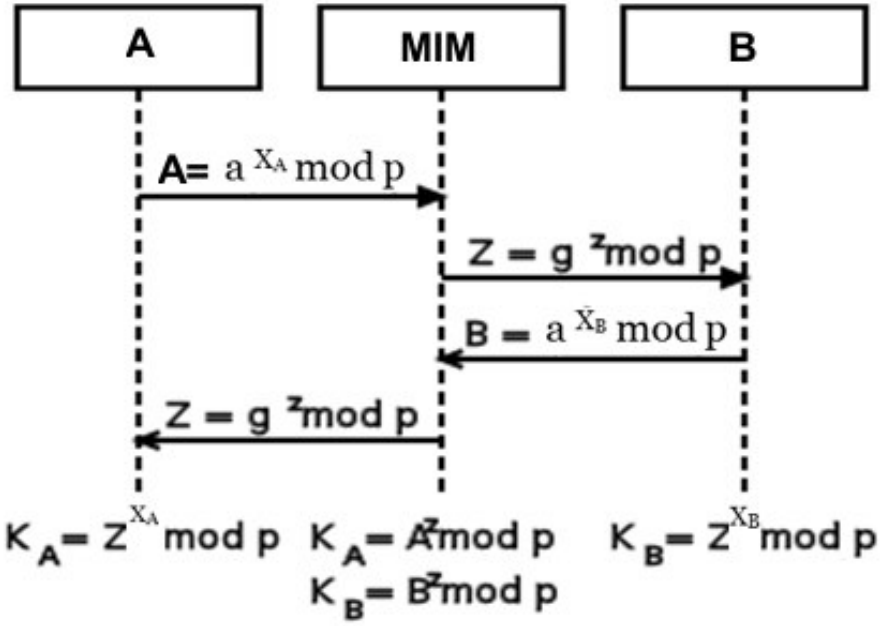
Seguridad del intercambio de clave de DH

- 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 $\alpha^i \bmod p = C$.
- Como p y α serán públicos, al capturar el valor C el atacante deberá resolver $i = \log_\alpha C \bmod p$, **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.

Cifrado Asimétrico: Diffie-Hellman

- Ataque activos:
  - El protocolo es sensible a ataques activos del tipo ***Man-in-the-middle***. Si la comunicación es interceptada por un tercero, **éste se puede hacer pasar por el emisor cara al destinatario y viceversa**, ya que no se dispone de ningún mecanismo para validar la identidad de los participantes en la comunicación. Así, el "*Man-in-the-middle*" podría acordar una clave con cada participante y retransmitir los datos entre ellos, escuchando la conversación en ambos sentidos.
  - Una vez establecida la comunicación simétrica el atacante tiene que seguir en medio interceptado y modificando el tráfico para que no se den cuenta.
  - Observar que para que el ataque sea operativo **el atacante tiene que conocer el método de cifrado simétrico que será utilizado**.

Cifrado Asimétrico: Diffie-Hellman



- Para evitar este tipo de ataque se suele usar una o más de las siguientes técnicas:
  - **Control de tiempos**
  - **Autenticación previa de las partes.** Por ejemplo usar en protocolo de capa subyacente autenticación. Podríamos primero establecer una conexión TLS y sobre esa capa aplicar el algoritmo de Diffie-Hellman.
  - **Autenticación del contenido**. Por ejemplo podríamos usar MAC sobre el contenido de los mensajes

Criptool: https://www.cryptool.org/en/

- **Ejercicio 5:**
  - Realiza una simulación del algoritmo Diffie-Hellman, usando ”procedimientos individuales”->”Protocolos”

- **Ejercicio 6 (PARA MATRICULA DE HONOR):**
  - Envía un mensaje en RSA a un compañero:
    - Define el mensaje
    - Codificalo (cuidado: codificar no es cifrar!)
    - Cifralo
    - Descifralo
    - Decoficalo
    - ¿coínciden?

- **Soluciones al Ejercicio 6:**
- Para facilitar los cálculos, partimos de una codificación del alfabeto que transforma A-Z en los números del 0 al 25.
- El usuario B ha elegido los primos pb = 281 y qb = 167, obteniendo nb = 281 · 167 = 46.927
- El orden de este grupo es $\phi(46.927) = 280 \cdot 166 = 46.480$.
- B elige eb = 39.423 y comprueba que mcd(39.423, 46.480) = 1.
- Entonces determina el inverso de 39.423 módulo 46.480 y obtiene db = 26.767.
- La clave pública de B es (nb, eb) = (46.927, 39.423) y el resto de valores se mantienen secretos.

- Cualquier otro usuario, para enviar un mensaje a B, ha de mirar primero su longitud: el mensaje tiene que pertenecer al grupo en que trabajamos;
- en este caso, no puede superar nb = 46.927. En la realidad los valores son mucho más grandes y no presentan problemas.

- Codificamos cada letra en base 26, debido a que $26^3 = 17.576 < nb < 456.976 = 26^4$, el mensaje a codificar en este ejemplo puede tener hasta tres letras.

- El usuario envía a B el mensaje YES. En base 26 cambia a :

$$Y \cdot 26^2 + E \cdot 26 + S = 24 \cdot 26^2 + 4 \cdot 26 + 18 = 16.346 = m$$

- Ahora ciframos m con la clave pública de B, y obtenemos: $c = m^{eb} \bmod nb = 16.346^{39.423} \bmod 46.927 = 21.166$
- Por lo tanto el mensaje a enviar será 21.166 o lo que es lo mismo en alfabeto: BFIC

UCLM
UNIVERSIDAD DE CASTILLA-LA MANCHA

Consejería de Educación, Cultura y Deportes
Castilla-La Mancha

DSI
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS

escuela superior ingeniería informática

Módulo 1: Criptografía y Aplicaciones Criptográficas

Sesión 2

José Luis Martínez
Universidad de Castilla–La Mancha

Índice

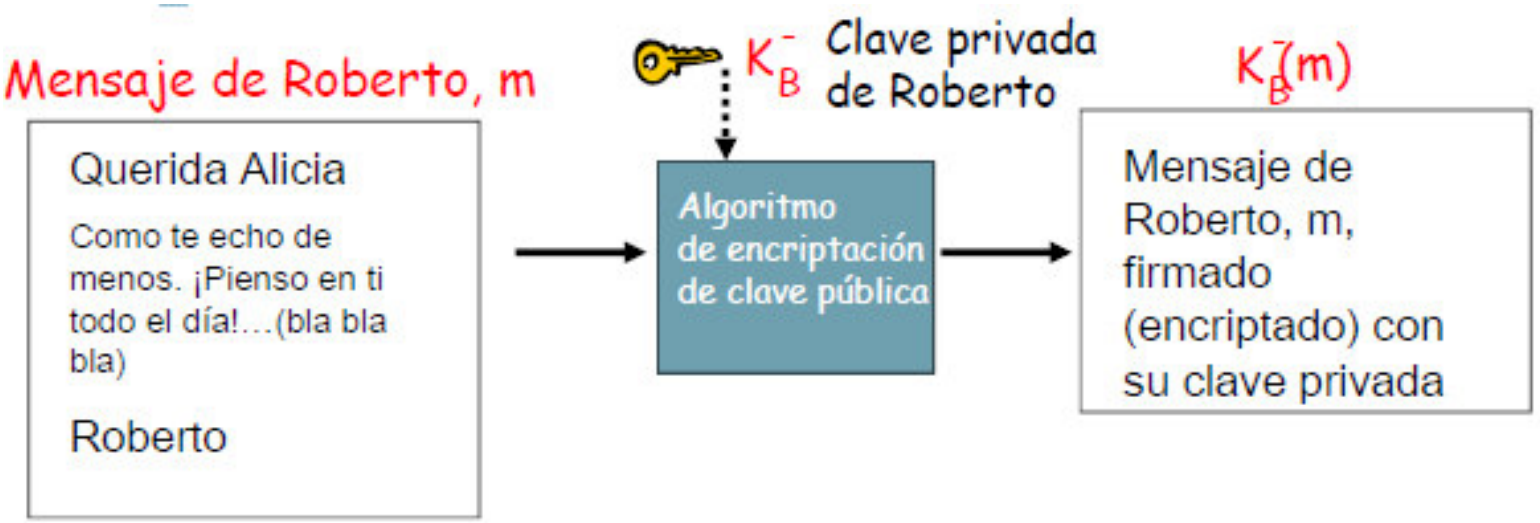
- Píldora 1:
  - Firma Digital
  - Distribución de Claves y Certificación
- Píldora 2:
  - PKI
  - Ejercicio

Firma Digital

- La firma digital es una técnica criptográfica que realiza las mismas funciones que la firma en papel, pero en el mundo digital:
- Atestigua que el firmante tiene conocimiento y está de acuerdo con el contenido del documento firmado
- La firma digital debe ser:
  - **Verificable:** se puede probar que el documento fue firmado por la persona indicada.
  - **No falsificable:** sólo la persona indicada pudo haber firmado el documento.
  - **No repudiable**
  - Se utilizan técnicas de criptografía de clave pública.

