Nociones generales sobre Seguridad de la Información

Generalidades: ¿qué es seguridad?

- Podemos entender como seguridad una característica de cualquier sistema (informático o no) que nos indica que ese sistema está libre de todo peligro, daño o riesgo, y que es, en cierta manera, infalible.

- Como esta característica, particularizando para el caso de sistemas operativos o redes de computadores, es muy **difícil de conseguir** (según la mayoría de expertos, imposible), se suaviza la definición de seguridad y se pasa a hablar de fiabilidad (probabilidad de que un sistema se comporte tal y como se espera de él) más que de seguridad; por tanto, se habla de sistemas fiables en lugar de hacerlo de sistemas seguros.

Generalidades: Aspectos de la seguridad

- A grandes rasgos se entiende que mantener un sistema seguro (o fiable) consiste básicamente en garantizar tres aspectos:
  - confidencialidad,
  - integridad y
  - disponibilidad.

- La confidencialidad exige que los objetos de un sistema han de ser accedidos únicamente por elementos autorizados a ello, y que esos elementos autorizados no van a convertir esa información en disponible para otras entidades.

- La integridad significa que los objetos sólo pueden ser creados o modificados por elementos autorizados, y de una manera controlada.

- La disponibilidad indica que los objetos del sistema tienen que permanecer accesibles a elementos autorizados; es el contrario de la negación de servicio.

Generalidades: elementos de la seguridad

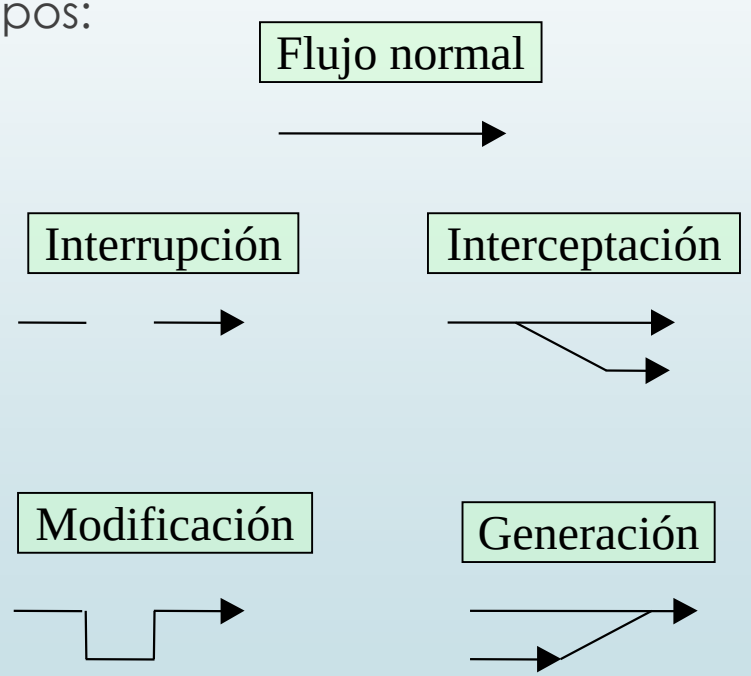
- Los tres elementos principales a proteger en cualquier sistema informático son:
  - el hardware,
  - el software y
  - los datos.

- Por hardware entendemos el conjunto formado por todos los elementos físicos de un sistema informático, como CPUs, terminales, cableado, medios de almacenamiento secundario o tarjetas de red.
- Por software entendemos el conjunto de programas lógicos que hacen funcional al hardware, tanto sistemas operativos como aplicaciones.
- Por datos entendemos el conjunto de información lógica que manejan el software y el hardware
  - (como por ejemplo paquetes que circulan por un cable de red o entradas de una base de datos).

Generalidades: amenazas a la seguridad

- Contra cualquiera de los tres elementos (pero principalmente sobre los datos) se pueden realizar multitud de ataques o, dicho de otra forma, están expuestos a diferentes amenazas.

- Generalmente, la clasificación más elemental de estas amenazas las divide en cuatro grandes grupos:
  - interrupción,
  - interceptación,
  - modificación
  - generación.



Generalidades: amenazas a la seguridad

- Un ataque se clasifica como:
  - Interrupción si hace que un objeto del sistema se pierda, quede inutilizable o no disponible.
  - Interceptación si un elemento no autorizado consigue un acceso a un determinado objeto del sistema.
  - Modificación si además de conseguir el acceso consigue modificar el objeto.
  - Generación o fabricación si se trata de una modificación destinada a conseguir un objeto similar al atacado de forma que sea difícil distinguir entre el objeto original y el fabricado.

Generalidades: amenazas a la seguridad

- Podemos clasificar a los elementos que potencialmente pueden amenazar a nuestro sistema en tres grupos:
  - Personas
  - Amenazas lógicas
  - Catástrofes

Generalidades: mecanismos de seguridad

- Los mecanismos de seguridad de un sistema se dividen en tres grandes grupos:
  - Prevención
  - Detección
  - Recuperación.

Prevención

- Los mecanismos de **prevención** son aquellos que aumentan la seguridad de un sistema durante el funcionamiento normal de éste, previniendo la ocurrencia de violaciones a la seguridad;
  - **por ejemplo**, el uso de **cifrado** en la transmisión de datos se puede considerar un mecanismo de este tipo, ya que evita que un posible atacante escuche las transmisiones de información que circulen por la red.

Detección

- Por mecanismos de **detección** se conoce a aquellos que se utilizan para detectar violaciones de la seguridad o intentos de violación;
  - ejemplos de estos mecanismos son los programas de **auditoría**.

Generalidades: mecanismos de seguridad

Recuperación

- Finalmente, los mecanismos de **recuperación** son aquellos que se aplican cuando una violación del sistema se ha detectado, para retornar a éste a su funcionamiento correcto;
  - ejemplos de estos mecanismos son la utilización de copias de seguridad o el hardware adicional.

- Dentro de este último grupo de mecanismos de seguridad encontramos un subgrupo denominado mecanismos de análisis forense, cuyo objetivo no es simplemente retornar al sistema a su modo de trabajo normal, sino averiguar el alcance de la violación, las actividades de un intruso en el sistema, y la puerta utilizada para entrar; de esta forma se previenen ataques posteriores y se detectan ataques a otros sistemas de nuestra red.

Generalidades: mecanismos de prevención

- Aunque los tres tipos de mecanismos son importantes para la seguridad de un sistema, se debe enfatizar en el uso de mecanismos de prevención y de detección;
  - la máxima popular "más vale prevenir que curar" se puede aplicar a la seguridad informática.

