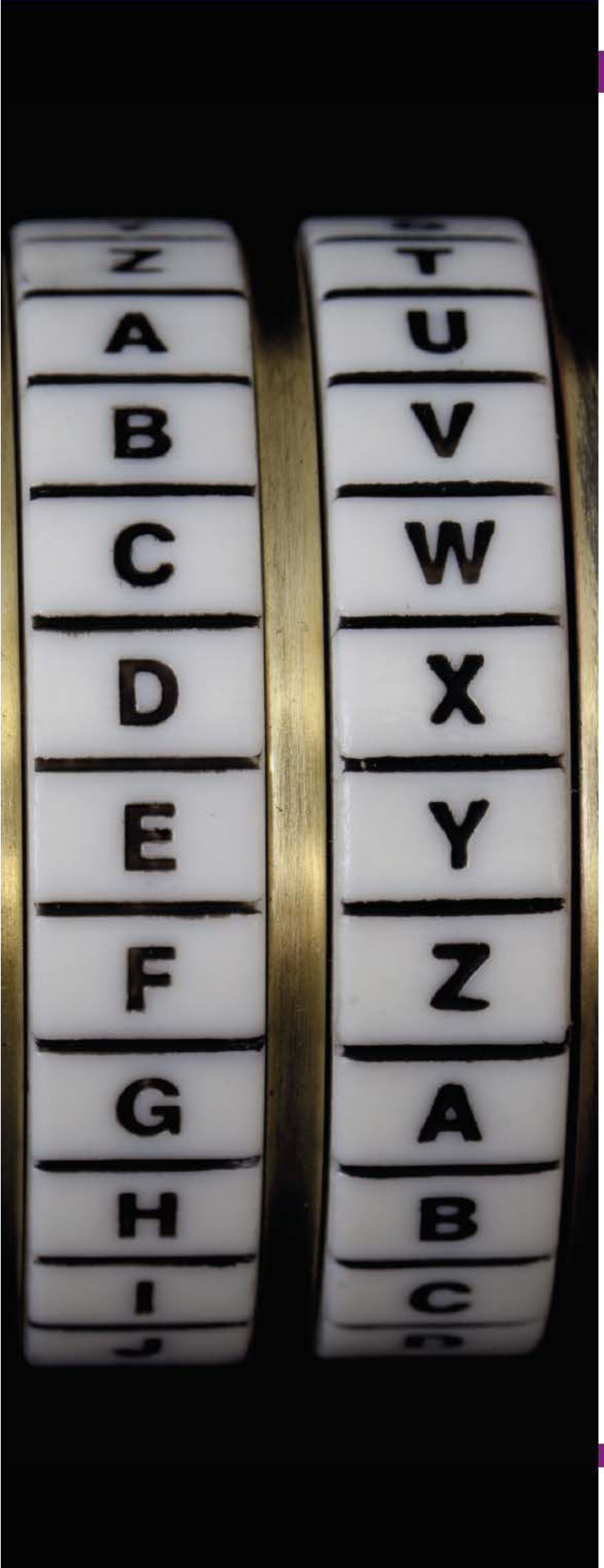


unidad 4

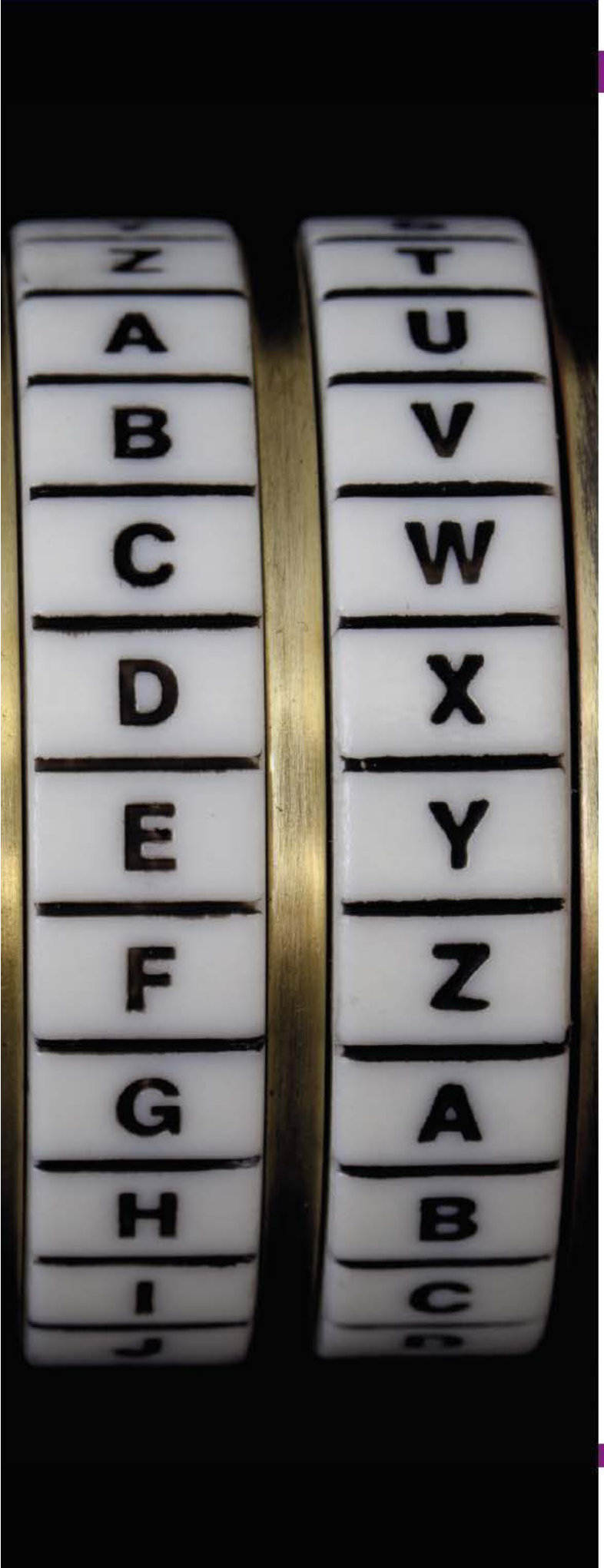
Criptografía

SUMARIO

- Introducción a la criptografía
- Cifrado de clave simétrica
- Cifrado de clave asimétrica
- Algoritmos de cifrado *hash*
- Criptosistemas híbridos

OBJETIVOS

- Conocer qué es la criptografía y para qué se utiliza.
- Distinguir los tipos de sistemas de cifrados utilizados en la criptografía.
- Describir las ventajas e inconvenientes de los criptosistemas.
- Conocer el algoritmo de cifrado *hash*.
- Apreciar las ventajas de los criptosistemas híbridos.



unidad 4

Criptografía

SUMARIO

- Introducción a la criptografía
- Cifrado de clave simétrica
- Cifrado de clave asimétrica
- Algoritmos de cifrado *hash*
- Criptosistemas híbridos

OBJETIVOS

- Conocer qué es la criptografía y para qué se utiliza.
- Distinguir los tipos de sistemas de cifrados utilizados en la criptografía.
- Describir las ventajas e inconvenientes de los criptosistemas.
- Conocer el algoritmo de cifrado *hash*.
- Apreciar las ventajas de los criptosistemas híbridos.

1 >> Introducción a la criptografía

Desde que el ordenador pasó a formar parte de nuestras vidas almacenando nuestros datos personales y convirtiéndose en una de las principales herramientas de comunicación, mediante el uso de Internet, ha ido creciendo la conciencia de que nuestros datos están expuestos a posibles intromisiones por parte de otras personas que quisieran apoderarse de ellos.