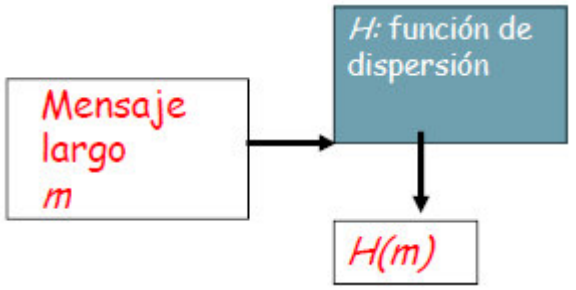
Firma Digital

- Firma digital simple para mensaje m
- Roberto firma m encriptándolo con su clave privada K_B^- , creando un mensaje "firmado", $K_B^-(m)$



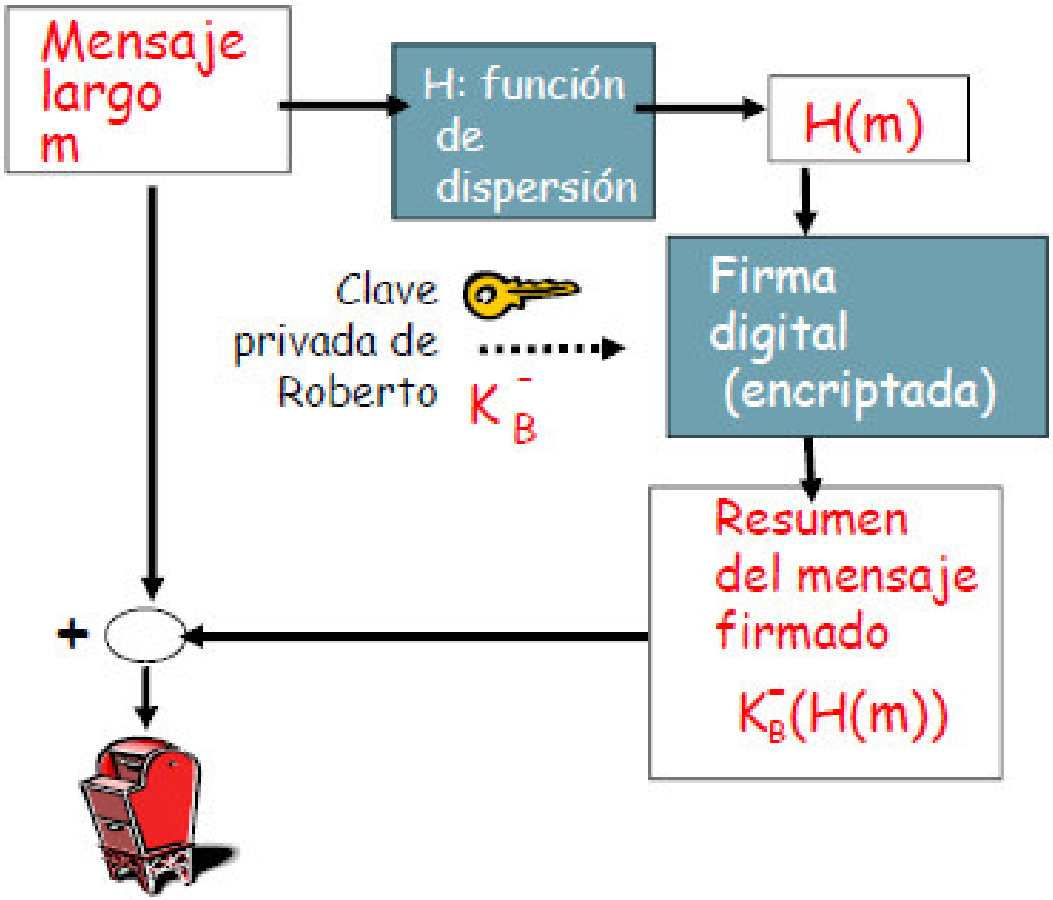
Firma Digital

- Problemas:
- Es **computacionalmente caro encriptar** con clave pública mensajes largos.
- Se pueden alcanzar los objetivos con un **resumen del mensaje**
- Procedimiento:
- Calcular, a partir de un mensaje m, una huella dactilar de longitud fija fácil de fija, computar: H(m).



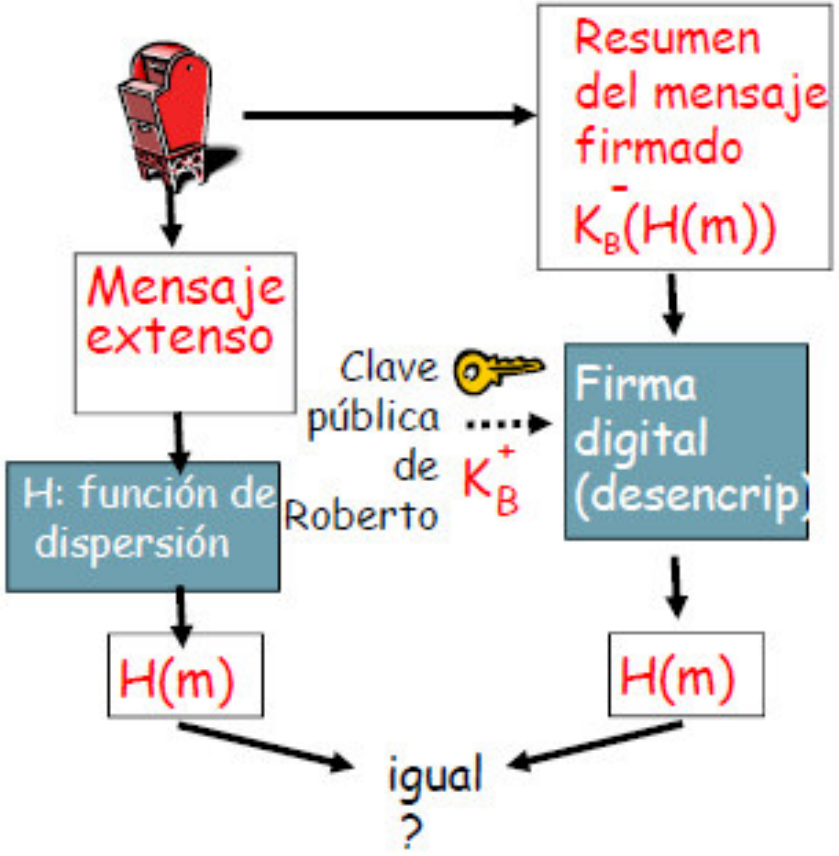
Firma Digital

- Roberto envía mensaje firmado digitalmente:



Firma Digital

- Alicia verifica la firma y la integridad del mensaje firmado digitalmente:

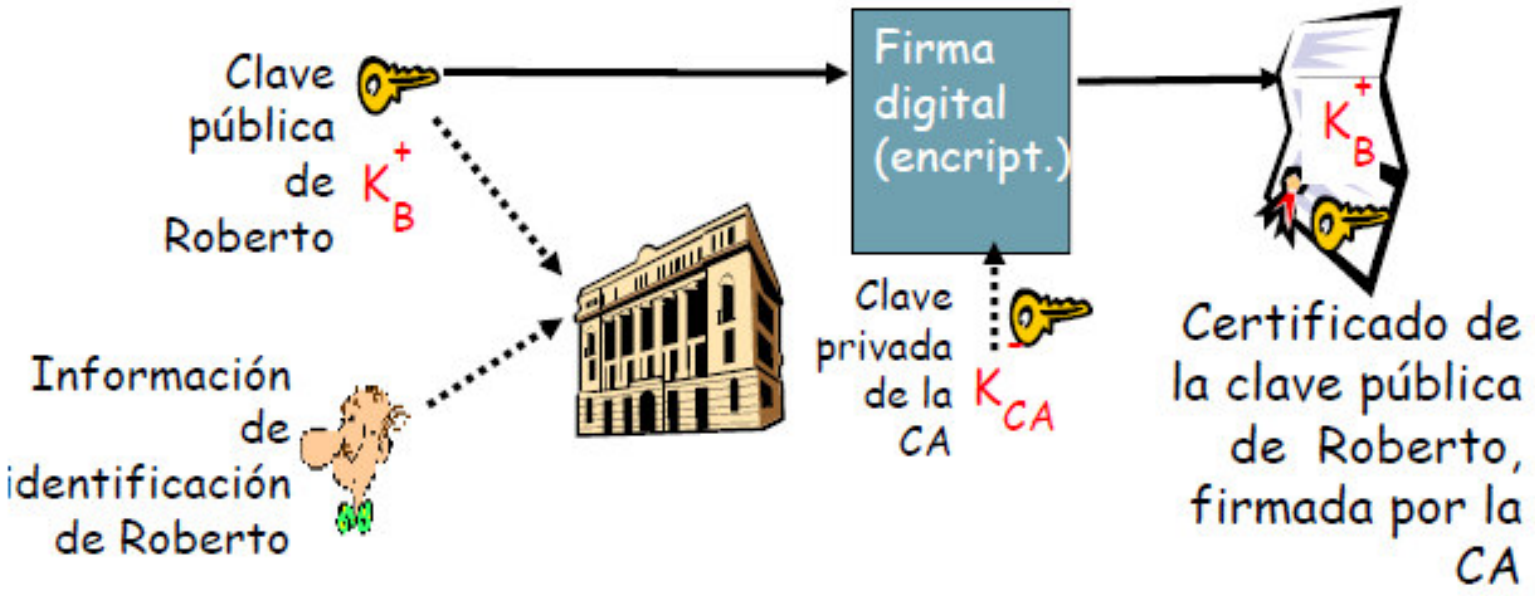


Distribución de claves y certificación

- **Autoridad de certificación (CA)**
- **Vincula una clave pública a una entidad particular** (persona, router, etc).
- Funciones de una CA:
- **Verifica que un entidad es quien dice que es** a través de diferentes procedimientos.
- **Una vez verificada la entidad, crea un certificado que asocia la clave pública de la entidad con su identidad.** El certificado contiene la clave pública y una información que identifica de forma única al poseedor de la clave pública. **El certificado se firma digitalmente por la CA**