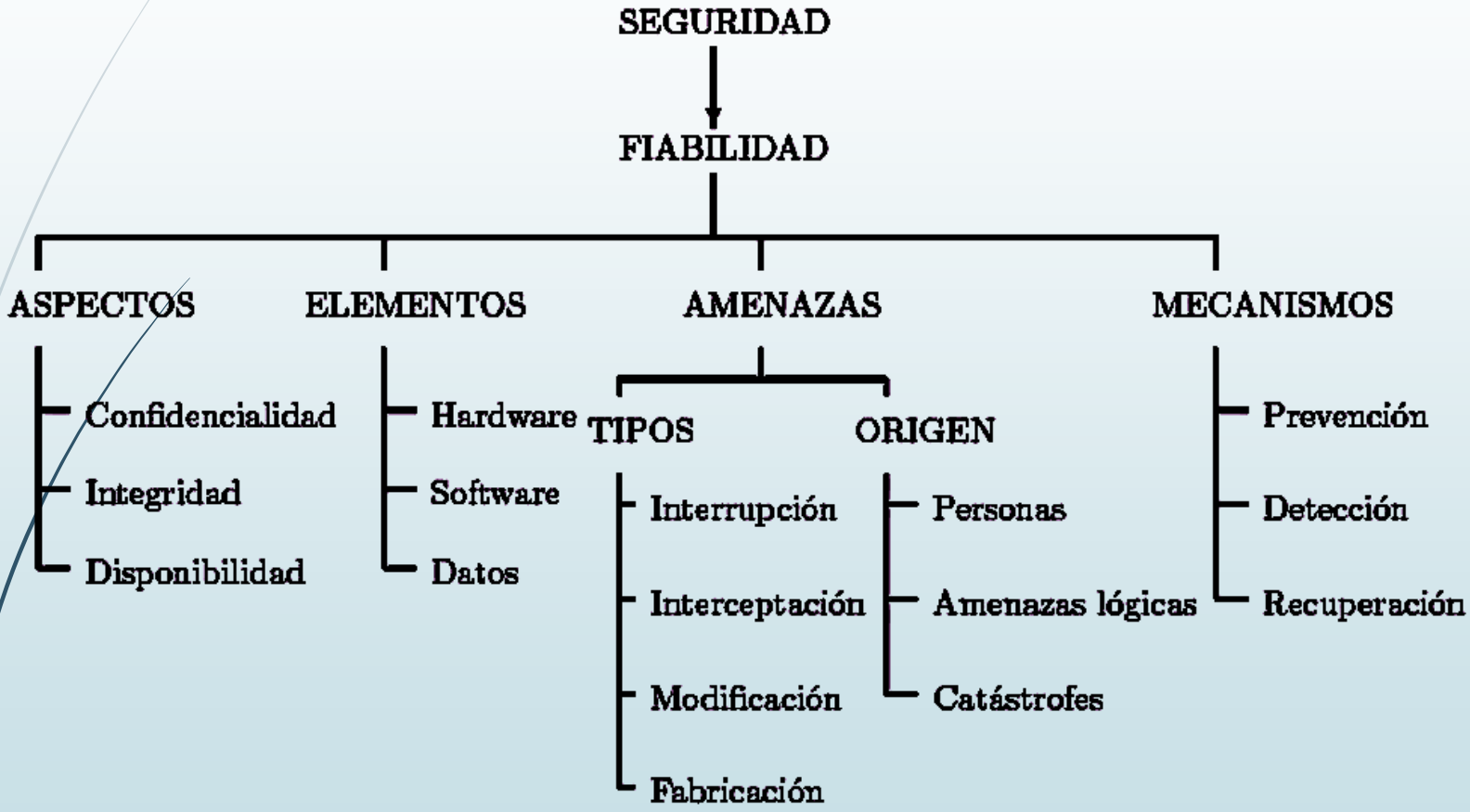
- Los mecanismos de prevención más habituales en redes son los siguientes:
  - Mecanismos de autenticación e identificación
  - Mecanismos de control de acceso
  - Mecanismos de seguridad en las comunicaciones

Generalidades: mecanismos de prevención

Mecanismos de seguridad en las comunicaciones

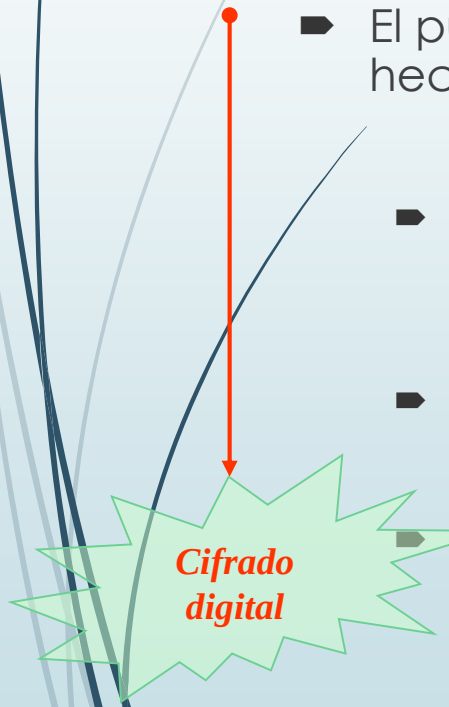
- Es especialmente importante para la seguridad de un sistema proteger la confidencialidad y la integridad de la información que se transmite a través de la red.
- Para garantizar la seguridad en las comunicaciones, se debe hacer uso de mecanismos que se basan en la **Criptografía** (cifrado de clave pública, de clave secreta, firmas digitales, …)
- No hay una única herramienta global criptográfica si no que existen distintas técnicas para lograr distintos objetivos, como
  - cifrar mensajes,
  - intercambio seguro de claves criptográficas,
  - mantener y asegurar la integridad de un mensaje así como
  - garantizar la autenticidad de un mensaje recibido.

Resumen nociones generales



Introducción histórica

- 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 C. Shannon sobre la Teoría de la Información.
  - En 1974 aparece el estándar de cifrado DES.
  - En el año 1976 se publica el estudio realizado por W. Diffie y M. Hellman sobre la aplicación de funciones matemáticas de un solo sentido a un modelo de cifra, denominado cifrado con clave pública.



https://www.youtube.com/watch?v=a99Qorfotv4

Terminología

- Aunque nos hemos referido ya a algunos términos, daremos a continuación un pequeño vocabulario específico.

Cifrar-Descifrar

- Cuando deseamos enviar un mensaje de manera que resulte incompresible para aquellas personas a las que no va dirigido utilizando determinada técnica, diremos que ***ciframos*** el mensaje. El destinatario del mismo debe ***descifrarlo*** utilizando esa misma técnica.

Terminología

Texto en claro-Criptograma

- Un mensaje que se desea cifrar es llamado ***texto en claro*** o ***texto plano***, una vez cifrado se dice que es un ***criptograma***.
  - Para cifrar un texto en claro normalmente se suele dividir en bloques de una longitud preestablecida, que a veces consisten en un sólo carácter. Así, cada bloque m de texto plano es convertido en un bloque de texto cifrado c.
  - Denotaremos este proceso con la letra E y lo representaremos como E(m)=c, por ejemplo en el método utilizado por Julio César

$$E(A)=D,\ E(B)=E,\ldots,E(X)=C$$

  - El proceso de descifrado lo expresaremos con D. En el cifrado de César, por ejemplo, denotaremos

$$D(D)=A,\ D(E)=B,\ldots,D(C)=X$$

- Por supuesto, resulta esencial que el proceso de descifrado sea opuesto al de cifrado, esto es

$$D(E(m))=m$$

Terminología

Criptosistema-Clave

- Una función cifradora E junto con su correspondiente función descifradora D es conocido como un ***criptosistema o sistema criptográfico***.
- El cifrado y descifrado de los mensajes regularmente requiere de una pieza especial para el conocimiento del criptosistema conocida como ***clave***.
  - Por ejemplo, en el método de J. César la clave es la cantidad con la que cada letra es desplazada a la derecha o izquierda dentro de la tabla

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	X
D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	X	A	B	C

  - Con ese número podemos conocer el orden para cifrar y descifrar los mensajes.

Terminología

Criptografía-Criptoanálisis

- La ciencia que estudia el diseño de criptosistemas es conocida como ***criptografía***.
- En la creación de un criptosistema, el ***criptógrafo*** debe tener como objetivo que el sistema sea lo más ***seguro*** posible, esto es, debe crear el sistema de manera que una persona no autorizada, que no conozca la clave, no pueda descifrar los mensajes.
- Las personas que intentan descifrar mensajes sin conocer la clave son conocidos como ***criptoanalistas*** y la ciencia que intenta romper los criptosistemas, descifrando en un tiempo razonable el contenido de un mensaje sin conocer la clave, es llamada ***criptoanálisis***.

Terminología

Criptología

- Los campos de la criptografía y el criptoanálisis, son conocidos en su conjunto como ***criptología***.
- En el análisis de la seguridad de un criptosistema frente al criptoanálisis el criptógrafo debe asumir que el criptoanalista tiene un gran conocimiento del sistema, de hecho éste puede conocer la forma de cifrar y descifrar y tener múltiples ejemplos de texto en claro y del correspondiente texto cifrado.