Esto ha hecho que, cada vez más, vayamos tomando conciencia de los peligros que supone dejar desprotegidos y a merced de posibles intrusos estos datos. La peor consecuencia de que alguien entre en nuestros equipos o intercepte nuestras comunicaciones es que se lleve datos importantes, como claves bancarias, números de tarjetas de crédito, etc., que pongan en riesgo nuestra integridad o permitan la suplantación de nuestra identidad.

Cuando se habla de medidas de seguridad informática, lo primero que nos viene a la cabeza son las medidas destinadas a impedir infecciones o accesos no autorizados en los equipos informáticos. Pero hay otro tipo de medidas que se pueden tomar respecto de los propios datos y son las consistentes en hacer que estos datos sean indescifrables por personas que accedan a ellos indebidamente.

Por ello, vamos a ver en qué consisten todas estas medidas: qué es un certificado digital, qué significa encriptar o cifrar un mensaje, qué significa el símbolo del candado en las comunicaciones por Internet, qué diferencia hay entre utilizar HTTP o HTTPS a la hora de navegar por Internet, etc.

1.1 > Definiciones

Esta materia utiliza una terminología específica con la que debemos familiarizarnos:

- **Criptología**: proviene del griego *krypto*, "oculto", y *logos*, "estudio". Se trata del estudio de los criptosistemas. Sus áreas principales de estudio son, entre otros, la criptografía y el criptoanálisis:
  - **Criptografía**: proviene del griego *krypto*, "oculto", y *graphos*, "escribir"; es decir, significa "escritura oculta". El diccionario de la RAE lo define como "el arte de escribir con clave secreta o de un modo enigmático". La criptografía no pretende ocultar un mensaje, sino únicamente su significado, a través de la codificación.
  - **Criptoanálisis**: es la ciencia que se ocupa de descifrar criptogramas rompiendo la clave utilizada para descubrir el contenido del mensaje. Es el reverso de la criptografía.
- **Criptosistema**: según el Centro Criptológico Nacional (CCN), es el conjunto de claves y equipos de cifra que, utilizados coordinadamente, ofrecen un medio para cifrar y descifrar.

Relacionando todos estos conceptos, podemos decir que la criptografía está integrada por las técnicas utilizadas para, utilizando una clave, convertir un mensaje inteligible (llamado texto nativo) en otro (texto cifrado), cuyo contenido solo puede ser comprendido por quienes conozcan la clave. Los algoritmos de cifrado son el método utilizado para ocultar el contenido del mensaje y el criptosistema es el conjunto de equipos y claves usados para cifrarlo.

Vocabulario

Cifrar: transcribir, utilizando una clave, un mensaje cuyo contenido se quiere ocultar.

Clave: conjunto de signos utilizados para la transmisión de un mensaje privado cuyo contenido se quiere ocultar.

Orígenes de la criptografía

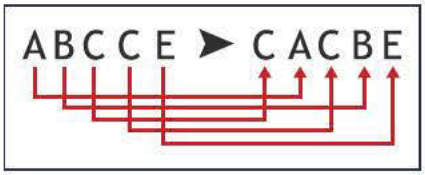
El primer método de criptografía conocido fue la escitala, utilizado en Esparta en el siglo V a. de C., que utilizaba técnicas de transposición.

Los sistemas evolucionaron hacia el uso de la sustitución de caracteres, como por ejemplo en el cifrado César, utilizado en Roma por Julio César.

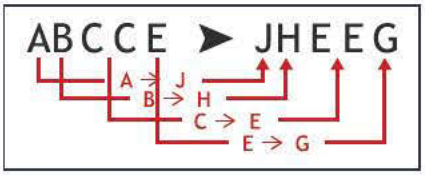
El uso de cifras y claves no es nuevo, ni ha venido de la mano de las nuevas tecnologías. Ya en la antigua Grecia, cuando se quería ocultar un mensaje, se utilizaba la **esteganografía**, que era una técnica consistente en ocultar el mensaje en sí mismo (por ejemplo, utilizando tinta invisible). Este sistema no ofrecía gran seguridad, ya que cualquiera que consiguiera acceder al mensaje podría ver su contenido. Por ello, comenzaron a utilizarse los primeros métodos de criptografía. La criptografía clásica se basaba métodos como el intercambio de letras, el ocultamiento del mensaje dentro de otro, etc. La criptografía moderna se basa en el uso de las matemáticas y en la utilización de mecanismos de cifrado (máquinas de cifrado o, actualmente, ordenadores).

Los sistemas criptográficos se basan en dos técnicas:

- **Transposición:** los signos o símbolos del mensaje original se cambian de posición.
- **Permutación o sustitución:** los signos o símbolos del mensaje original son sustituidos por otros.



4.1. Transposición.



4.2. Sustitución.

Máquina Enigma

La máquina Enigma se usó en la Segunda Guerra Mundial por las fuerzas militares de Alemania para cifrar los mensajes.

Usaba un mecanismo de cifrado rotatorio que le permitía tanto cifrar como descifrar. Su sistema de cifrado fue finalmente descubierto.

A lo largo de la historia, por tanto, las claves para mantener oculto un mensaje eran la técnica utilizada y el algoritmo empleado, que únicamente debían ser conocidos por el emisor y el receptor del mensaje. Si se daba esta premisa, el mensaje resultaría indescifrable; ahora bien, el problema de todos estos sistemas es que en toda lengua existen una serie de patrones (distribución de los espacios entre palabras, letras que se repiten con más frecuencia, etc.) a partir de los cuales es posible deducir el algoritmo, con lo cual se pierde la privacidad del mensaje.

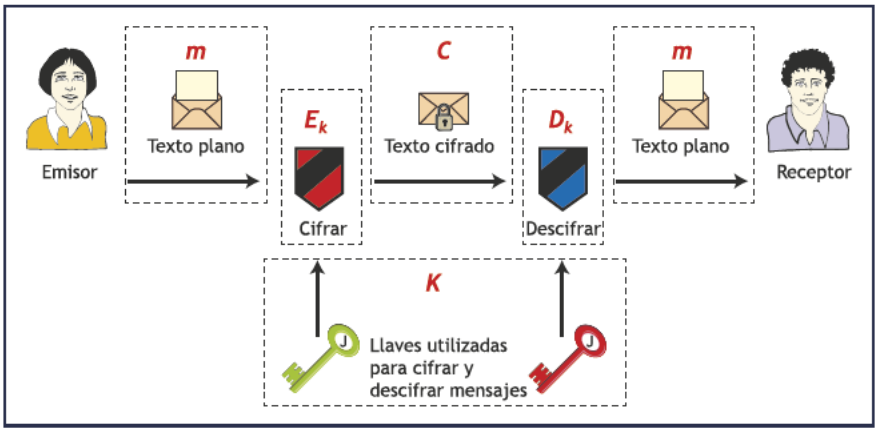
Por ello, la gran novedad de los criptosistemas modernos ha sido la introducción de métodos en los que el algoritmo es públicamente conocido y el secreto está en la clave que se utiliza como base para el cifrado. También es posible que el sistema simultanee ambas operaciones (algoritmo y clave desconocidos).

1.2 > Elementos de un criptosistema

Los criptosistemas están compuestos por los siguientes elementos:

- **Mensajes sin cifrar, texto plano o texto nativo** *(m)*: son los documentos originales sin haber sido cifrados.
- **Mensajes cifrados** *(C)* **o criptogramas.**
- **Conjunto de claves** *(K)*: son los datos o llaves que permiten cifrar los mensajes.
- **Transformaciones de cifrado** *(E)*: existe una transformación diferente para cada valor de la clave *k*.
- **Transformaciones de descifrado** *(D)*.