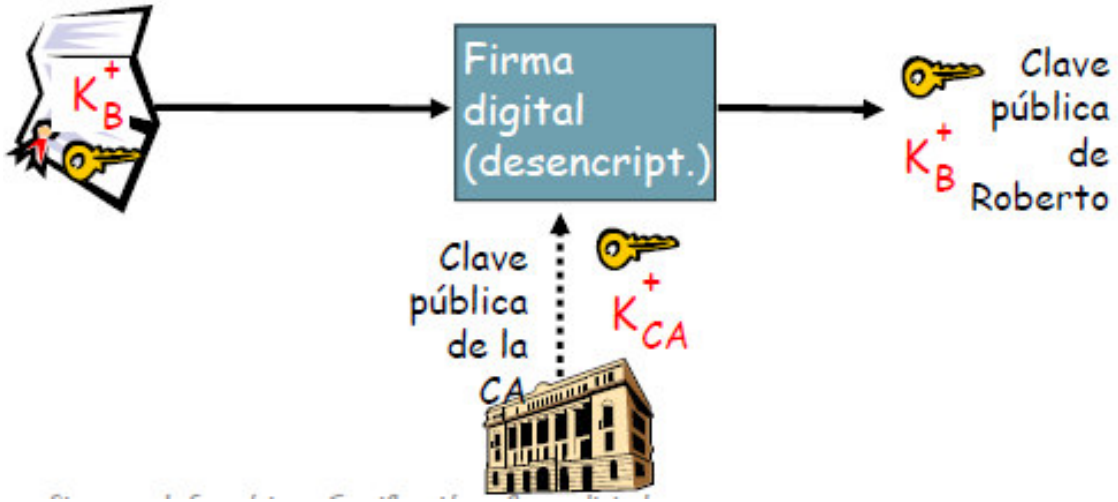
Distribución de claves y certificación

- Autoridad de certificación (CA)



Distribución de claves y certificación

- Autoridad de certificación (CA)
  - Cuando Alicia quiere la clave pública de Roberto:
  - Obtiene el **certificado de Roberto** (de su página web, en un mensaje de correo, en un servidor de certificados, ..)
  - **Aplica la clave pública CA al certificado de Roberto, obtiene la clave pública de Roberto**.

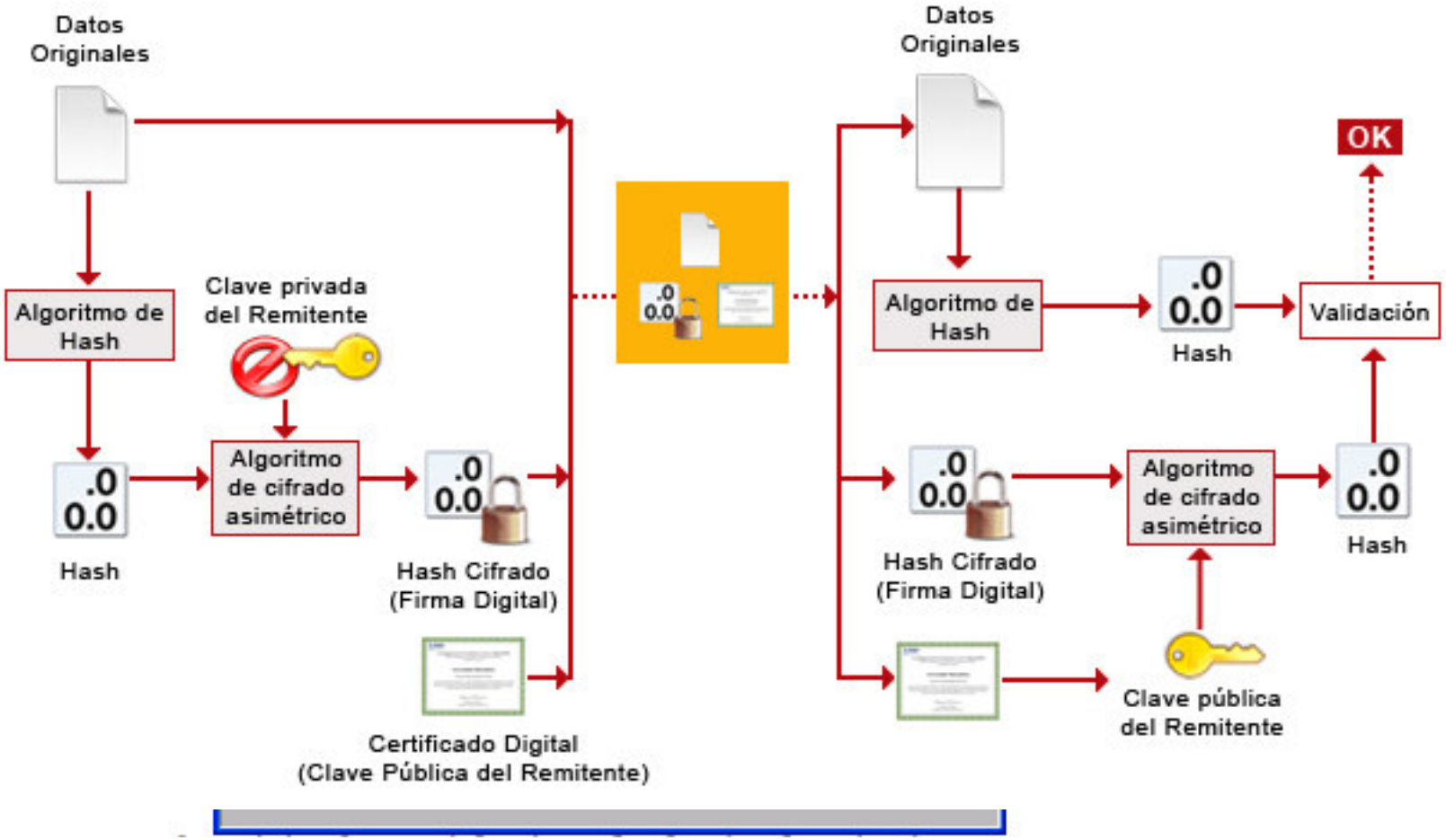


Distribución de claves y certificación

- Certificado Digital permite:
  - **Identificarnos.**
  - **Firmar digitalmente un Documento Digital.**
  - Trabajar con Documentos Digitales firmados digitalmente teniendo certeza respecto del remitente y el destinatario.
  - Efectuar transacciones de tipo comercial con total seguridad y sustento legal.
  - Mantener la confidencialidad de la información entre el Remitente y el Destinatario utilizando cifrado.
  - Estar seguros de que un Documento Digital no ha sido alterado.
- En síntesis, **la utilización de Certificados Digitales garantiza Autentificación, Integridad, Confidencialidad, No Repudio**.

Distribución de claves y certificación

- Firma Digital con Certificado



Criptool: https://www.cryptool.org/en/

- **Ejercicio 1:**
  - Realiza una simulación de firma digital con certificado en “Firma Digital”-> Demostración de Firma

- **Ejercicio 2:**
  - Prueba a firmar un documento cualquier y pásaselo a un compañero para que pueda comprobar la veracidad de dicho documento

PKI

- PKI. Infraestructura de clave pública Vamos a ver ahora como es posible crear e implantar sistemas criptográficos basados en la clave pública que hemos visto que permitan el uso generalizado de los servicios de integridad, confidencialidad, autenticidad y no repudio. Se verá cuál es el problema asociado a la implementación de la criptografía asimétrica, y presentaremos la solución de la infraestructura de clave pública (o PKI, public key infrastructure).

PKI

- PKI.
- Como hemos visto, los criptosistemas de clave pública **permiten el intercambio de mensajes confidenciales** e íntegros de manera ágil, siempre que dispongamos de la clave pública de quien nos queremos comunicar de manera segura. El problema viene dado en el proceso de obtención de esa clave pública y cómo estar seguros de que pertenece a quien necesitamos y no a otra persona

PKI

- Un intruso lo bastante hábil podría tener acceso al directorio donde están ubicadas las claves públicas y sustituir esta de algún usuario por la suya propia. De esta manera, los interlocutores que utilizaran la clave obtenida de ese directorio pensarían que se están comunicando con el primer usuario pero realmente lo harían con el intruso. Es lo que se conoce como ataque del hombre a medio camino (MiTM) .

PKI

- Diffie y Hellman ya pensaron que el problema de la distribución de claves se resolvería con algún **directorio seguro en Internet** en el cual se estableciera de manera unívoca el vínculo entre un identificador de usuario y una clave pública. Si el sistema de directorio público firma todas sus operaciones, nadie puede suplantar una identidad. El problema radicaba en el **bajo rendimiento** que el sistema ofrece con muchos usuarios.