Terminología

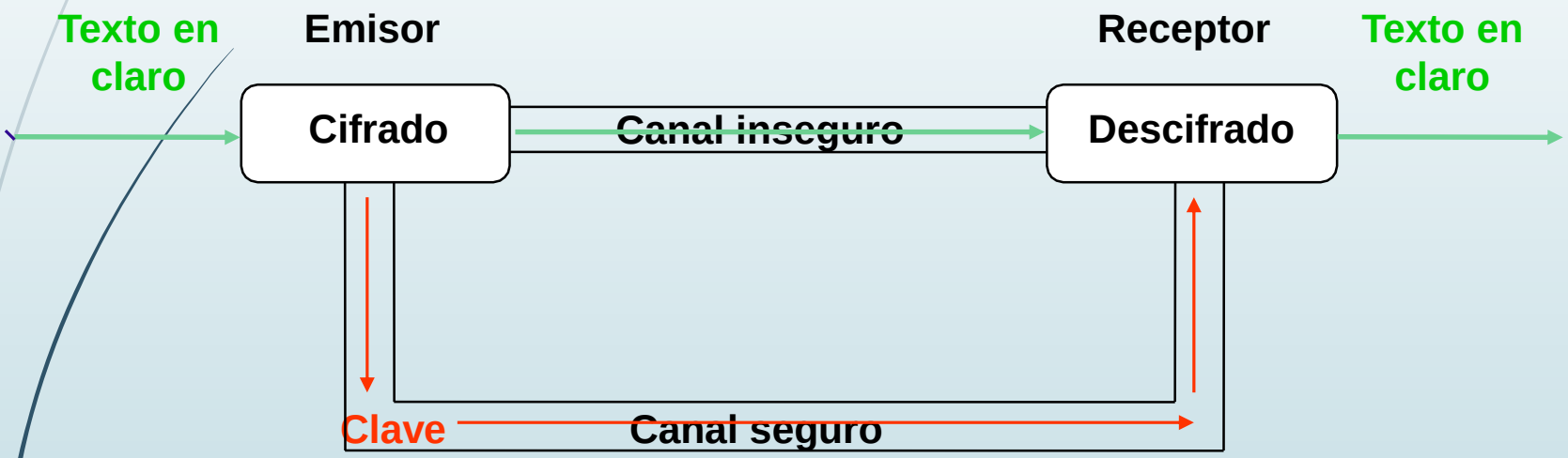
Espacio de claves

- El componente fundamental del criptosistema del que carece el criptoanalista es la clave.
  - El objetivo de este último es descubrir la clave correcta de entre el conjunto K de todas las posibles a utilizar, este conjunto es conocido como el ***espacio de claves***.

- El criptoanalista puede conocer, por ejemplo, que el criptosistema es el método de cifrado de César pero desconocer el número de posiciones que es desplazada cada letra.
  - Su meta es averiguar la clave utilizada dentro del espacio $K=\{1,2,...,21\}$.

Criptosistemas

Esquema de un sistema de cifrado



Clasificación

- Podemos realizar varias ***clasificaciones*** de los criptosistemas, atendiendo a diversos criterios.

Antiguedad

- **Clásicos**: Si su origen es anterior a la aparición del ordenador.
- **Modernos**: Se fundamentan en la informática. Necesitan ser actualizados (en muchos casos son desestimados) al ritmo de progreso de la potencia de cálculo de los ordenadores.

Igualdad entre las claves de cifrado y descifrado

- **Simétricos**: La clave de descifrado es la misma que la del cifrado o se puede obtener a partir de ella.
- **Asimétricos**: Las claves de cifrado y descifrado son distintas y no puede obtenerse una a partir de la otra, salvo que se disponga de alguna información adicional.

Clasificación

Forma de cifrar

- **Cifrado en bloque**: El mismo algoritmo de cifrado se aplica a un bloque de información (grupo de caracteres, número de bytes, etc.) repetidas veces, usando la misma clave.
- **Cifrado en flujo**: El algoritmo de cifrado se aplica a un elemento de información (carácter, bit) mediante un flujo de clave en teoría aleatoria y mayor que el mensaje.

Conocimiento de la clave

- **De clave privada o secreta**: Las claves de cifrado y descifrado sólo son conocidas por personal autorizado.
- **De clave pública**: El conocimiento de la clave de cifrado se puede hacer pública, pero la de descifrado sólo es conocida por el receptor del mensaje.

Números grandes

- Los actuales algoritmos criptográficos emplean claves con un elevado número de bits, y usualmente se mide su calidad por la cantidad de esfuerzo computacional que se necesita para romperlos.

- El tipo de ataque mas simple es la ***fuerza bruta***, que simplemente trata de ir probando una a una todas las claves.
  - Por ejemplo, el algoritmo DES tiene 2^{56} posibles claves.
  - ¿Cuanto tiempo nos llevaría probarlas todas si, por ejemplo, dispusiéramos de un computador capaz de probar un millón de claves por segundo?

    Tardaríamos más de 2200 años
  - Si la clave tuviera 128 bits, el tiempo requerido sería de 10^{24} años, que es aproximadamente cien billones de veces la edad del universo.

- Este dato nos debe disuadir de emplear mecanismos basados en la fuerza bruta para *reventar* claves de 128 bits.

Fundamentos de la seguridad de la información

- Tres son los pilares sobre los que descansa toda la teoría asociada a los criptosistemas:
  - Teoría de la Información
    - Estudios realizados por Claude Shannon
  - Teoría de los Números
    - Estudio de las matemáticas discretas
  - Teoría de la Complejidad de los Algoritmos
    - Clasificación de los problemas