Veamos cada elemento en un esquema para comprenderlo mejor:



4.3. Esquema de un criptosistema.

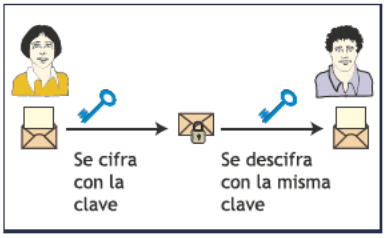
Todo criptosistema cumple la propiedad $D_k \cdot [E_k(m)] = m$.

Es decir, si se tiene un mensaje m y se cifra utilizando la clave k, se obtiene E_k. Si a ese mensaje cifrado se le aplica la transformación de descifrado para esa misma clave (D_k), se obtiene el mensaje original m.

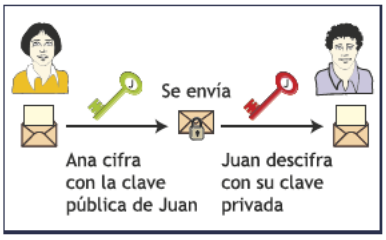
1.3 > Tipos de sistemas de cifrado

Hay dos tipos de sistemas de cifrado basados en claves, aunque existen también otros sistemas no basados en clave, de los que hablaremos más adelante:

- **Criptosistemas simétricos o de clave secreta.** En estos sistemas existe una única clave secreta que conocen y comparten emisor y receptor y que es utilizada para cifrar y descifrar el mensaje. La seguridad de este tipo de sistemas consiste en mantener dicha clave en secreto.
- **Criptosistemas asimétricos o de clave pública.** En este tipo de sistemas cada usuario crea un par de claves inversas: una privada y otra pública. Lo que el emisor cifra con una clave, el receptor lo descifra con la clave inversa. La seguridad de este tipo de sistemas radica en la dificultad de averiguar la clave privada a partir de la pública.



4.4. Sistema de clave simétrica.



4.5. Sistema de clave asimétrica.

Actividades propuestas

1·· Averigua qué es una escítala y explica en qué consistía. Consigue una imagen y referencias web.

2·· Investiga los orígenes del cifrado César. En qué consiste y qué vulnerabilidades presenta. Crea un mensaje cifrado utilizando este método y entrégaselo a un compañero para que lo descifre.

3·· Cifra el mensaje "Bienvenidos a la criptología" mediante la técnica de sustitución, usando la siguiente tabla de equivalencias:

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

2 >> Cifrado de clave simétrica

Como ya hemos avanzado en el punto anterior, en estos sistemas se utiliza la misma clave para cifrar y descifrar un mensaje. Dicha clave solo deberá ser conocida por el emisor y el receptor del mensaje y deberá mantenerse en secreto y a buen recaudo, pues en cuanto un atacante la descifre el criptosistema se ha roto.

Estos sistemas son mucho más rápidos y sencillos de implementar que los de clave pública (asimétrica) y resultan apropiados para el cifrado de grandes volúmenes de datos.

Hay dos grandes grupos de algoritmos de cifrado:

- **Cifradores de flujo**: cifran bit a bit.
- **Cifradores de bloque**: cifran un bloque de bits (habitualmente, cada bloque es de 64 bits) como una unidad.

Uno de los inconvenientes de este tipo de cifrado es que la clave debe ser conocida por el emisor y el receptor, quienes deben encontrar un modo seguro de comunicarla entre ambos.

El manejo de claves de este tipo de sistemas es costoso, ya que se necesita una clave por cada par de usuarios, lo que hace crecer exponencialmente el número de claves según se van incrementando los usuarios. El número de claves sería una combinación de m elementos (el número de usuarios) tomados de n en n (en este caso, de dos en dos, pues las claves las comparten siempre dos personas). Para calcular el número de claves necesarias aplicaríamos la siguiente fórmula:

$$C_m^n = \frac{m!}{n!(m-n)!}$$

Por ejemplo, para cinco usuarios harían falta:

$$C_5^2 = \frac{5!}{2!(5-2)!} = 10$$

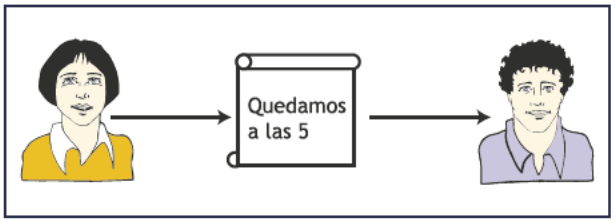
Hay dos sistemas para atacar un cifrado asimétrico:

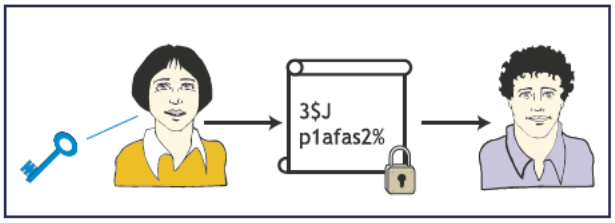
- **Criptoanálisis.** Se basa en la naturaleza del algoritmo y el conocimiento de algunas características del texto nativo y algunos pares texto nativo/texto cifrado. Este tipo de ataque aprovecha las características del algoritmo para intentar averiguar el texto originario o bien la clave secreta que se está utilizando. Este último sería el peor de los casos, ya que compromete todas las comunicaciones cifradas con esa clave. Es el caso del cifrado WEP en redes inalámbricas.
- **Método de fuerza bruta.** Consiste en probar todas y cada una de las posibles claves empleadas para cifrar el texto. Una vez que se haya encontrado la clave adecuada, ya se podrá descifrar el mensaje. La fortaleza del cifrado en casos de ataque por fuerza bruta depende de la complejidad de la clave que se haya empleado para cifrar. Si la clave es corta o fácilmente deducible, el sistema será vulnerable a este tipo de ataques aunque el algoritmo sea robusto.

Ejemplos

Utilización de clave simétrica

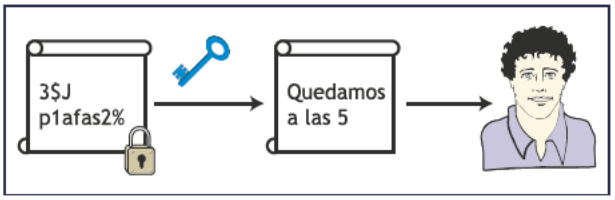
Vamos a ver en un ejemplo el funcionamiento de la clave simétrica. Ana quiere enviar un mensaje a Juan y, como pretende que su contenido sea secreto, lo cifra utilizando una clave simétrica. Esta clave debe ser conocida tanto por Ana como por Juan.



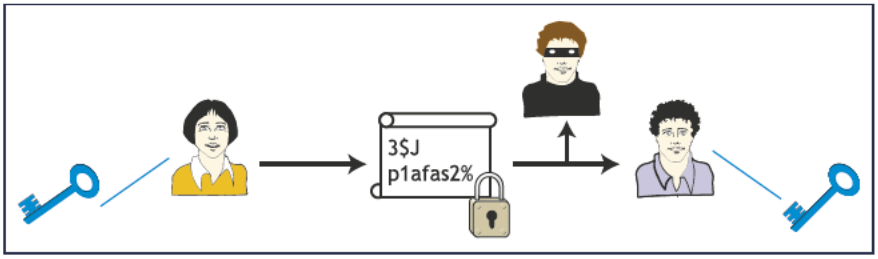


Una vez que Ana ha utilizado la clave, el mensaje será indescifrable para todo el mundo excepto para Juan, que conoce la clave. ¿Qué pasa si una tercera persona intercepta el mensaje?

El mensaje solo puede descifrarse si se conoce la clave. En este caso, solo lo puede descifrar Juan, que es quien conoce la clave.



Si un intruso interceptase el mensaje, solo podría descifrarlo si conociera la clave compartida por Ana y Juan. Eso podría suceder si Ana o Juan no la guardaran adecuadamente o si hubieran fijado una clave no muy compleja, fácil de ser averiguada por ataques de fuerza bruta.