PKI

- L. Kohnfelder en el año 1978 se basó en la idea de una autoridad **central de confianza** y propuso crear unos registros de datos firmados –los certificados– que permitirían que la distribución de claves se hiciera desde directorios públicos que no requirieran confianza. En este caso, un certificado digital es una estructura de datos que contiene información del propietario de las claves criptográficas, la clave pública y una firma digital de los dos campos anteriores que le da validez. Creamos una estructura con la clave pública y una certificación de que es de quien dice ser. La firma, realizada por un usuario o entidad externa leal, asegura la integridad contra una posible modificación no deseada de los datos. Pero eso

PKI

- Creamos una estructura con la clave pública y una certificación de que es de quien dice ser. La firma, realizada por un usuario o entidad externa leal, asegura la integridad contra una posible modificación no deseada de los datos. Pero eso resuelve en parta la distribución de claves, ya que tenemos que decidir quien firma los certificados y si confiamos en esa entidad o no

PKI

Entidad Certificadora

Certificado Digital

- Versión
- Número de serie
- Algoritmo de firma
- Emisor
- Período de validez
- Llave pública
- Extensiones
- Firma AC

HASH

Resumen del Documento de longitud fija

Resumen cifrado con clave secreta

PKI

- Componentes de una infraestructura de clave pública Esenciales:
  - la autoridad de certificación, (CA)
  - los suscriptores,
  - los repositorios,
  - Otros:
    - la autoridad de registro,
    - la autoridad de validación,
    - la autoridad de sellado de tiempos,
  - Todos estos dan una visión completa de las funciones que puede llegar a proporcionar una PKI. Aunque hay otros componentes opcionales como el repositorio de claves o la autoridad documental.

PKI

- Autoridad de certificación
- La autoridad de certificación, conocida como CA, es la entidad responsable de emitir y revocar los certificados. Es la entidad de confianza que proporciona legitimidad a la relación de una clave pública con la identidad de un usuario o servicio. Una PKI puede tener una o más autoridades de certificación. La creación de una CA se inicia con la generación del par de claves (públicas y privadas) que se utilizarán para firmar y validar los certificados digitales que emita esta autoridad de certificación. Las claves tienen que ser lo bastante fuertes para que la probabilidad de que un atacante las rompa sea extremadamente baja durante el tiempo de vida de los certificados que se firmarán.

PKI

- Una vez generado el par de claves, la clave pública se tiene que distribuir de una manera segura entre todas las entidades de confianza per quieran estar en la infraestructura. Esta distribución se puede hacer tanto por medio de un certificado digital emitido por una autoridad de certificación en la que los usuarios ya confían, o por un certificado generado por la propia autoridad de certificación que se acabe de crear (este tipo de certificados reciben el nombre de certificados autofirmados). Para garantizar la seguridad de la transmisión, estos certificados digitales de las autoridades de certificación se tendrán que distribuir desde un canal externo (pasar el certificado físicamente, por correo, incorporado al software, etc.).
- **La seguridad de una infraestructura de clave pública depende de que se utilicen prácticas adecuadas para crear y gestionar toda la infraestructura de confianza**. Es importante que el control de acceso a las claves de la autoridad de certificación sea muy estricto

PKI

- Autoridad de registro
- La autoridad de registro, o RA, es la encargada de **verificar el vínculo entre las claves públicas y la identidad de sus titulares**. Por ejemplo, en España, las delegaciones y administraciones que la Agencia Tributaria tiene en cada localidad y las oficinas consulares actúan como autoridades de registro de la Fábrica Nacional de Moneda y Timbre. Las autoridades de registro son un componente opcional de una infraestructura de clave pública que se utiliza para descargar la autoridad de certificación de muchas de las funciones administrativas. En particular, son especialmente útiles en organizaciones grandes y geográficamente dispersas.

PKI

- Suscriptores y entidades finales
- Los suscriptores y las entidades finales **son aquellos que poseen un par de claves (pública y privada)** y un certificado asociado a la clave pública. Con este par de claves podrán efectuar firmas digitales y cifrar y descifrar documentos. Una entidad final representa un organismo, mientras que un suscriptor es una persona.

PKI

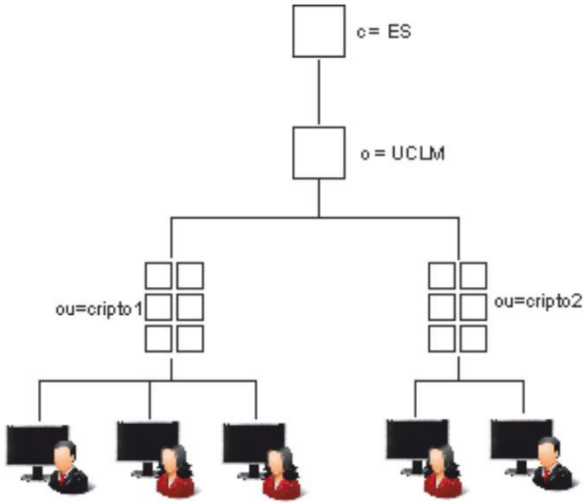
- Usuarios
- Los usuarios son los agentes **que validan firmas digitales** y su ruta de certificación a partir de claves públicas emitidas por autoridades de certificación de confianza. También pueden cifrar documentos para suscriptores y entidades finales. A diferencia de los suscriptores y las entidades finales de una infraestructura de clave pública, los usuarios no tienen por qué disponer de ningún par de claves ni de certificado alguno. Evidentemente, suscriptores y entidades finales son, en particular, usuarios.

PKI

- Repositorios
- Los repositorios son las estructuras encargadas de almacenar la información relativa a la infraestructura de clave pública. Los dos repositorios más importantes en una infraestructura de clave pública son el **repositorio de certificados** y el **repositorio de listas de revocación de certificados**. Una lista de revocación de certificados (CRL) incluye todos aquellos certificados que por diversos motivos son inválidos antes de la fecha de caducidad establecida en el mismo certificado. El tipo de repositorios más utilizados son los directorios. Que son una base de datos optimizada para los accesos de lectura, la navegación y las grandes buscas, Su objetivo es dar respuestas rápidas a un alto volumen de peticiones. **El X.500 es un directorio basado en la arquitectura OSI**. En el modelo X.500, la información está organizada de manera jerárquica y cada nodo final contiene objetos de una clase determinada con los datos.

PKI

- Repositorios
- Repositorios Las entradas al directorio están dispuestas en un árbol y utilizan el subconjunto de atributos denominado nombre distinguido (DN). La situación de la entrada anterior en el árbol X.500 sería, la información está organizada de manera jerárquica y cada nodo final contiene objetos de una clase determinada con los datos.



PKI

- Repositorios
- Inicialmente para acceder a los directorios X.500 se definió el protocolo de acceso al directorio (DAP). Pero este, al ser un protocolo de la ISO y tener el defecto de que no era compatible con el protocolo de Internet TCP/IP, se descartó rápidamente. La IETF en el 1995 añadió a su línea de trabajo el desarrollo de un protocolo que utilizara un modelo de datos X.500, pero que funcionara sobre TCP/IP. Se denominó LDAP (lightweight directory access protocol). En diciembre de 1997 salió a la luz la versión 2 ya como propuesta de estándar para Internet para las infraestructuras de claves públicas.

PKI

- Autoridad de validación
- La autoridad de validación , VA, **es la encargada de comprobar la validez de los certificados digitales que puede ser la propia autoridad de certificación** o una entidad externa. Los protocolos de validación de certificados son mecanismos que proporcionan información sobre el estado actual del certificado (revocado, suspendido, válido o estado desconocido) o relativa a la cadena de certificación necesaria para validar la autenticidad del certificado. Una de las principales deficiencias de las listas de revocación de certificados es que la periodicidad con que se publican las renovaciones no está bajo el control de las aplicaciones que tienen que

PKI

- Autoridad de sellado de tiempo
- La autoridad de sellado de tiempo, TSA, es la encargada de **firmar un mensaje con la finalidad de demostrar que existía antes de un determinado instante de tiempo**. En el documento de definición del servicio, se especifica el protocolo basado en mensajes de petición y respuesta que permiten asociar marcas temporales de confianza a los documentos. La necesidad de una autoridad de sellado de tiempo es importante para la propiedad de no repudio. Los servicios de no repudio tienen que poder establecer la existencia de unos datos antes de determinados momentos. El papel de la TSA consiste en sellar estos datos con el fin de establecer su evidencia.