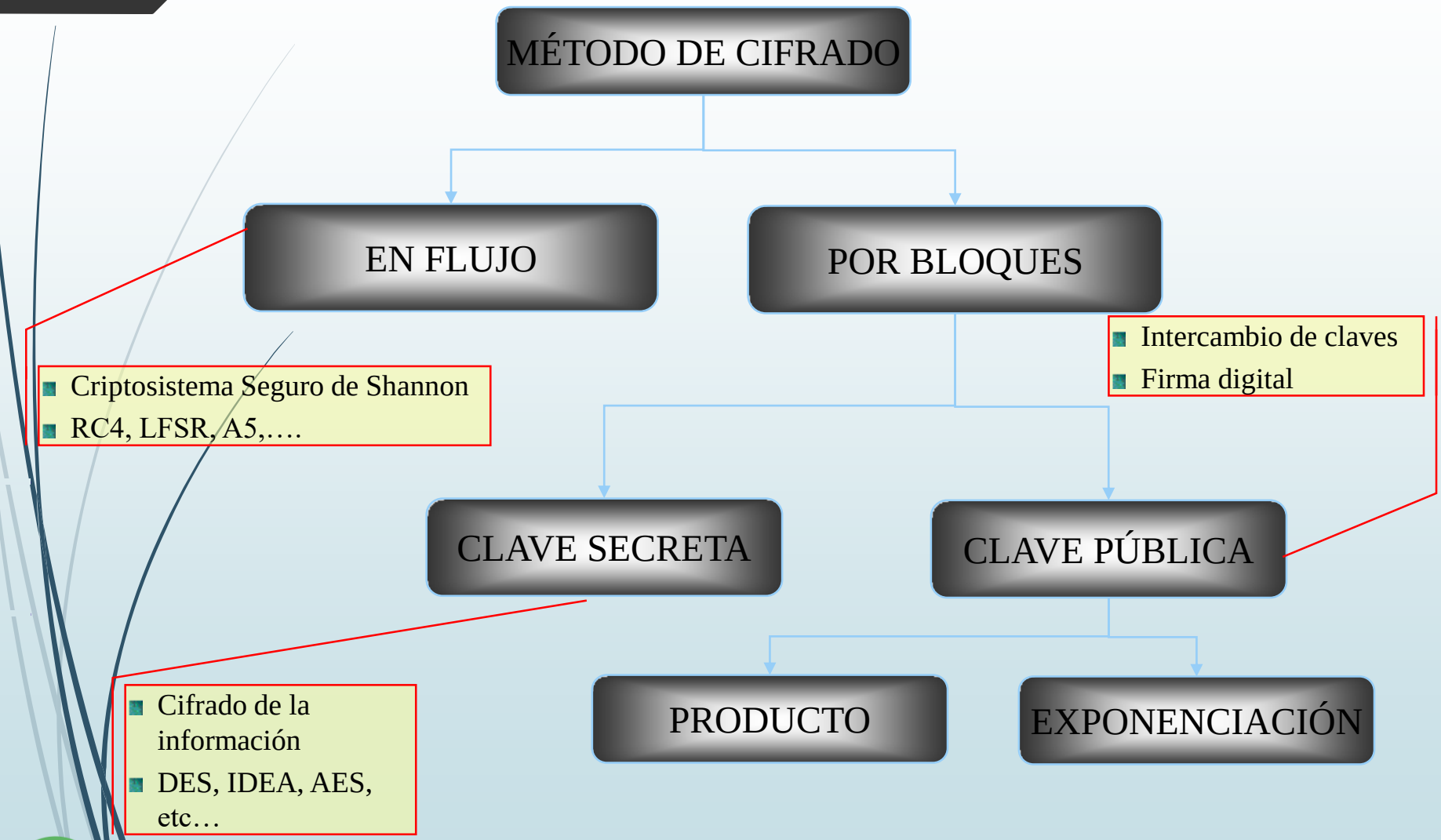
Introducción a criptosistemas modernos

- Cuando se habla de **criptografía moderna** o criptosistemas modernos, en contraposición a los estudiados anteriormente y denominados clásicos, se hace referencia a aquellos sistemas de cifrado, bien de clave secreta, bien de clave pública, que
  - por una parte, **realizan el cifrado en bits**, orientado a todos los caracteres del sistema de representación numérica que utilicen, es decir ASCII o ANSI, sin que necesariamente el módulo de trabajo deba coincidir con el número de elementos del alfabeto o código utilizado

Por lo general, mediante una operación algebraica dentro de un cuerpo finito

  - y por otra, **basan su seguridad en la fortaleza de la clave** y no en el secreto de un algoritmo de cifrado.

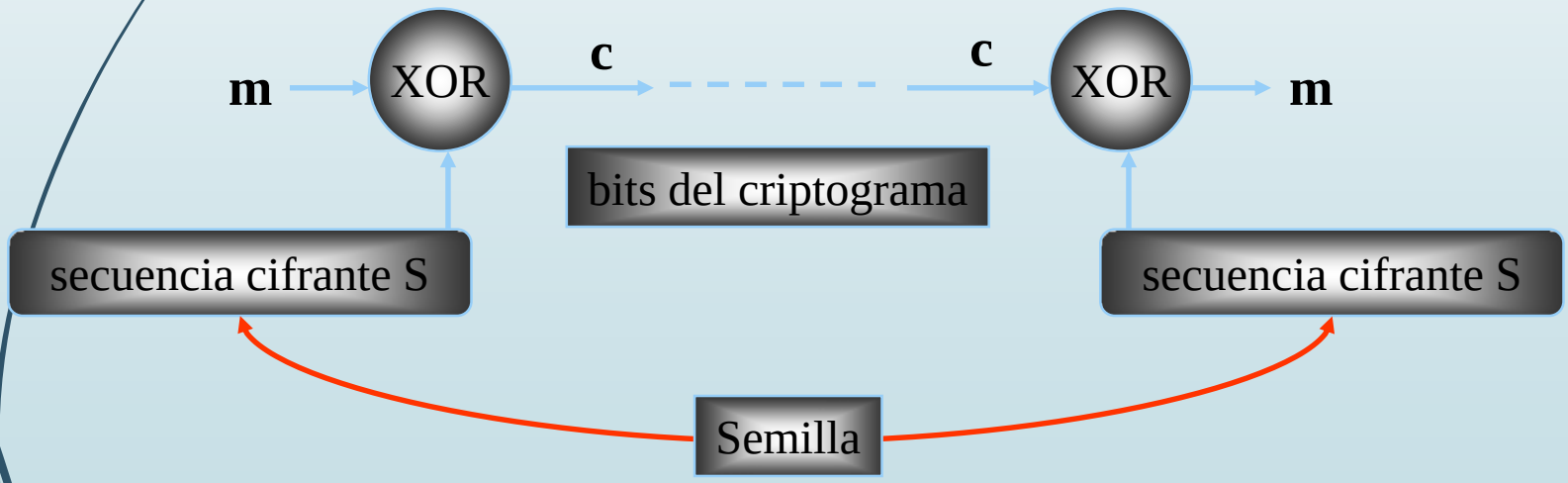
Clasificación de los criptosistemas modernos



Introducción al cifrado en flujo

- El mensaje en claro se **leerá bit a bit**.
- Se realizará una operación de cifrado (normalmente la función **XOR**) con una **secuencia cifrante** de bits S que debe cumplir ciertas condiciones:
  - Un período muy alto.
  - Aleatoriedad en sus propiedades.

Esquema



Cifrado en bloque con clave secreta

- El texto en claro se **fracciona en bloques** de un tamaño constante y se aplica, con la misma **clave**, el **algoritmo** a cada **bloque** de forma independiente.

- El cifrado se realiza con una **clave secreta**

- Algunos de los más conocidos son
  - **DES** (*Data Encryiption Standard)* por su uso en aplicaciones bancarias.
  - **IDEA** (*International Data Encryption Algorithm)* por su uso en el cifrado de correo electrónico.
  - **AES** (*Advanced Encryption Standard*): nuevo estándar avanzado de cifrado del Instituto Nacional de Estándares y Tecnología (National Institute of Standards and Technology, NIST) ***(Rijndael)***

Cifrado en bloque con clave secreta

Algoritmos más utilizados

- AES
- DES (Triple DES)
- Serpent
- Blowfish, Twofish

Otros algoritmos

- Camellia
- CAST-128
- IDEA
- RC2, RC5, RC6
- SEED
- ARIA
- Skipjack
- TEA, XTEA

Modos de cifrado en bloque

- Se ha convenido en denominar al uso directo de un cifrador en bloque como modo de «Libro Electrónico de Códigos» (*Electronic Codebook*, **ECB**).

- Entre otros, el NIST (U.S. National Institute for Standards and Technology) recomienda **cuatro modos de uso**, tanto para AES como para cualquier cifrado en bloque:
  - Encadenamiento de bloques cifrados (*Cipher Block Chaining*,**CBC**)
  - Realimentación del texto cifrado (*Cipher Feedback*, **CFB**)
  - Realimentación de la salida (*Output Feedback*, **OFB**)
  - Modo contador (*Counter mode*, **CTR**)

Cifrado en bloque con clave secreta

Debilidades del cifrado con clave secreta

- **Mala gestión de claves:** para una red de n usuarios el número de claves es $O(n^2)$. Para un número grande de usuarios es intratable👎.
- **Mala distribución de claves:** no existe posibilidad de enviar, de forma segura, una clave a través de un medio inseguro👎.
- **No tiene firma digital:** aunque es posible autentificar el texto en claro mediante una marca, no es posible firmarlo digitalmente👎.

Mala gestión de claves 👎.
Mala distribución de claves 👎.
No tiene firma digital 👎.

¿Tiene algo de bueno el cifrado en bloque con clave secreta?

Sí, la velocidad de cifrado es muy alta 👍

Cifrado con clave pública

- Uno de los mayores inconvenientes que presentan los sistemas de clave secreta cuando existe un gran número de usuarios es que cada par de ellos debe poseer su clave secreta, lo que conlleva gran dificultad en la distribución segura de esas claves.

- Otro gran inconveniente es que no se puede firmar digitalmente el mensaje, con lo que el receptor del mismo no puede estar seguro de su autenticidad (o sea, no puede estar seguro de que quien dice que le envía el mensaje sea realmente quien lo ha hecho)

Cifrado con clave pública

- Un criptosistema de clave pública permite la comunicación cifrada entre dos usuarios sin necesidad de compartir una clave, así como la firma digital de los mensajes.

- Los algoritmos de clave pública, o algoritmos asimétricos, han demostrado su interés para ser empleados en redes de comunicación inseguras (Internet)

- Introducidos por Whitfield Diffie y Martin Hellman a mediados de los años 70, su novedad fundamental con respecto a la criptografía simétrica o de clave secreta es que las claves no son únicas, sino que forman pares con la propiedad de que una es capaz de descifrar lo que ha sido cifrado por la otra.