Ventajas e inconvenientes de los sistemas de cifrado simétricos	
Ventajas	- Son rápidos y eficientes. - Resultan apropiados para el cifrado de grandes volúmenes de datos.
Inconvenientes	- Exigen una clave diferente por cada pareja de interlocutores (el espacio de claves se incrementa enormemente conforme aumentan los interlocutores). - Requiere un control estricto sobre el intercambio seguro de la clave entre el emisor y el receptor. - Son vulnerables a ataques por fuerza bruta, por lo que la fortaleza de la clave es fundamental.

Algoritmos de cifrado

Algunos de los algoritmos de cifrado más utilizados son los siguientes:

DES (*Data Encryption Standard*)

Nació en los años 70 y fue un algoritmo de cifrado muy utilizado hasta no hace mucho, no caracterizándose precisamente por su seguridad.

Utiliza cifrado por bloques con bloques de 64 bits, esto es, toma un texto plano de esa longitud y lo transforma, mediante una serie de operaciones, en texto cifrado de la misma longitud.

Se utilizan claves de 64 bits, de los cuales solo se utilizan 56, para realizar el cifrado de los bloques. El resto llevan información de paridad. Esta longitud tan corta se considera insuficiente para protegerse frente a ataques de fuerza bruta y es uno de los motivos por los que se considera inseguro, ya que estas claves se han llegado a romper en 24 horas. Aun así se sigue utilizando en las transacciones realizadas en cajeros automáticos.

AES (*Advanced Encryption Standard*)

El estándar de encriptación avanzada es uno de los algoritmos más populares de clave simétrica y, de hecho, remplazará al DES utilizado habitualmente.

Es rápido y eficiente y proporciona una encriptación segura utilizando un cifrado por bloques, con bloques de 128 bits y claves de 128, 192 o 256 bits. Se utiliza fundamentalmente en aplicaciones bancarias por Internet, comunicaciones inalámbricas, protección de datos en discos duros, etc.

RC5 (*Rivest Cipher*)

Diseñado por Ronald Rivest en 1994, se trata de un algoritmo que opera con un tamaño variable de bloques (32, 64 y 128 bits) y un número también variable de claves (entre 0 y 2040 bits).

Puede implementarse tanto por hardware como por software, consumiendo poca memoria y adaptándose a microprocesadores con distintos tamaños de palabra.

Aunque es muy seguro, se puede mejorar su efectividad modificando sus modos de operación: RC5-cifrador en bloque, RC5-CBC, RC5-CBC-relleno, RC5-CTS.

IDEA (*International Data Encryption Algorithm*)

Es un algoritmo descrito por primera vez en 1991 y propuesto para reemplazar al DES.

Este algoritmo trabaja con bloques de 64 bits y utiliza una clave de 128 bits, lo que hace que sea inmune al criptoanálisis diferencial. Además, el enorme número de posibles claves que hay que analizar hace que, con el estado actual de la computación, sea imposible de averiguar a través de ataques por fuerza bruta.

Al igual que ocurría con DES, en IDEA se usa el mismo algoritmo para el cifrado y el descifrado.

Triple DES

Triple DES aumenta la seguridad de DES ejecutando el algoritmo DES tres veces, cada una de ellas con una clave distinta. Aunque está siendo sustituido por AES, aún es el algoritmo utilizado por muchas tarjetas de crédito.

Algoritmo Rijndael

AES también es conocido como algoritmo Rijndael. El NIST (Instituto Nacional de Estándares y Tecnología de EEUU) convocó un concurso de algoritmos de encriptación para incorporarlos al estándar AES que estaban preparando. Los belgas Joan Daemen y Vincent Rijmen enviaron su algoritmo Rijndael, que resultó el ganador del concurso y fue adoptado por el estándar.

Sistemas de cifrado en redes inalámbricas

Los sistemas de cifrado más comunes en las redes inalámbricas como WEP, WPA-PSK, WPA2-PSK utilizan sistemas de clave simétrica como RC4 o AES.

Ejemplos

Criptografía de clave simétrica

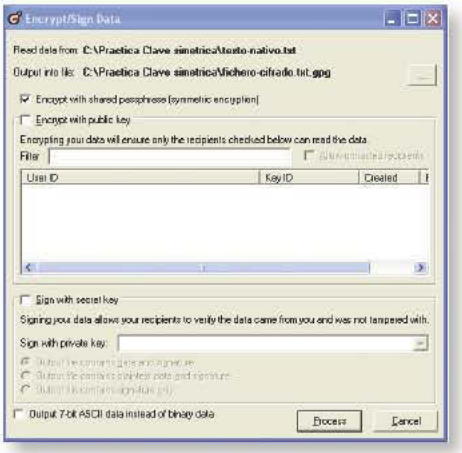
Vamos a mostrar con un ejemplo cómo funciona un criptosistema de clave simétrica.

Para ello utilizaremos un programa llamado Cryptophane, que funciona bajo Windows. Se trata de una aplicación de software libre que además permite utilizar el sistema de cifrado simétrico y el asimétrico. Para el cifrado de clave pública utiliza GnuPG, una versión libre de PGP.

Cryptophane permite encriptar archivos y firmarlos para verificar su autenticidad. También permite descifrar archivos cifrados y verificar firmas generadas con cualquier aplicación de OpenPGP.

En primer lugar instalamos el programa Cryptophane descargándolo de la siguiente dirección web: http://code.google.com/p/cryptophane/

A continuación creamos en el equipo un documento de texto. No es necesario que su extensión sea .doc u .odt, basta con un archivo en texto plano de tipo .txt. Lo llamamos *Fichero-nativo.txt* y escribimos un contenido dirigido a otra persona.



Abrimos el programa y seleccionamos *File / Encrypt* (o <Ctrl> + <E>) para seleccionar un documento a encriptar.

Seleccionamos el archivo e indicamos que se guarde con el nombre *Fichero-cifrado.txt* (botón *Output into file...*). En la ventana que aparece después, seleccionamos *Encrypt with shared passphrase (symmetric encryption)* (y desmarcamos *Encrypt with public key)*. Hacemos clic en el botón *Process* y luego escribimos dos veces la clave que vamos a usar para el cifrado simétrico.

Enviamos el documento a otra persona y le indicamos la clave a través de un canal seguro, porque si no la encriptación no serviría de nada (por ejemplo, si le enviamos el documento cifrado, lo que no podemos es enviarle la clave en el mismo mensaje).

Cuando el destinatario del mensaje quiera abrir el fichero cifrado con el Bloc de notas, le pedirá la clave utilizada para encriptarlo, que le habrá sido proporcionada por el emisor. Sin esa clave, no podrá abrir el archivo. Evidentemente este archivo es ininteligible para cualquier persona que lo abra sin tener la clave.

Actividades propuestas

4·· Si tenemos seis usuarios que quieren comunicarse por medio de cifrado simétrico, ¿cuántas claves serán necesarias? ¿Cuántas se necesitarían si se aumentara en un usuario más?

5·· Investiga qué otros algoritmos de clave simétrica existen además de los expuestos en este apartado y explica brevemente en qué consisten.

3 >> Cifrado de clave asimétrica

Los sistemas de clave simétrica vistos en el epígrafe anterior se basaban en que el proceso de descodificación de un mensaje era esencialmente igual al utilizado para la codificación, solo que a la inversa. La gran novedad de los sistemas de cifrado de clave asimétrica es, precisamente, que la clave y el sistema utilizado para cifrar el mensaje son diferentes a los usados para el descifrado.

Estos sistemas utilizan **un par de claves: una privada** (que solo conoce su propietario) y **otra pública**, que es de conocimiento general y, de hecho, se distribuye a todo el mundo. Pese a la existencia de estas dos claves, estos sistemas se suelen conocer popularmente como sistemas de clave pública.