PKI

- Autoridad de sellado de tiempo
- Algunos ejemplos en que la TSA tiene un papel importante son: • Verificación de la firma digital de un documento: si el certificado correspondiente ha sido revocado, el sello de tiempo nos permitirá establecer si en el momento en que se efectuó la firma del documento, el certificado todavía era válido o no. • Entrega de documentos antes de una fecha límite: el sello de tiempo nos permite indicar el momento en qué se hizo la entrega de determinada información. • Auditorias: el sello de tiempo permite tener fechadas todas las entradas de los históricos

PKI

- Autoridad de sellado de tiempo
- Para que el sellado de tiempo sea válido, la estructura de datos que lo contiene tiene que estar protegida de forma criptográfica y, además, la marca de tiempo que lleva se tiene que haber obtenido de una fuente de confianza. Eso se puede asegurar con la utilización de proveedores oficiales de valores temporales basados en el tiempo coordinado universal (UTC, Universal Time Coordinated), que garantizan una alta precisión en los datos temporales que suministran. Los pasos para poner un sello de tiempo en un mensaje son los siguientes: 1. Se calcula un resumen digital de los datos que se quieren sellar. 2. Se envía el resumen a una TSA, que añade la fecha y la hora, firma

PKI

- Modelos de Confianza
- Para trabajar con criptografía de clave pública es necesario el uso de unos certificados que nos aseguren la correspondencia entre la clave pública y alguna identidad de la entidad o la persona propietaria del certificado. Los métodos de certificación absolutos son imposibles, ya que un certificado no se puede certificar a sí mismo. Por esta razón se han propuesto diferentes métodos que pretenden tratar esta situación:
  - el modelo distribuido de red de confianza,
  - el modelo plano,
  - el modelo jerárquico,
  - el modelo de navegación por lista de confianza, • el modelo de

PKI

- El sistema distribuido de red de confianza es el modelo más sencillo de utilizar, adecuado para un grupo pequeño de usuarios que ya tenían tratos antes de la implantación de la infraestructura de clave pública. En el modelo distribuido cada usuario crea y firma certificados para la gente que conoce. No se tiene que desarrollar ninguna infraestructura central con una tercera entidad de confianza en que dé fe de la identidad de los usuarios. Un ejemplo de este sistema es el software PGP. Se trata de un proyecto iniciado a principios de los años noventa que con el paso del tiempo se ha convertido en uno de los mecanismos más populares y fiables para proporcionar servicios criptográficos en las comunicaciones, sobre todo por correo electrónico el modelo Bridge CA

PKI

- El software PGP asume que sólo los usuarios individuales tienen competencias para decidir en quién confían y en qué grado lo hacen. Los mismos usuarios firman o dan autenticidad a los certificados. Si un usuario A se quiere comunicar con un usuario B, se tendrán que intercambiar las claves públicas por un canal inseguro. Estas claves estarán empaquetadas en un formato de certificado autofirmado que, además, podrá ser firmado por otras personas. Con el fin de asegurar que las claves públicas pertenecen efectivamente a cada usuario, tanto A como B calculan el valor del hash de sus claves y se lo intercambian por un canal inseguro diferente al que han utilizado para intercambiarse la clave (por ejemplo, por teléfono o correo regular). Los usuarios calculan el hash de la clave del

PKI

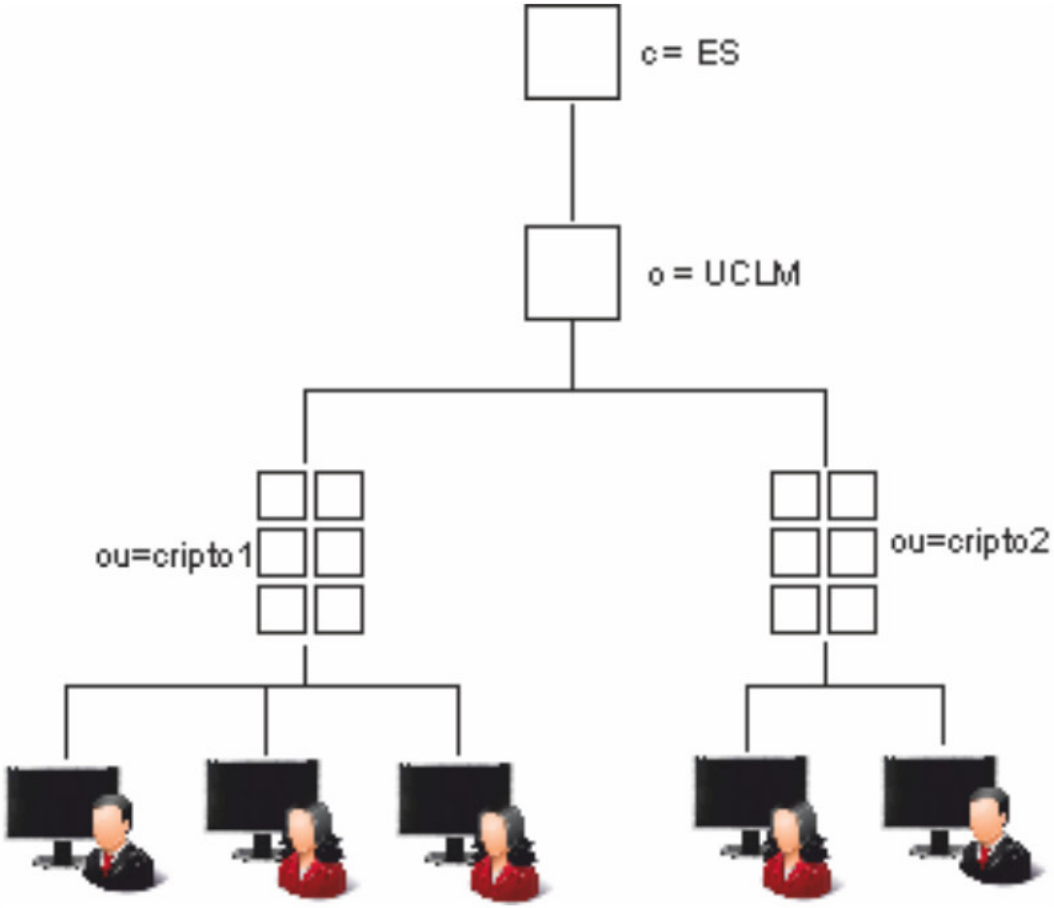
- En el caso del PGP, cada usuario guarda las claves públicas en unas estructuras denominadas anillos de claves. Una clave es válida si está firmada (certificada) por el mismo usuario propietario del anillo o por bastantes personas de fiar. Cada clave tiene asociado un nivel de confianza donde establece hasta qué punto se confía en los certificados emitidos por esta clave. Cuanto más bajo es el nivel de confianza, más certificados hacen falta para validar la clave. Cuando queremos validar el certificado de un usuario cualquiera, utilizamos aplicaciones de busca de rutas de certificación para crear una cadena de confianza adaptada a nuestras necesidades. Damos la aplicación al identificador de clave de nuestro certificado y el identificador del certificado para validar. El servidor de

PKI

- El usuario user x confirma con la emisión de una firma que la clave B pertenece realmente al user B. Por otra parte el user A ha firmado la clave del user x, puede confirmar que la clave B pertenece al usuario esperado. Además, hay otro camino de validación que también va desde la clave de user A hasta la del user B. Es importante que todo el mundo firme claves para que otros usuarios se puedan beneficiar de las firmas. Todas estas firmas forman una especie de red, y por eso se afirma que el PGP es un modelo de red de confianza. El PGP utiliza RSA (con hash MD5) y Diffie-Hellman (con SHA-1), con longitud de clave de hasta 2.048 (RSA) o 2.096 (DH). Los algoritmos de clave simétrica que puede utilizar son CAST (predeterminado), IDEA y Triple DES.

PKI

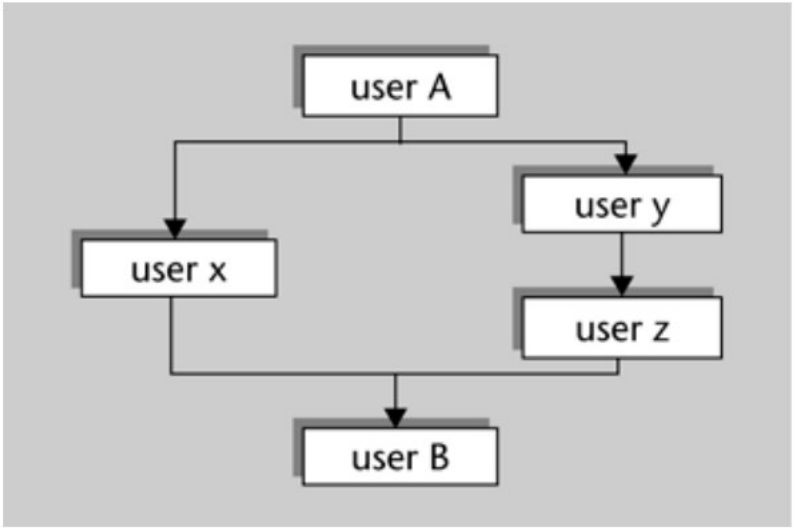
- El



PKI

- El modelo plano Es el sistema más sencillo de PKI que incluye una única autoridad de certificación como tercera parte de confianza encargada de la emisión y la gestión de los certificados. Los usuarios pueden validar la identidad de los suscriptores a partir del certificado de la autoridad de certificación. La autoridad de certificación posee un certificado que ella misma ha generado y en la cual los usuarios depositan su confianza. En un certificado autofirmado, la clave pública que se certifica corresponde a la clave privada que se utiliza para firmar el certificado. El nombre del emisor y el titular del certificado son el mismo. Suele usarse en el ámbito de las intranets.