Cifrado con clave pública

- Los algoritmos asimétricos, por lo general, basan su seguridad en el enfrentamiento del atacante a problemas matemáticos que requieren mucha computación.
- Emplean, generalmente, longitudes de clave mucho mayores que los simétricos.
  - Por ejemplo, mientras que para algoritmos simétricos se considera segura una clave de 128 bits, para algoritmos asimétricos (si exceptuamos aquellos basados en curvas elípticas) se recomiendan claves de al menos 1024 bits.
- La complejidad de cálculo que comportan los hace considerablemente más lentos que los algoritmos de cifrado simétricos.
- ***En la práctica los métodos asimétricos se emplean únicamente para cifrar la clave de sesión (simétrica) de cada mensaje o transacción particular***.

Cifrado con clave pública

- Fijadas las **dos claves** de cada usuario como clave **pública** y clave **privada**.
- La clave **privada** deberá ser **custodiada por el usuario** y es imprescindible que se mantenga en secreto.
- La clave **pública**, por el contrario, se **publicará** junto con la **identidad del usuario**.
- Así, cuando se quiera **enviar** un **mensaje** seguro a un usuario se hará uso de la **clave pública**, que se utilizará para **cifrar** el mensaje a enviar.

Cifrado con clave pública

- El resultado de esta operación será el **texto cifrado** que sólo el propietario de la clave privada correspondiente a esa clave pública **podrá descifrar**.

- Estas claves tienen características matemáticas especiales.
  - Se **generan siempre a la vez**, por parejas, estando cada una de ellas **ligada intrínsecamente a la otra**,
  - de tal forma que si **dos claves públicas son diferentes**, entonces sus **claves privadas asociadas también lo son** y viceversa.

Cifrado con clave pública

- Los algoritmos de clave pública están **basados** en **funciones matemáticas fáciles de resolver en un sentido**, pero muy **complicadas de realizar en sentido inverso**, salvo que se conozca alguna trampa.

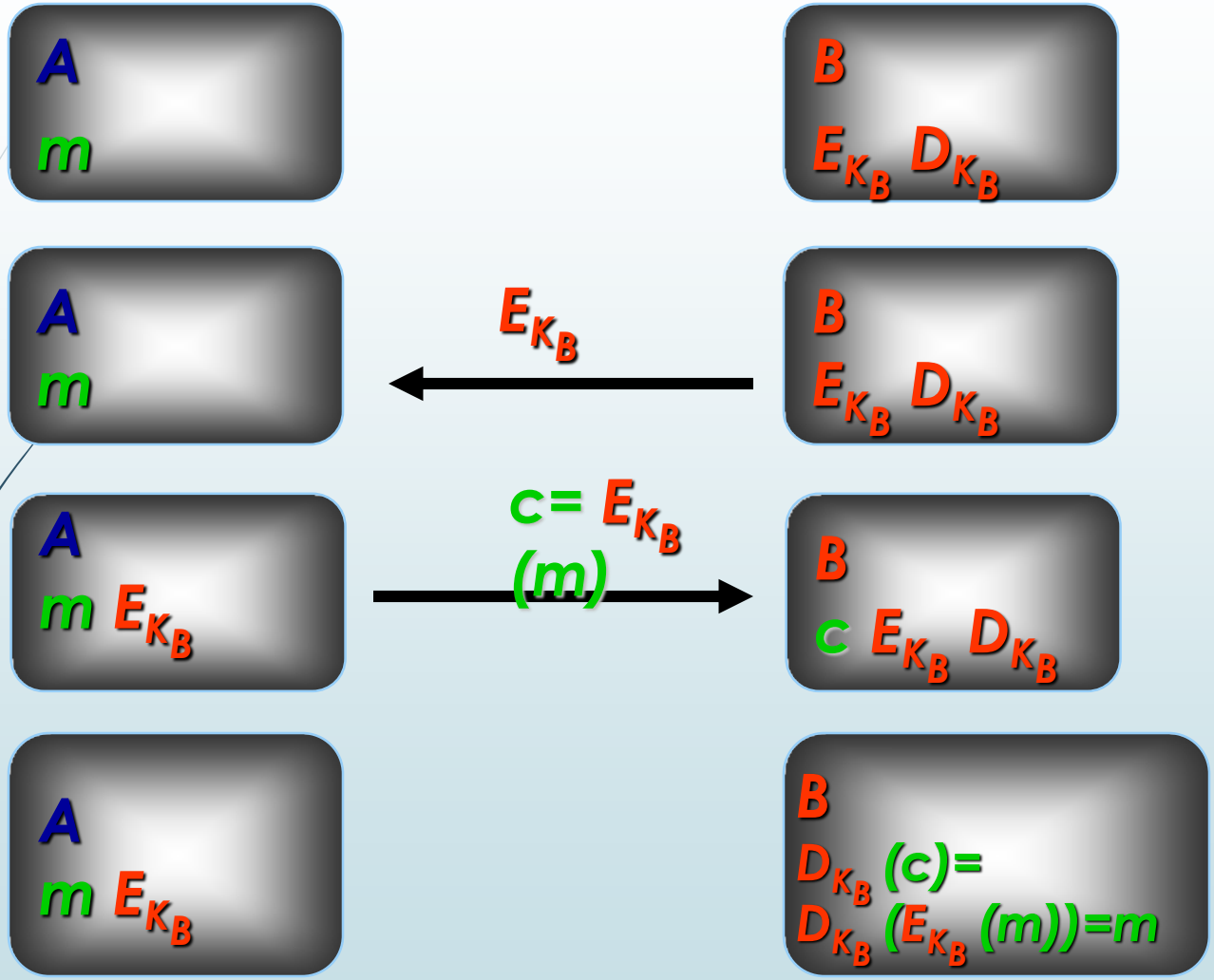
- Ambas claves, **pública y privada, están relacionadas matemáticamente**, pero esta relación debe ser lo suficientemente compleja como para que resulte **muy difícil obtener** una a partir de la otra.

- Este es el motivo por el que normalmente estas claves no las elige el usuario, si no que **lo hace un algoritmo** específico para ello.

Cifrado con clave pública

- En este tipo de ciptosistemas, para enviar un mensaje con seguridad, el emisor (A) cifra el mismo con la clave pública del receptor (B) y lo envía por el medio inseguro.
- Este mensaje está totalmente protegido en su viaje, ya que sólo se puede descifrar con la clave privada correspondiente, conocida solamente por B.
- Al llegar el mensaje cifrado a su destino, el receptor usa su clave privada para obtener el mensaje en claro.
- El algoritmo más utilizado es RSA

Cifrado con clave pública



Firma digital

- Estamos pasando, gracias a Internet y a la aparición de las nuevas tecnologías, de un sistema tradicional de realizar operaciones comerciales a uno nuevo que las efectúa mediante métodos electrónicos.

- Resulta necesario contar con **técnicas electrónicas** que **suplanten la tradicional firma autógrafa** y dar así validez a documentos electrónicos.

- La firma digital se justifica desde el momento en que los contratos, las transacciones económicas, las compras, etc. se realizan on-line