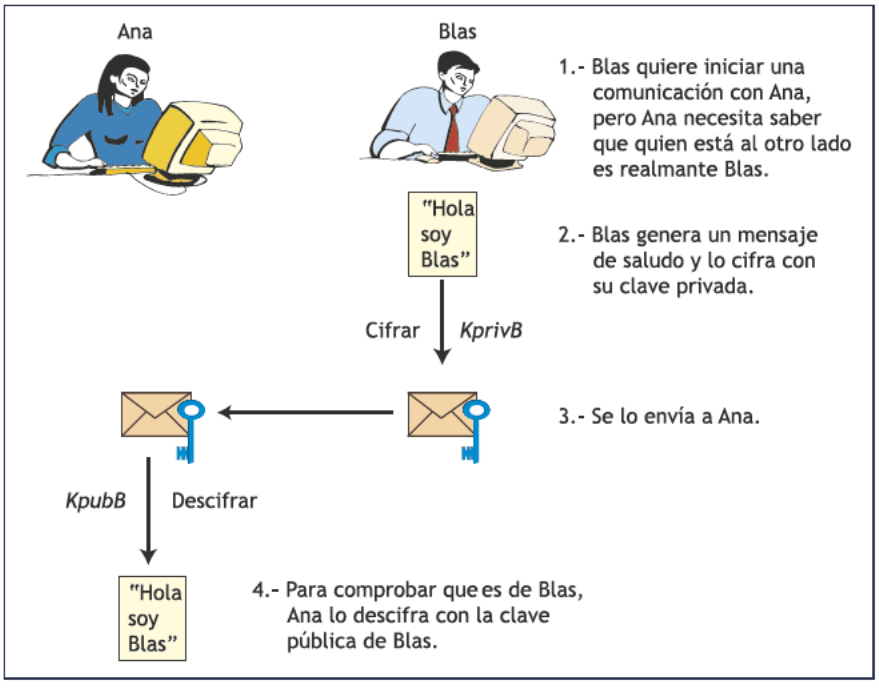
La criptografía asimétrica tiene dos usos principales:

– Autenticación.
– Confidencialidad.

3.1 > Autenticación con claves asimétricas

Para garantizar que el remitente de un mensaje es quien dice ser, este cifra el mensaje con su clave privada (no se cifra el mensaje completo, ya que esto supondría un mayor consumo de CPU, sino que se genera un mensaje *digest* que es el que se cifra). Todo el mundo que posea la clave pública de ese remitente podrá descifrar el *digest* y comprobar que procede de esa persona porque solo él, que es quien posee la clave privada, ha podido generar el mensaje.

Un ejemplo de este uso es el intercambio de claves SSH entre servidores.



4.6. Esquema de un sistema de autenticación mediante clave asimétrica.

Ejemplos

Intercambio de claves SSH entre servidores Debian

Queremos configurar un servidor de gestión de logs para que se conecte a otros servidores, recoja los ficheros de logs y los guarde en un repositorio centralizado para su análisis posterior. Se quiere que este proceso se realice de forma automatizada mediante *scripts* programados para ejecutarse a una determinada hora. Para ello, queremos habilitar un método seguro para que el servidor de logs pueda conectarse al resto sin necesidad de que haya un operador introduciendo las contraseñas en cada caso.

Los sistemas Linux suelen llevar instalado con la distribución el paquete openSSH, que permite realizar determinadas operaciones relacionadas con las claves privada y pública del servidor. Vamos a usar las utilidades de openSSH para realizar un intercambio de claves entre el servidor que va a recoger los logs (debian1) y el servidor donde se generan estos logs (debian2).

El primer paso es generar la pareja de claves SSH que identifican a debian1. Cada pareja de claves va asociada a un usuario del sistema operativo, por lo que deberemos generar las claves con el mismo usuario que van a utilizar los *scripts* para conectarse. Además, mediante openSSH podemos generar claves con el algoritmo RSA o DSA. Elegiremos DSA, mediante el parámetro –t, por ser más seguro:

```
debian1:~# ssh-keygen -t dsa
```

En el proceso nos pregunta dónde queremos guardar el par de claves (por defecto se guardarán en */root/.ssh/)*. También nos preguntará por una palabra de seguridad, pero en este caso deberemos dejarlo en blanco porque de lo contrario habrá que escribirla cada vez que queramos acceder al servidor destino. Comprobamos que nos ha creado correctamente el par de claves:

```
debian1:~# ls /root/.ssh/
id_dsa     id_dsa.pub
```

El primer fichero contiene la clave privada, el segundo la pública. Tendremos que incorporar el contenido de este segundo al fichero de claves autorizadas de debian2. Así, desde debian1, enviamos el fichero a debian2:

```
debian1:~# scp /root/.ssh/id_dsa.pub debian2:/tmp
```

En debian2, creamos su pareja de claves y añadimos el contenido del fichero de clave pública de debian1 al fichero de claves autorizadas:

```
debian2:~# ssh-keygen -t dsa
debian2:~# cat /tmp/id_dsa.pub >> /root/.ssh/authorized_keys
```

Si ahora nos logueamos en debian2 desde debian1, comprobaremos que ya no se nos pide contraseña:

```
debian1:~# ssh debian2
Linux debian2 2.6.26-2-686 #1 SMP Sun Mar 4 22:19:19 UTC 2012 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jul 27 23:27:17 2012 from debian1.pruebas.es
debian2:~#
```

En cualquiera de los ficheros *id_dsa.pub* podrás ver el aspecto de una clave pública generada con DSA.

3.2 > Confidencialidad con claves asimétricas

Los sistemas de cifrado con claves asimétricas sirven para garantizar la confidencialidad del mensaje, al igual que la criptografía simétrica.

Cuando alguien quiera cifrar un mensaje dirigido a mí, utilizará mi clave pública (que es conocida) para cifrarlo, pero únicamente yo lo podré leer, ya que soy el único que posee la clave privada. Funcionaría de forma similar a un candado, cualquiera puede cerrarlo, pero solo quien tenga la llave de ese candado puede abrirlo.

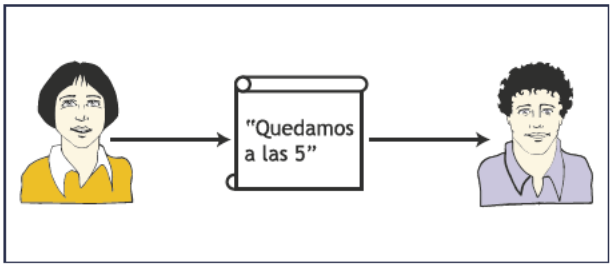
La clave pública tiene un valor, que es un número *(X)* que es el resultado de multiplicar dos números primos (*y*, *z*), que serían la clave privada. El valor de *X* es conocido, por lo que cualquiera que lo conozca puede cifrar un documento, pero solo quien conozca los valores de *z* e *y* podrá descifrarlo. Para que este método tenga éxito debe ser imposible hallar la clave privada a partir de la pública. ¿Cómo se consigue esto? Haciendo que los números primos utilizados sean muy grandes, lo que hará que el número *X* sea enorme, con lo que su factorización para poder averiguar la clave privada o el descubrimiento de esta, casualmente, sea una labor casi imposible. Por ejemplo, un valor de *X* aceptable para transacciones seguras es superior a 10^{300}.