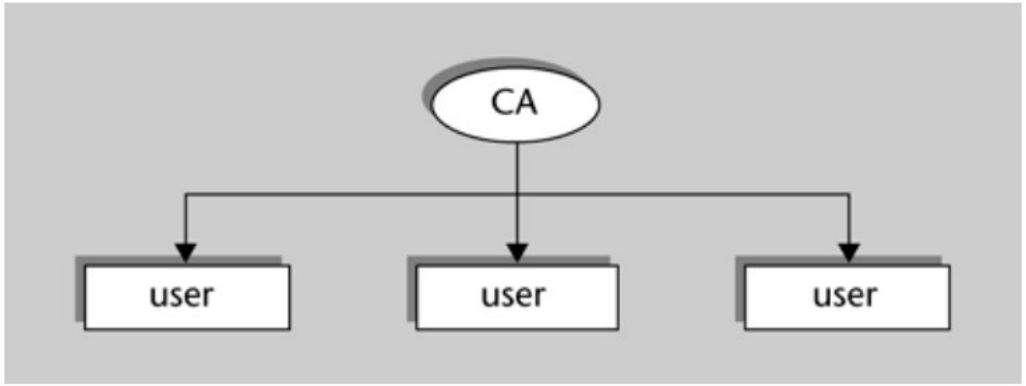
PKI



PKI

- El modelo jerárquico Los certificados de los suscriptores y las entidades finales están firmados por una entidad externa que también se identifica con certificados que emitirá una autoridad de certificación de jerarquía superior. Los certificados de la autoridad de certificación de jerarquía superior pueden estar a la vez certificados por otras autoridades de certificación, y así sucesivamente, hasta llegar a una autoridad de certificación que tiene un certificado autofirmado. Esta se denomina autoridad de certificación raíz, y las que dependen de ella son las autoridades de certificación subordinadas. Si los usuarios confían en la autoridad de certificación raíz, este voto de confianza se extenderá por toda la jerarquía. Para verificar la autenticidad de un certificado un usuario puede

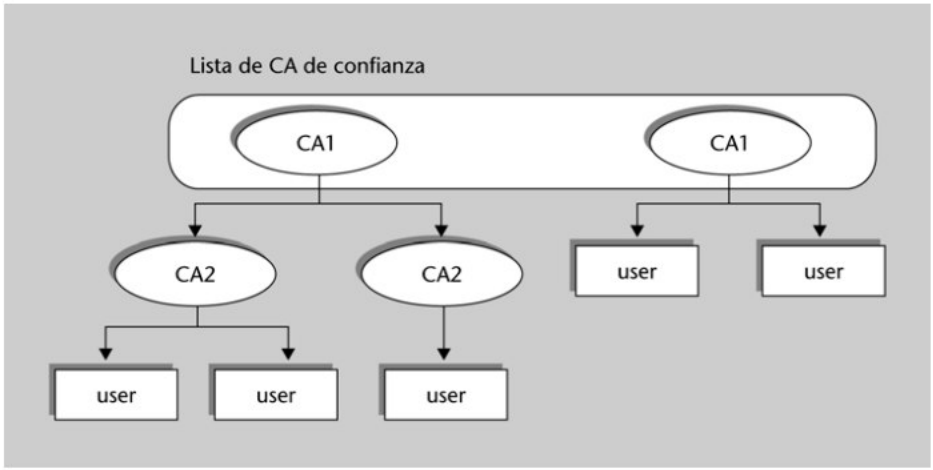
PKI



PKI

- El modelo de navegación por lista de confianza Existen muchas y muy diferentes CA raíces, y cada una ofrece servicio a pequeñas comunidades de usuarios. Para conectar entre sí a estas islas de infraestructuras de clave pública han surgido formas híbridas del modelo jerárquico que permiten la interfuncionalidad entre diferentes grupos de usuarios controlados por una autoridad de certificación raíz. La solución más común para interconectar infraestructuras de clave pública jerárquicas es el modelo de navegación por lista de confianza, también conocido como modelo centrado en el usuario, en el que cada aplicación final tiene una lista de las claves públicas de todas las autoridades de certificación en que confía. Este modelo, implementado por la mayoría de los navegadores

PKI


Lista de CA de confianza
CA1
CA1
CA2
CA2
user
user
user
user
user

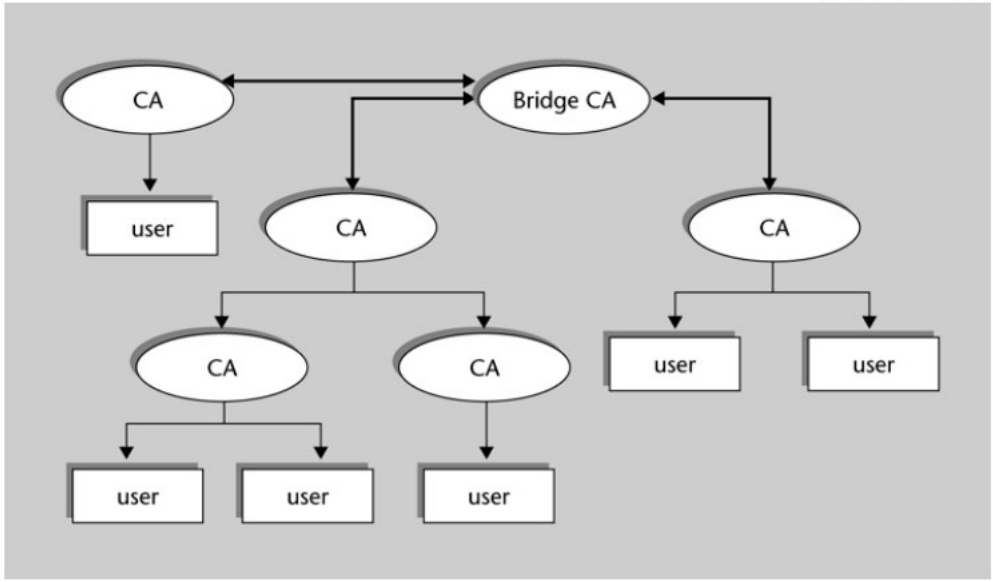
PKI

- El modelo de certificados cruzados En el modelo de certificados cruzados, las autoridades de certificación raíz de cada comunidad de usuarios emiten certificados para otras autoridades de certificación que tienen funciones y políticas equivalentes a las suyas. Como en el modelo jerárquico, cada usuario sólo confía en una autoridad de certificación raíz, pero en este modelo, la autoridad de certificación raíz es de ámbito local, y no central, como en el anterior. La interconexión entre diferentes islas de infraestructuras de clave pública se realiza por el certificado cruzado de una autoridad de certificación a otra. En un certificado cruzado, el titular y el emisor son autoridades de certificación diferentes. Este tipo de

PKI

- El modelo Bridge-CA La interconexión de diversas infraestructuras de clave pública por certificados cruzados es muy práctica porque no requiere el establecimiento de ninguna autoridad de certificación raíz común, pero presenta el inconveniente de que es un modelo poco reescalable. Para resolver este inconveniente surgió el modelo Bridge-CA, que incorpora una autoridad externa que actuará de puente entre las infraestructuras de clave pública que se quieren unir. Todas las autoridades de certificación establecen una certificación cruzada con el punto central de referencia, el Bridge-CA, y de esta manera se crea una comunidad de confianza más amplia. El Bridge-CA no es un punto central de confianza, sino una entidad que facilita la certificación cruzada entre todas las CA

PKI



Ejercicio

- Creación de una entidad certificadora(PKI) para desplegar una red inalámbrica mediante autenticación por certificado dentro de una empresa

Ejercicio

- Como especialistas en criptografía el CIO de nuestra empresa nos pide que cambiemos la configuración actual de la red WiFi interna ya que ha detectado que ha habido muchas conexiones desde dispositivos ajenos a empleados de la compañía. Actualmente la red WiFi dispone de un cifrado WPA, que es el que viene por defecto en el router. Para asegurarnos de que no vuelva a pasar y nadie ajeno a la compañía se conecte, proponemos cambiar la configuración y crear identidades digitales a todos los empleados, que haremos servir para poder conectarse a la red mediante estos certificados emitidos y controlados por nuestro departamento de IT.