Firma digital

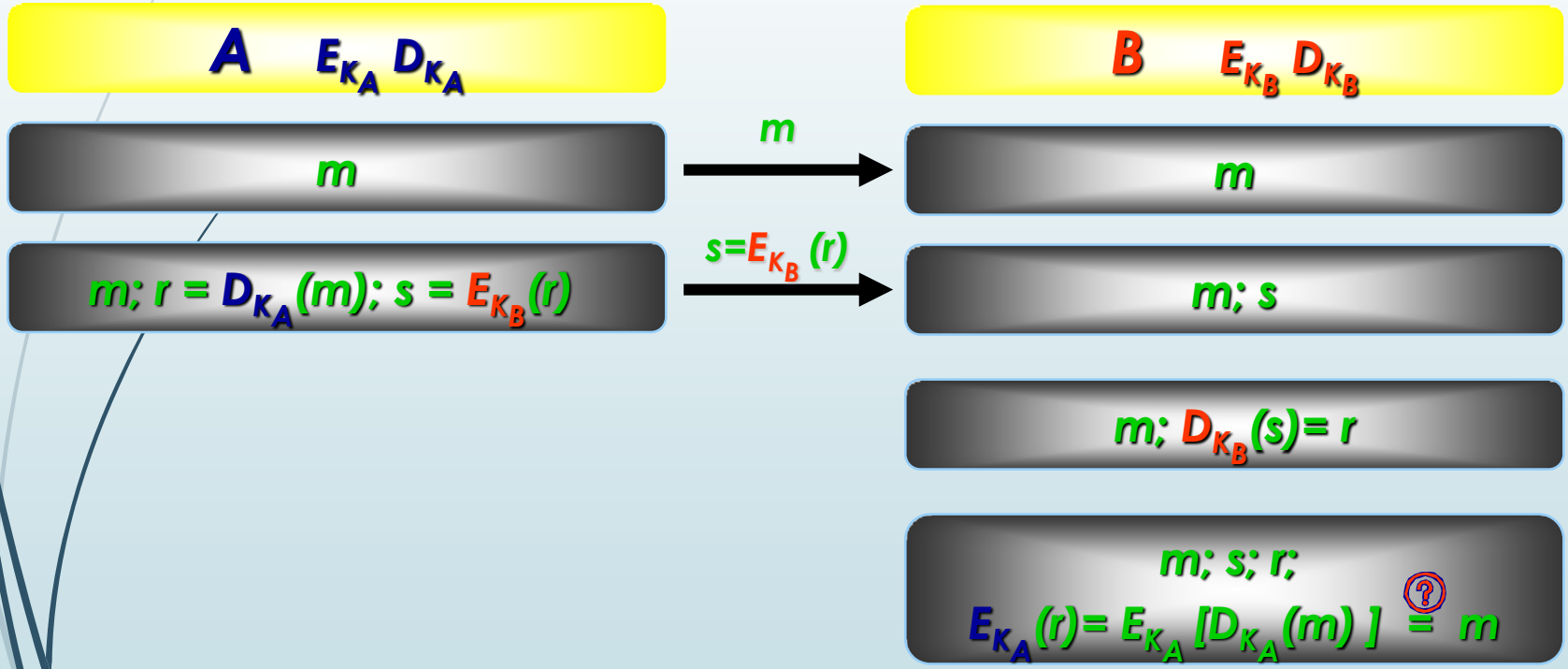
- Dos problemas aquejan a estos documentos electrónicos:
- ***Confidencialidad***: Capacidad de mantener un documento electrónico inaccesible a todos, excepto a una lista determinada de personas.
  - Se resuelve con métodos de cifrado.
- ***Autenticidad***: Capacidad de averiguar de forma segura e irrevocable la procedencia del mensaje.
  - Se resuelve con técnicas como la firma digital.

Firma digital

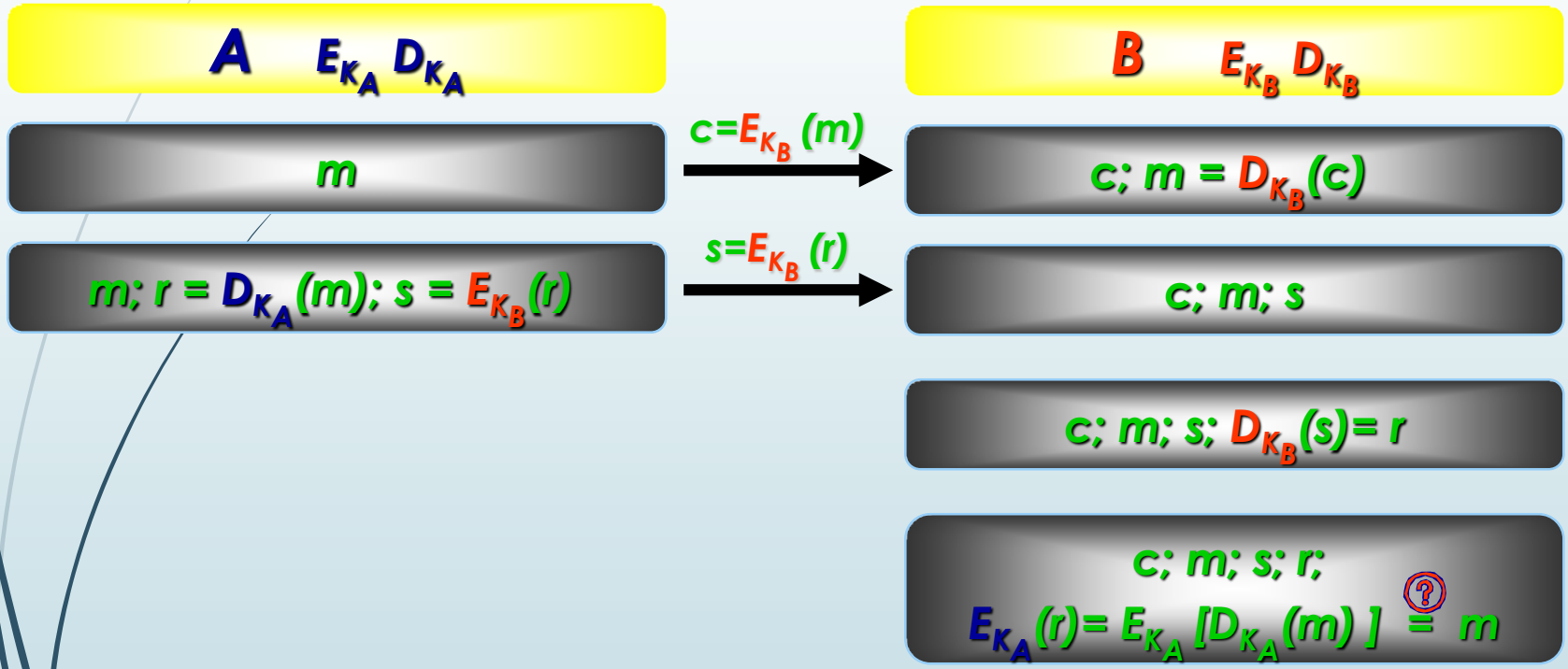
- Si A desea firmar digitalmente el mensaje m, envía el mensaje cifrado $c=E_{k_B}(m)$ al usuario B y para firmarlo,
  - en primer lugar calcula su **rúbrica** cifrando el mensaje a enviar con su clave secreta, $r= D_{k_A}(m)$, y
  - a continuación determina su firma para el mensaje m, cifrando con la clave pública de B esa rúbrica, $s= E_{k_B}(r)= E_{k_B}[D_{k_A}(m)]$.

- B verifica la firma digital de A determinando,
  - en primer lugar, la rúbrica de A, $D_{k_B}(s) = D_{k_B}[E_{k_B}(r)]$, y
  - a continuación comprobando que $E_{k_A}(r) = E_{k_A}[D_{k_A}(m)]$ coincide con el mensaje m.

Esquema de firma digital sin cifrado



Esquema de firma digital con cifrado



Funciones hash

- Los criptosistemas de **clave pública**, por lo general, son mucho **más lentos** que los de **clave secreta**.
- Los esquemas de **firma digital**, también suelen ser **muy lentos** y, en ocasiones, la longitud de la firma suele ser similar o mayor que el propio mensaje que se firma.
- La necesidad de firmar los mensajes y el hecho no deseable de que la longitud de la firma sea extensa, hace pensar en la conveniencia de buscar una solución a ese problema.
  - La solución consiste en utilizar unas funciones llamadas **hash** (picadillo, resumen).

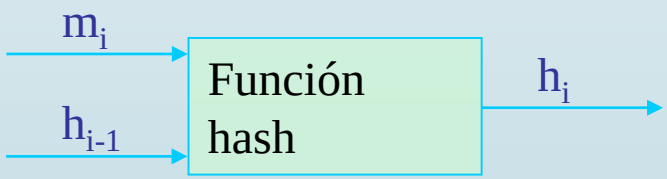
Funciones hash

- En lugar de firmar digitalmente el mensaje completo, se **firma digitalmente un resumen** o hash de dicho mensaje, representado por sólo una centena de bits.

- Las funciones hash también se utilizan para verificar la **integridad** de los mensajes, ya que si se produce un cambio, el resumen que se genera es diferente.