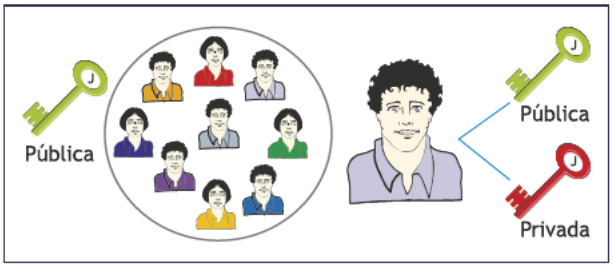
Ejemplos

Utilización de clave asimétrica

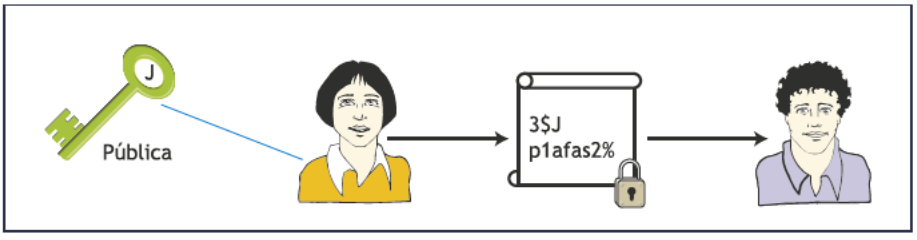
Vamos a ver el funcionamiento de la clave asimétrica a través de un ejemplo que muestra un proceso de comunicación segura entre dos usuarios.

Ana quiere enviar un mensaje secreto a Juan cifrándolo mediante una clave asimétrica. Juan dispone de una pareja de claves pública y privada y ha distribuido su clave pública de forma que cualquiera que quiera enviarle un mensaje pueda usarlo para cifrarlo.



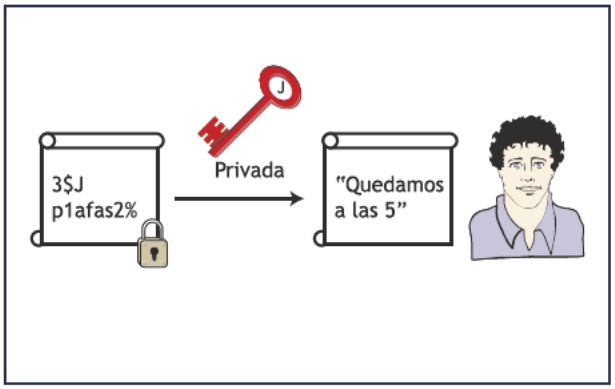


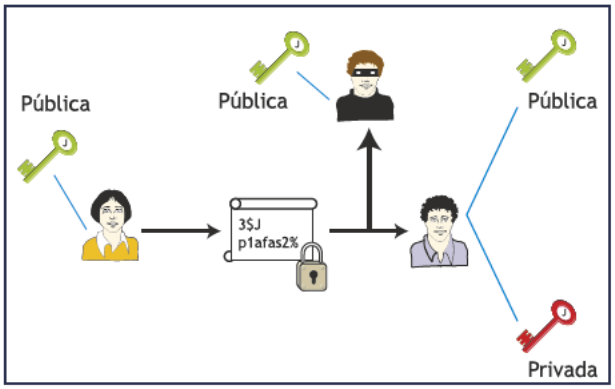
Como la clave pública de Juan es conocida, Ana la utilizará para cifrar el mensaje.



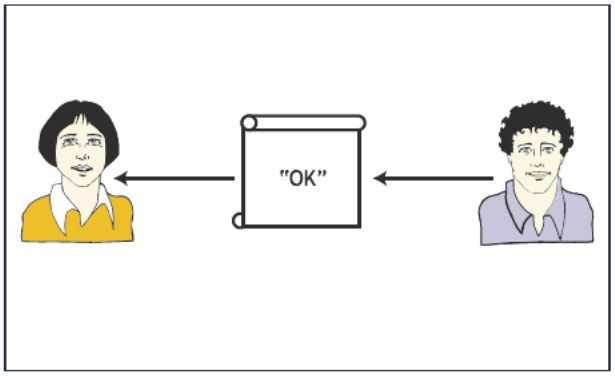
Una vez cifrado el mensaje con la clave pública, solo Juan podrá leerlo, porque él es la única persona que tiene su clave privada. Al estar cifrado con la clave pública, la privacidad del mensaje para Juan está garantizada, puesto que solo quien conozca la clave privada podrá descifrar el mensaje.

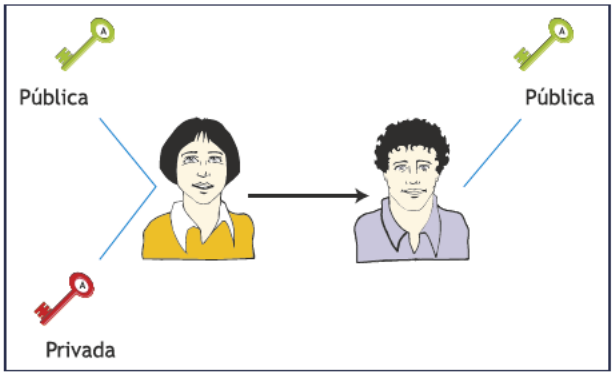
Por esto no hay ningún inconveniente para que la clave pública de Juan sea públicamente conocida, pues por sí misma no será suficiente para descifrar el mensaje. Lo realmente importante es que la clave privada esté a buen recaudo, pues si un intruso la averiguara, sí que podría acceder al contenido del mensaje.



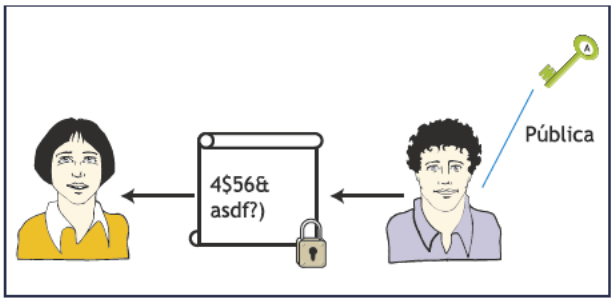


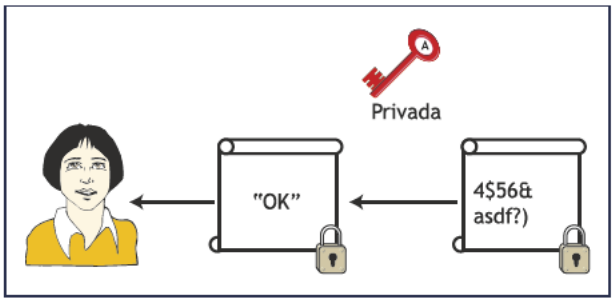
Ahora, supongamos que Juan quiere contestarle a Ana en secreto. Para cifrar el mensaje, Juan debe tener la clave pública de Ana. Por ello, Ana debe generar su propio par de claves pública/privada y enviar a Juan su clave pública.





Juan utiliza la clave pública de Ana para proteger el mensaje y solo Ana podrá leer el mensaje porque es la única persona que conoce la clave privada.





Ventajas e inconvenientes de los sistemas de cifrado asimétricos	
Ventajas	- La clave pública se distribuye libremente, por lo que ya no existe el problema del intercambio de la clave que había en los métodos simétricos. - Solo es necesario un par de claves por interlocutor, con independencia del número de estos, por lo que el espacio de claves es más manejable cuando los interlocutores son muchos.
Inconvenientes	- Requieren mayor tiempo de proceso que el cifrado simétrico. - Dan lugar a mensajes cifrados de mayor tamaño que los originales. - Para garantizar la seguridad, requieren claves de mayor tamaño que en el caso de los métodos simétricos. - Puesto que las claves públicas se distribuyen libremente, hace falta un esquema de confianza que garantice la autenticidad de las claves públicas (que la clave pública sea de quien dice que es, que no ha sido comprometida, etc.).

Infraestructuras de clave pública

Para que se genere un certificado personal (por ejemplo, el DNIe), es necesario ir en persona a una oficina para que el funcionario verifique nuestra identidad. Así se nos proporcionarán un par de claves asociadas a un certificado que dice que hemos sido correctamente identificados.