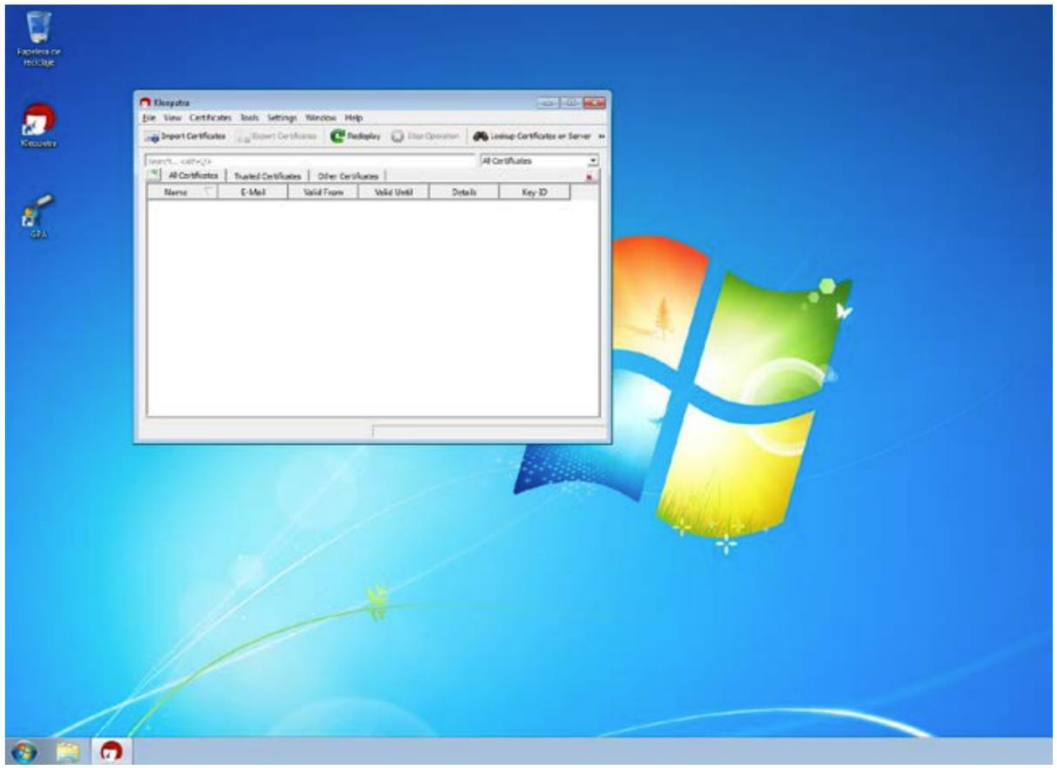
Ejercicio

- Para ello seguiremos los siguientes pasos: Realizar los ejercicios mediante algún sistema que permita generar claves asimétricas que nos deje cifrar y firmar documentos o mensajes. Por ejemplo hacer servir GnuPG (Gnu Privacy Guard) para la versión Linux o Windows y generar un par de claves para poder firmar y cifrar documentos. Guardar bien estas claves.

- https://gnupg.org

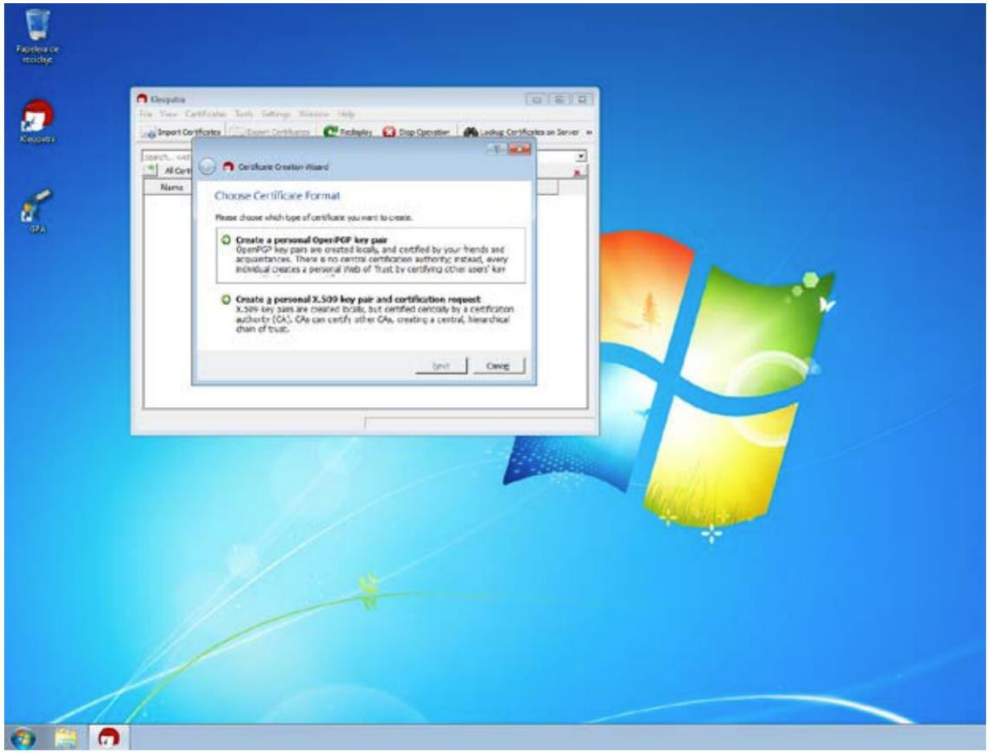
Ejercicio

- Para Windows podemos instalar el GnuPG. Veremos que hay dos programas instalados, GPA y Kleopatra. Este último es la versión más sencilla del primero



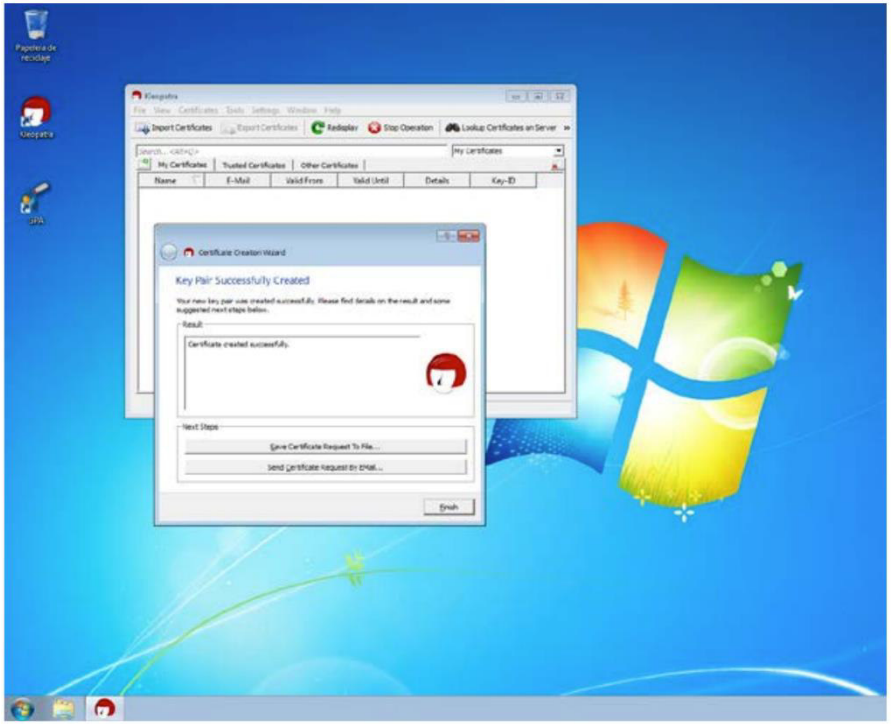
Ejercicio

- Se puede abrir el menú File del programa Kleopatra y crear un nuevo certificado, que puede ser una pareja de claves para firmar y cifrar o además crear un certificado asociado a estas claves, y crear también la CA correspondiente.



Ejercicio

- Se debe crear una pareja de claves con certificado, y la consecuente petición a CA asociada que, para tener más seguridad las crearemos con RSA de 4096 bits, todo y que por ahora con 2048 bits sería suficiente.



Ejercicio

- Con esto ya tenemos el archivo de solicitud de certificado. Es decir, tenemos el par de claves y la solicitud de certificado asociado a ellas, que podemos enviar por correo o guardar, para la CA que queramos que nos dé autenticidad. Lo guardamos para poder después usarlo en el siguiente paso. A partir de aquí vamos a crear una entidad de certificación propia en un sistema Linux Debian/Ubuntu. Aunque no la validemos con otra entidad superior, para nuestra empresa será suficiente. En el caso de querer asociarnos a otra entidad deberíamos validar esta CA que vamos a crear. Crearemos un directorio raíz para toda la CA.

Ejercicio

- # mkdir -m 0755 /CA
- # mkdir -m 0755 /CA/privado
- # mkdir -m 0755 /CA/certificados
- # mkdir -m 0755 /CA/nuevoscertificados
- # mkdir -m 0755 /CA/crl

Ejercicio

- Así tendremos ordenados las claves privadas en “privado”, los certificados de petición en “certificados”, los certificados nuevos que se generen en “nuevoscertificados” y la lista de certificados revocados en “crl”
- Para configurar inicialmente la CA copiaremos el fichero de configuración por defecto de openssl que en Debian/Ubuntu está en /etc/ssl/ y lo protegeremos
- # cp /etc/ssl/openssl.cnf /CA
- # chmod 0600 /CA/openssl.cnf.

Ejercicio

- Creamos dos archivos que funcionan como bases de datos para openssl, después crearemos la base de datos para la aplicación gráfica.
- # touch /CA/ca_db
- # echo '01' > /CA/serial

Ejercicio

- Vamos a crear un certificado autofirmado que será utilizado como el certificado de nuestra CA. Este será utilizado para firmar los pedidos de certificados que se reciban de los empleados:
- # cd /CA
- # openssl req -config openssl.cnf -new -x509 –newkey rsa:4096 –days 3650 –extensions v3_ca -keyout privado/ca.key –out certificados/ca.pem.