Funciones hash

- En general, las funciones hash se basan en la idea de funciones de **compresión**, que dan como resultado **bloques de longitud n** a **partir** de **bloques de longitud m**.
- Estas funciones se encadenan de forma iterativa, haciendo que la entrada en el paso i sea función del i-ésimo bloque del mensaje y de la salida del paso i-1
- Frecuentemente, se suele incluir en alguno de los bloques del mensaje m (al principio o al final), información sobre la longitud total del mensaje.



Algunas funciones hash

- **MD5**: Ron Rivest 1992. Mejoras al MD4 y MD2 (1990), es más lento pero con mayor nivel de seguridad que estas. Resumen de 128 bits.

- **SHA-1**: Del NIST, National Institute of Standards and Technology, 1994. Similar a MD5 pero con resumen de 160 bits. El NIST ha publicado una revisión del estándar, FIPS 180-2 , en la que se añaden 3 algoritmos de hash adicionales:
  - SHA-256, SHA-384, SHA-512,

  diseñados para que sean compatibles con el estándar de cifrado AES.

http://unaaldia.hispasec.com/2017/02/demostracion-practica-de-colision-en.html

https://shattered.io/

- **SHA-2**: es un conjunto de funciones hash criptográficas (SHA-224, SHA-256, SHA-384, SHA-512) diseñadas por la Agencia de Seguridad Nacional (NSA) y publicadas en 2001 por el Instituto Nacional de Estándares y Tecnología (NIST) como un Estándar Federal de Procesamiento de la Información (FIPS). Incluye un significante número de cambios respecto a su predecesor, SHA-1.

Algunas funciones hash : SHA-3

- **SHA-3 (Keccak)**: La competición para seleccionar el algoritmo criptográfico de hash para reemplazar a SHA-1 y a SHA-2, lanzada por el NIST EN 2007, finalizó con la elección oficial por parte del equipo NIST de Keccak como el nuevo algoritmo SHA-3. Tras seis años de proceso, se tomó una decisión y el algoritmo Keccak fue elegido como el nuevo SHA-3. Keccak es obra de Guido Bertoni, Joan Daemen, Michaël Peeter y Gilles Van Assche trabajadores de STMicroelectronics y NXP Semiconductors.

NIST Cyptographic Hash Algorithm Competition (SHA-3)

http://csrc.nist.gov/groups/ST/hash/sha-3/index.html

Función ganadora del concurso SHA-3: Keccak

Algunas funciones hash

- **Píldoras formativas THOTH - Criptored**

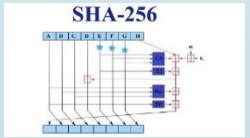
http://www.criptored.upm.es/thoth/#

43. ¿Qué son y para qué sirven las funciones hash?

44. ¿Cómo funciona el hash MD5?

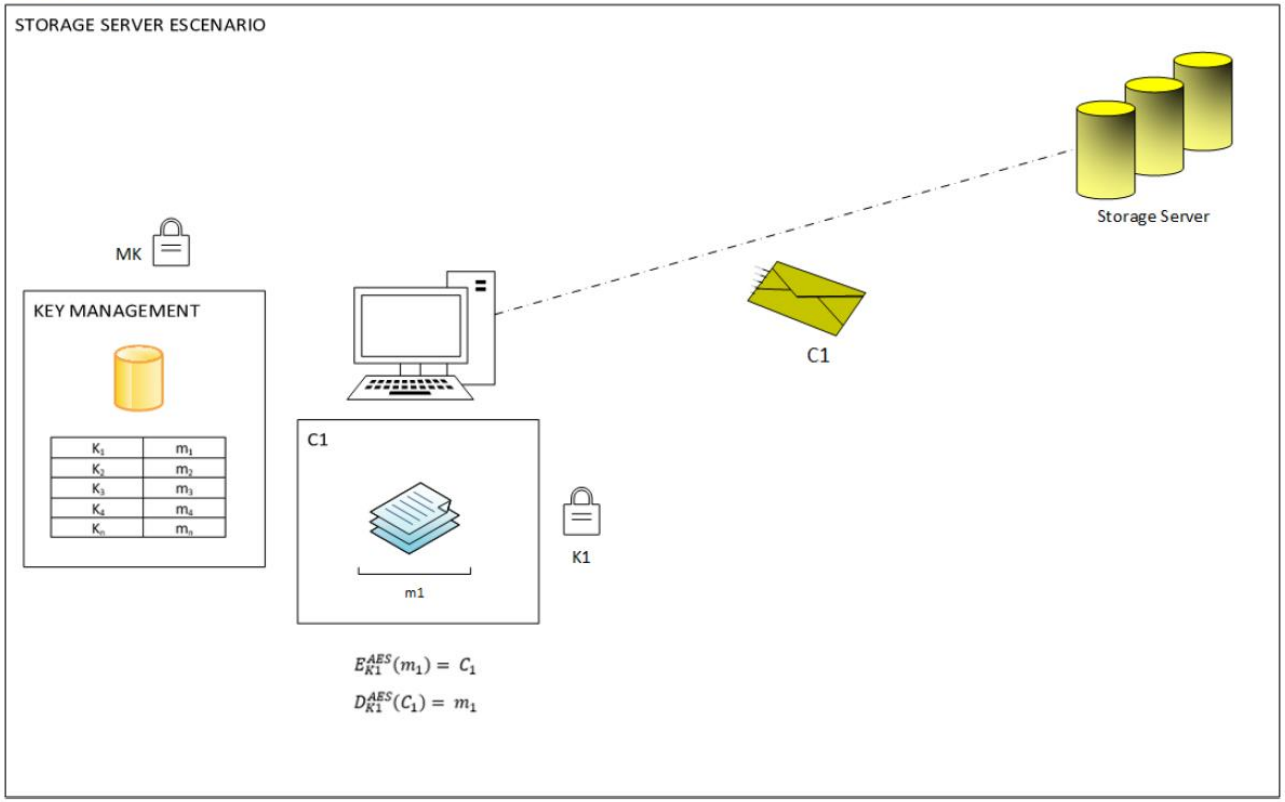
45. ¿Cómo funciona el hash SHA-1?

46. ¿Qué son SHA-2 y SHA-3?



Ejemplo de 1ª memoria

Escenario



Notación

K_n = *clave de sesión n*

$E_{K_n}^{AES}$ = *función de encriptado*

$D_{K_n}^{AES}$ = *función de desencriptado*

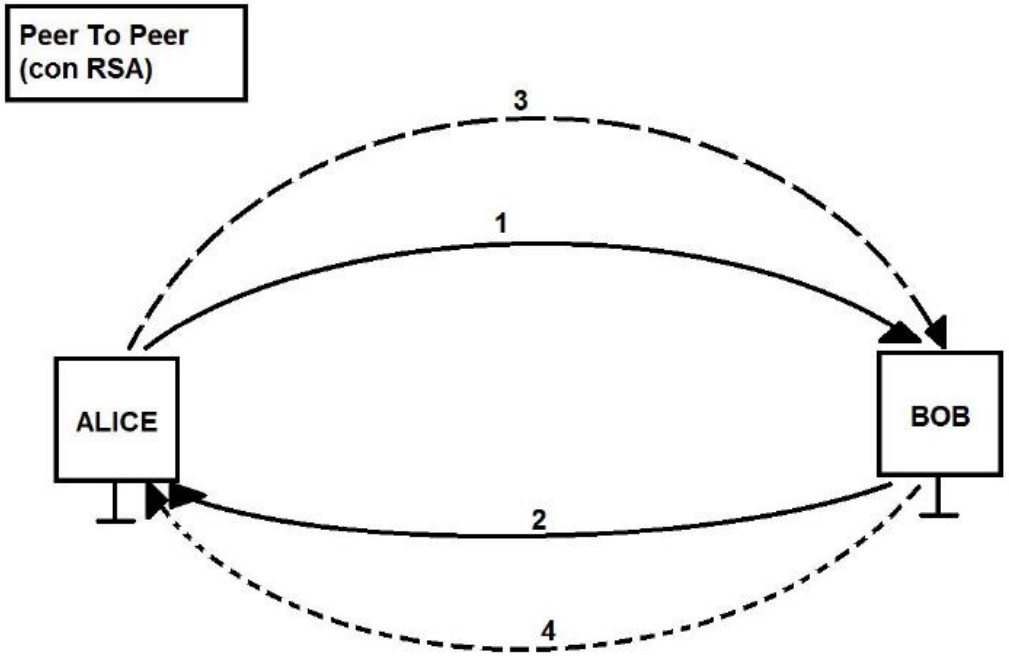
m_n = *mensaje en claro n*

C_n = *criptograma n*

MK = *clave maestra*

Ejemplo de 1ª memoria

PROCESO DE COMUNICACIÓN



0 - Ambas partes generan previamente su par de claves pública / privada con RSA de 2048 bits.