Un asunto importante es la autenticidad de las claves públicas, ¿quién nos garantiza que la clave pública del interlocutor es suya realmente?

Existen varios mecanismos para comprobar la autenticidad de las claves públicas, ya que es aquí donde radica la debilidad principal de estos sistemas:

- **Infraestructuras de clave pública (PKI)**: utilizadas, por ejemplo, para verificar las claves públicas de los certificados del DNI electrónico y el resto de certificados generados por la Fábrica Nacional de Moneda y Timbre.
- **Listas de revocación de certificados**.

3.3 > Algoritmos de cifrado

Los algoritmos de cifrado asimétrico mas conocidos son los siguientes:

RSA *(Rivest-Shamir-Adelman)*

Fue creado en 1977 y es uno de los algoritmos más utilizados. Permite cifrar y firmar digitalmente, aunque es mucho más lento que DES y que otros sistemas de cifrado de clave simétrica. Es el sistema que desarrollamos en el apartado 3.2 y está basado en la factorización de números primos grandes.

DSA *(Digital Signature Algorithm)*

Algoritmo de firma digital, estándar del Gobierno Federal de Estados Unidos. Para entornos críticos, se ha demostrado que DSA es más seguro que RSA. Permite firmar digitalmente, sin embargo no permite cifrar la información. Una desventaja es que requiere más tiempo de cómputo que el algoritmo RSA.

ElGamal

Fue escrito por Taher ElGamal en 1984. Algoritmo de uso libre utilizado en software GNU Privacy Guard, en versiones recientes de PGP. Puede ser utilizado para cifrar y firmar digitalmente, con un tiempo de cómputo similar a RSA. Su nivel de seguridad está basado en la dificultad de calcular un logaritmo discreto.

Casos prácticos 1

Criptografía de clave asimétrica

•• Utilizando la aplicación de software libre Cryptophane para Windows, realiza las siguientes tareas:

a) Genera un par de claves una para ti y otra para el destinatario de los mensajes, que será un compañero de clase.
b) Exporta e importa las claves públicas.
c) Crea un documento que deberás enviar a un compañero cifrado con la clave pública del destinatario.

Solución ••

a) Generación de pares de claves

Utiliza el menú *Keys /Generate Secret Key* para generar un par de claves. Te pedirá tu nombre, *email*, descripción y fecha de caducidad de la clave. Introduce la siguiente información:

Caducidad de claves: un año.

Longitudes de clave: 1024 bits, que ya proporciona un nivel de seguridad elevado. A mayor longitud de clave, más seguridad, pero son necesarios más recursos para cifrar y descifrar.

En la ventana de la aplicación puedes ver la clave pública asociada a la clave privada que has creado. En la ventana principal, lo que se visualiza son todas las claves públicas de tus contactos. Para observar las propiedades de la clave, haz doble clic sobre una de ellas o bien haz clic con el botón secundario y selecciona *Key Properties*. A este repositorio de claves se le conoce como anillo de claves.

b) Exportación/Importación de claves públicas

Para exportar la clave pública debes ir al menú *File / Export Public Keys*. Exporta tu clave pública para pasársela a tu compañero. Tu compañero debe realizar la misma tarea y pasarte su clave pública a ti.

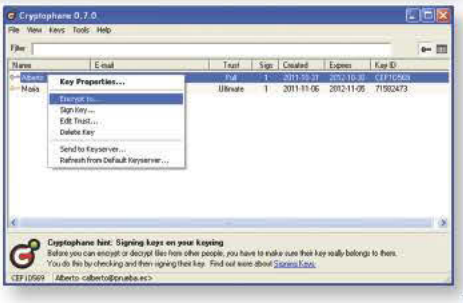
Para importar la clave pública de tu compañero debes ir al menú *File / Import Keys*.

Edita el nivel de confianza (*trust*) de la clave que has importado del compañero y, como te fías de ella, márcala como *I trust this user fully*. Con esta acción has dado el mayor nivel de confianza a la clave de tu compañero.

c) Cifrado de documentos

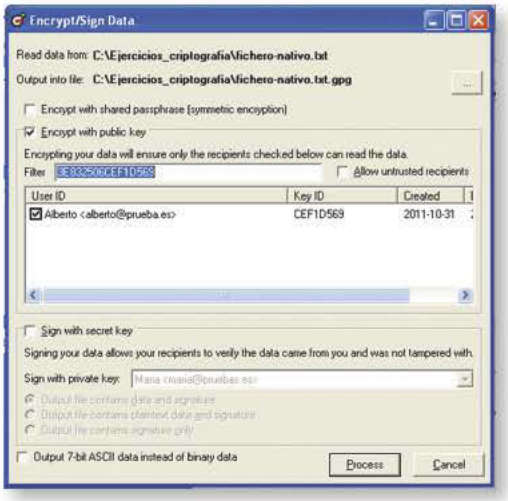
Crea en tu equipo un documento de texto (no hace falta que sea en .doc u .odt, con texto plano .txt es suficiente) llamado *Asimétrico.txt* y escribe un contenido para el compañero o compañera que va a recibir el documento cifrado.

Selecciona su clave, haz clic con el botón secundario y selecciona *Encrypt to...* En este momento estás cifrando el documento que enviarás a tu compañero usando su clave pública.



1

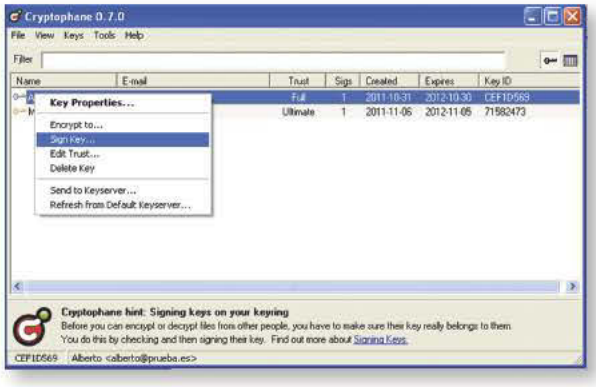
Selecciona el fichero que has escrito antes y marca en la casilla de verificación *Encrypt with public key* (cifrar con clave pública). Haz clic en *Process*. Puedes observar que la aplicación no te deja usar claves públicas de tu anillo hasta que firmes con tu clave privada.

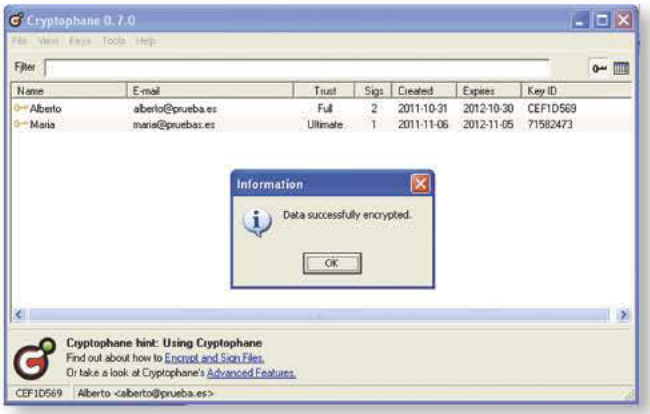


Verifica con el compañero a quien vas a enviar el mensaje su clave pública y fírmala utilizando tu clave privada. Para ello, utiliza el menú contextual que se abre al hacer clic con el botón secundario del ratón. Una vez firmada, el programa confía plenamente en esa clave pública.

Vuelve a intentar de nuevo cifrar el documento destinado a tu compañero y comprueba ahora que el programa ya te permite hacerlo. Envíale el documento cifrado al compañero y pídele que lo descifre con su clave privada. Haz lo mismo tú con el documento cifrado que él te pase.