Ejercicio

- Nos pedirá diferentes datos, pero el más importante es el Common Name que es el que da nombre a la CA que se quiere crear.
- Se crearán dos nuevos ficheros en certificados/ca.pem, certificado público de la CA y privado/ca.key, clave privada del certificado de la CA, a pesar de que está protegida por una contraseña mejor restringir el acceso:
- # chmod 0400 /CA/privado/ca.key

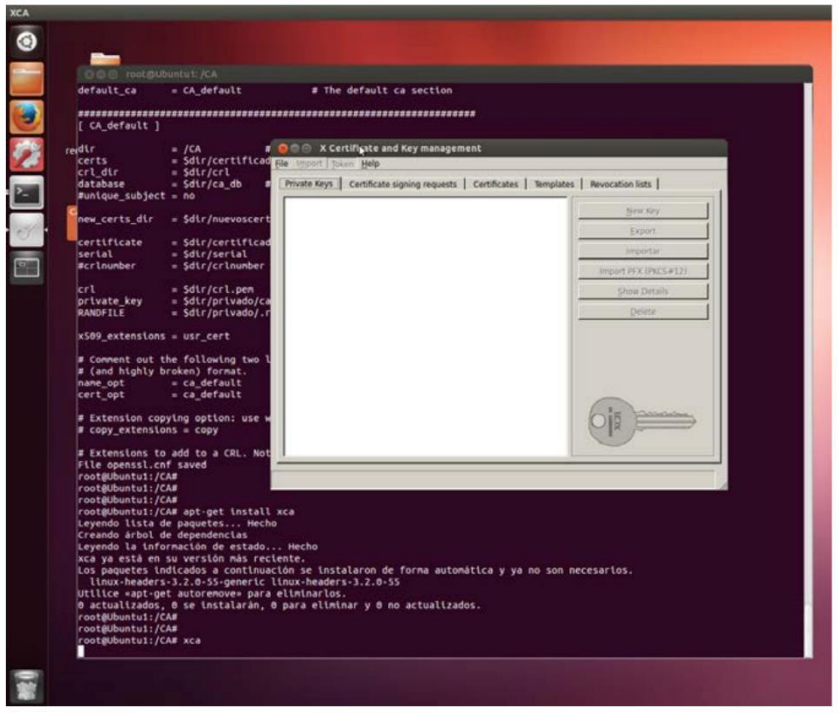
Ejercicio

- Como usamos un directorio propio para la CA, se debe modificar el archivo /CA/openssl.cnf para que las siguientes líneas queden así:

- ##
- [CA_default]
- dir = /CA
- certs = $dir/certificados
- crl_dir = $dir/crl
- database = $dir/ca_db
- #unique_subject = no
- new_certs_dir = $dir/nuevoscertificados
- certificate = $dir/certificados/ca.pem
- serial = $dir/serial #crlnumber = $dir/crlnumber
- crl = $dir/crl.pem
- private_key = $dir/privado/ca.key RANDFILE = $dir/privado/.rand
- x509_extensions = usr_cert

Ejercicio

- Una vez tenemos ya la estructura y la configuración podemos instalar si no lo tenemos el programa grafico XCA, que nos servirá para gestionar la CA Desde el usuario root llamamos a la aplicación XCA una vez instalada con el gestor de paquetes de Linux, si es que no lo está (apt-get install xca)



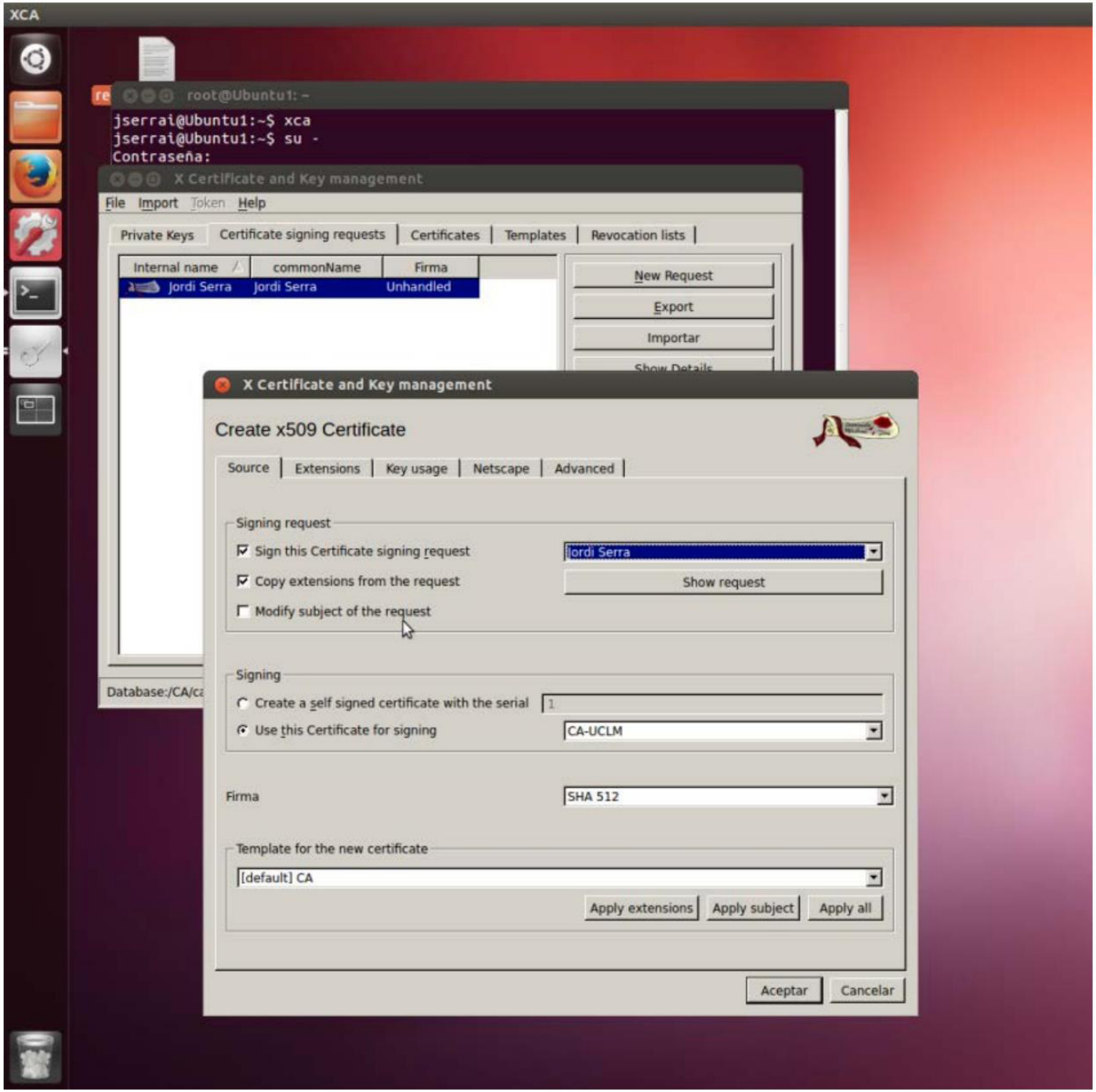
Ejercicio

- Así, otra vez como usuario root, abriremos la base de datos, desde el menú FILE, e importaremos la clave privada, en private Keys y pública en Certificates, que se han generado en el paso anterior para la CA que estamos creando. En este caso se puede ver ya que la aplicación da un aviso de que la clave pública está firmada por sí misma, por lo que la marca como no segura. Si fuese necesario para una empresa real se enviaría esta clave a firmar a una CA de nivel superior para dar confianza delegada a nuestra CA que estamos implantando.

Ejercicio

- Solo queda importar las peticiones de firma que se hagan llegar, en este caso hay que firmar la que se ha generado en el primer paso desde el sistema Windows. Con importarla, seleccionarla y firmarla es suficiente. Podemos firmarla con una SHA512 para mayor seguridad, le podemos decir el tiempo por el que será válido el certificado, y el uso que se permite de éste.

-


XCA
root@Ubuntu1: ~
jserrai@Ubuntu1:~$ xca
jserrai@Ubuntu1:~$ su -
Contraseña:
X Certificate and Key management
File Import Token Help
Private Keys
Certificate signing requests
Certificates
Templates
Revocation lists
Internal name
commonName
Firma
Jordi Serra
Jordi Serra
Unhandled
New Request
Export
Importar
Show Details
Database:/CA/ca
X Certificate and Key management
Create x509 Certificate
Source
Extensions
Key usage
Netscape
Advanced
Signing request
Sign this Certificate signing request
Jordi Serra
Copy extensions from the request
Show request
Modify subject of the request
Signing
Create a self signed certificate with the serial
1
Use this Certificate for signing
CA-UCLM
Firma
SHA 512
Template for the new certificate
[default] CA
Apply extensions
Apply subject
Apply all
Aceptar
Cancelar

Ejercicio

- Ahora ya se pueden ver en Certificados la lista de los certificados que se han creado a partir de la CA que tenemos. Y podemos exportar entonces el certificado .crt que será el que está firmado por la entidad CA-UCLM que acabamos de crear.

- Crear un documento en PDF con todo el proceso de instalación de la CA descrito e implementado en vuestra compañía ficticia.
- Firmar ese fichero PDF con vuestro certificado creado a partir de esa CA.

Ejercicio

- PGP:
- https://ssd.eff.org/es/module/como-usar-pgp-para-windows-pc
- https://www.openpgp.org/software/