1 - Alice envia a BOB su clave pública K_A

2 - Bob hace lo mismo con K_B.

3 - Alice usa K_B para cifrar la semilla de 128 bits ({M}K_B) que se usa para generar la clave privada.

4 - A partir de ahora, Bob hace uso de la clave privada para cifrar los mensajes con Alice.

Ejemplo de 1ª memoria: PROGRAMACION AES

- java.security.Key
- javax.crypto.Cypher
- javax.crypto.KeyGenerator
- Javax.crypto.spec.SecretKeySpec

Ejemplo de 1ª memoria: PROGRAMACION RSA

- java.security.KeyPair
- java.security.KeyPairGenerator
- java.security.PrivateKey
- java.security.PublicKey

Ejemplo de memoria

Hyper Desktop

Servidor en la nube con cifrado.

Funcionamiento

- Hyper Desktop es un servidor en la nube, donde se cifra tanto en la conexión entre el servidor y el cliente como en los datos almacenados en el servidor. Por lo que el servidor no podrá leer tus archivos a pesar de tener acceso a estos.

- El cliente se conectará a través de una aplicación de escritorio escrita en Java.

- El servidor almacenará, además de los ficheros, algunos datos necesarios para la aplicación. Este también estará escrito en Java.

Datos

- ID: Cada archivo tendrá una ID propia para poder identificarlo, que estará relacionado con el usuario al que le pertenece.
- User: Se utilizará para identificar al usuario.
- Password: Será necesaria una contraseña de 16 caracteres mínimo, con al menos una mayúscula y un número.
- Archivos cifrados: Los archivos a subir serán cifrados en el cliente, tras lo que serán subidos al servidor. Por lo que este en ningún momento tendrá acceso a estos.
- Una serie de hashes y claves que se explicarán detalladamente a continuación.

Algoritmos

- Algoritmo simétrico: El algoritmo simétrico elegido es AES con una longitud de clave de 128KB. Debido a que al ser el estándar y además seguro nos ha parecido la mejor opción.

- Algoritmo asimétrico: El algoritmo asimétrico elegido es RSA, con una longitud de clave de 1024KB. Que aunque no sea el estándar de forma oficial, si que es el estándar de facto y también seguro a día de hoy.

- Hash: El algoritmo hash elegido ha sido SHA-3, con una longitud de 512KB Debido a que un algoritmo reciente que se considera un buen hash para la actualidad.

Notación cifrado

- **Usuario U**
  - Clave privada del usuario U: KU_{priv}^{RSA}
  - Clave pública del usuario U: KU_{pub}^{RSA}
- **Usuario V**
  - Clave privada del usuario V: KV_{priv}^{RSA}
  - Clave pública del usuario V: KV_{pub}^{RSA}
- **Fichero File**
  - Clave del fichero: K_{File}^{AES}
  - Contraseña del usuario U: Upass

Registro

- El usuario que desee registrarse, necesitará:
  - Un Nick
  - Una contraseña de 16 caracteres al menos.
  - Un par de claves pública/privada que será generada por el programa del cliente.

- Los pasos a seguir son estos:
  - El usuario solicita al servidor registrarse.
  - Una vez el servidor haya autorizado al usuario su registro, este generará un par de claves pública y privada.
  - El usuario envía la clave pública tal cual.
  - El usuario envía la clave privada encriptada con su contraseña: $E_{Upass}^{AES}\ (KU_{priv}^{RSA})$
  - El usuario envía un hash de su contraseña: $Hash(Upass)$

- De esta forma, el servidor dispondrá de todo lo necesario para que el usuario opere pero al mismo tiempo no dispondrá de la contraseña en sí que es el desencadenante de todos los servicios que ofrece.

Login

- Se envía al servidor el nombre de usuario y el hash de la contraseña:
  - $Hash(Upass)$
- Una vez el servidor compruebe que los datos son correctos, este le devolverá su clave privada la cual está encriptada con su contraseña:
  - $E_{Upass}^{AES}(KU_{priv}^{RSA})$
- Y una vez en local lo desencripta con su Upass para obtener su clave:
  - $D_{Upass}^{AES}\left[E_{Upass}^{AES}\left(KU_{priv}^{RSA}\right)\right] = KU_{priv}^{RSA}$

Subir un archivo

- Ahora el usuario quiere subir un archivo, en primer lugar, genera una clave AES que será única para ese archivo:
  - K_{File}^{AES}
- Y encripta el fichero *File* con dicha clave:
  - $E_{K_{File}^{AES}}^{AES}(File)$
- Además, encripta la clave del fichero con su clave pública:
  - $E_{KU_{pub}^{RSA}}^{RSA}\left(K_{File}^{AES}\right)$
- Y se suben los dos pares que hemos generado. De esta forma el servidor ha recibido los archivos y las claves de estos, pero no conoce su contenido.

Descargar archivo privado

- El usuario que desee descargar un archivo se descargará el archivo encriptado y la contraseña de este encriptado:
  - $E^{RSA}_{KU^{RSA}_{pub}}(K^{AES}_{File})$
  - $E^{AES}_{K^{AES}_{File}}(File)$
- Una vez descargado, desencripta la clave del fichero:
  - $D^{RSA}_{KU^{RSA}_{priv}}\left[E^{RSA}_{KU^{RSA}_{pub}}(K^{AES}_{File})\right] = K^{AES}_{File}$
- Y una vez tenga la clave del fichero, lo decifra:
  - $D^{AES}_{K^{AES}_{File}}\left[E^{AES}_{K^{AES}_{File}}(File)\right] = File$
- Y ya habrá conseguido el archivo que deseaba.

Descargar archivo público

- El usuario introducirá el ID del archivo que desee descargar y se bajará esto:
  - $E^{AES}_{K^{AES}_{File}}(File)$
- Introduce la clave AES que se le ha compartido previamente y ya obtendrá el archivo que deseaba:
  - $D^{AES}_{K^{AES}_{File}}\left[E^{AES}_{K^{AES}_{File}}(File)\right] = File$

Compartir archivos

- Escenario:
  - el usuario U tiene un fichero File ya subido previamente al servidor y este desea compartirlo con el usuario V.

- U se descarga del servidor la clave del archivo y recibe esto:
  - $E^{RSA}_{KU^{RSA}_{pub}}\left(K^{AES}_{File}\right)$

- Como U dispone de su clave privada RSA, lo descifra y obtiene la clave del archivo:
  - $D^{RSA}_{KU^{RSA}_{priv}}\left[E^{RSA}_{KU^{RSA}_{pub}}\left(K^{AES}_{File}\right)\right] = K^{AES}_{File}$

Compartir archivos

- Una vez ha obtenido K_{File}^{AES}, le solicita al servidor la clave pública del usuario V y cifra la clave del fichero con la clave pública de V:
  - $E_{KV_{pub}^{RSA}}^{RSA}(K_{File}^{AES})$

  y lo sube al servidor.

- Cuando el usuario V desee descargar dicho archivo, se descargará la clave que se ha generado a partir de su clave pública y este será capaz de conseguir la clave del archivo:
  - $D_{KV_{priv}^{RSA}}^{RSA}\left[E_{KV_{pub}^{RSA}}^{RSA}(K_{File}^{AES})\right] = K_{File}^{AES}$

Librerías

- JavaFX: Librería de Oracle para la creación de entornos gráficos en Java.

- Bouncy Castle: Librería opensource disponible tanto en C# y Java para criptografía con más de 15 años de desarrollo. En este caso utilizaremos la versión de Java para nuestro proyecto.