Comprueba que se te pide la clave con la que has protegido tu clave privada para descifrar el documento.





Actividades propuestas

6·· ¿En qué consiste el cifrado de clave asimétrica? Cita algunos ejemplos, además de los indicados en el texto, de algoritmos de cifrado de este tipo.

4 >> Algoritmo de cifrado *hash*

Tan importante como la autenticación y la confidencialidad es la integridad de los mensajes. De nada sirve que el canal sea seguro si no es posible garantizar que el mensaje no ha sido alterado y se corresponde exactamente con el original.

Una función *hash* es un algoritmo que mapea un conjunto grande de datos de tamaño variable, llamados claves, en pequeños conjuntos de datos de longitud fija.

Los algoritmos *hash* resultan de vital importancia en la firma digital de documentos y mecanismos como el sobre digital. Garantizan la integridad dado que, con cambiar un solo bit del mensaje original, el resultado obtenido al aplicar la función *hash* será diferente.

Las propiedades más importantes de las funciones *hash* son:

- Independientemente del tamaño del mensaje original, al aplicarle la función *hash,* la huella resultante siempre tendrá el mismo tamaño.
- Con cambiar un único bit del mensaje original, la huella resultante será completamente distinta.
- **Resistencia a la preimagen:** si tenemos el resultado de aplicar una función *hash,* resulta computacionalmente imposible obtener el mensaje original a partir de este.
- **Resistencia a la segunda preimagen:** dado un mensaje *x*, no es posible encontrar otro mensaje *x'* que produzca el mismo valor *hash.*
- **Resistencia a colisiones:** no es posible encontrar dos entradas que den lugar al mismo valor *hash.*

Entre las aplicaciones de las funciones *hash* se destacan:

- Protección de contraseñas.
- Se utilizan como parte de algunos de los pasos de los algoritmos de cifrado simétrico y asimétrico vistos anteriormente.
- Es una parte fundamental del mecanismo de firma digital.
- Se emplea para garantizar la integridad de un flujo de datos, como por ejemplo el software que nos descargamos (es muy usado en sitios de software libre, en los que al lado del enlace al software suele encontrarse un enlace a la huella *hash* del archivo de forma que, antes de instalar, podamos comprobar que el software se ha descargado íntegramente y sin errores).

Como algoritmos *hash* destacados tenemos SHA, SHA-1, MD5 y RIPE-MD.

Sobre digital

El sobre digital es un mecanismo que garantiza las propiedades de confidencialidad de un documento. Utiliza criptografía simétrica y asimétrica.

Combinando los sobres digitales con las firmas digitales obtenemos un sobre digital firmado, garantizándose así las propiedades de integridad, confidencialidad y autenticación.

Encriptación de contraseñas

Una medida eficaz cuando trabajamos con sistemas de autentificación de usuarios mediante usuario y contraseña por Internet, por ejemplo, es encriptar las contraseñas con MD5, de forma que en caso de que alguien accediese a ellas solo pudiera ver su encriptación y no la contraseña.

Actividades propuestas

7·· Busca en Internet una web que permita cifrar *online* un texto usando el algoritmo MD5 (por ejemplo, la siguiente: http://www.cuwhois.com/herramienta-seo-genera-md5.php). Cifra el siguiente texto: "Buenos días, soy un alumno". ¿Cuál es el resultado de cifrar la cadena anterior?

8·· Realiza un breve resumen con las características más significativas de los algoritmos *hash* que se citan en el texto.

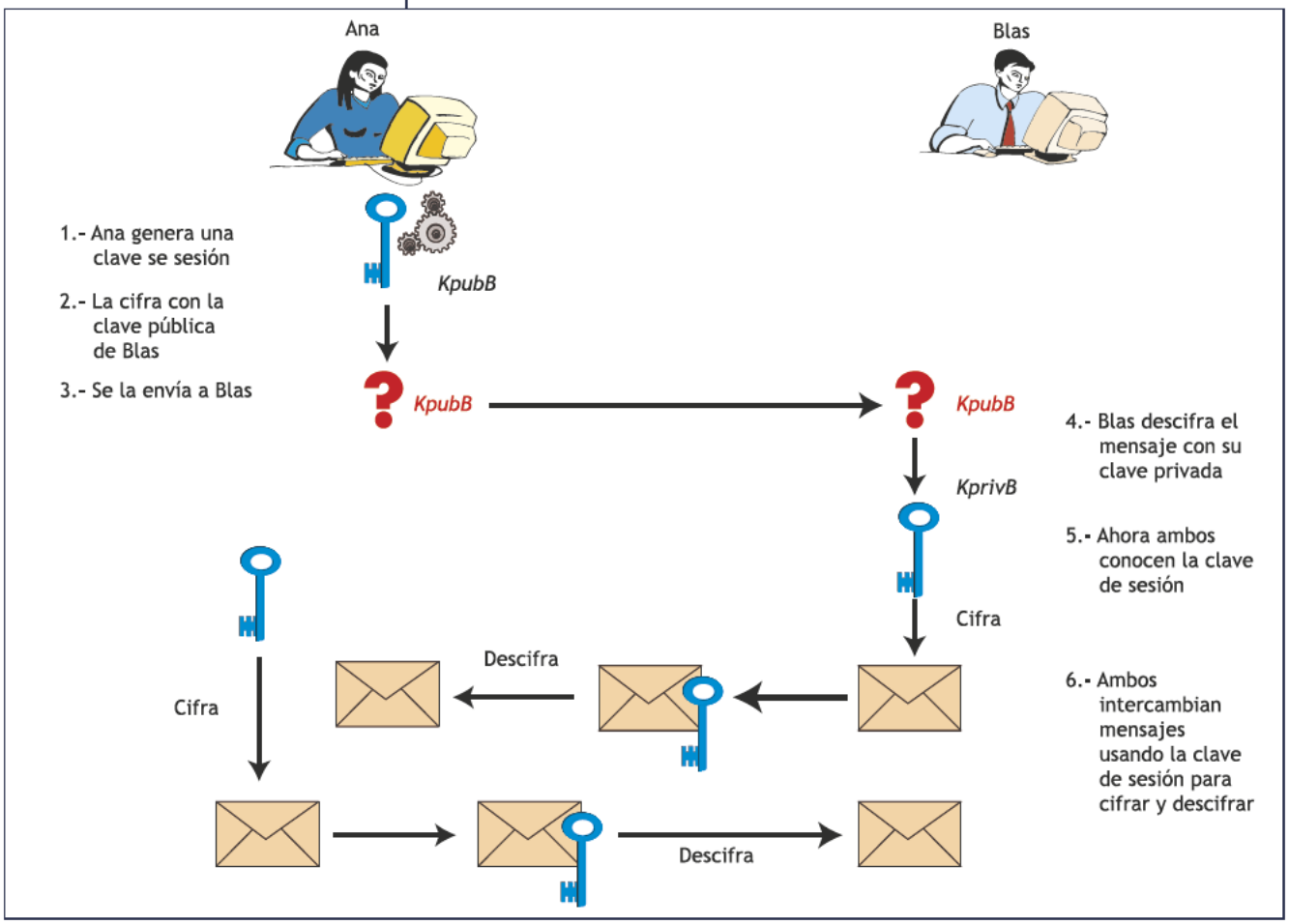
5 >> Sistemas híbridos

Como se ha visto en los apartados anteriores, tanto los sistemas simétricos como los asimétricos presentan ventajas e inconvenientes. Con los sistemas híbridos se pretende aprovechar la mayor eficiencia para cifrar y descifrar de los métodos simétricos solucionando al mismo tiempo el problema de la transmisión segura de la clave que presentan estos sistemas.

Para ello, cuando dos agentes quieren entablar una comunicación segura, en primer lugar establecen una conexión segura haciendo uso del sistema de clave pública.

A través de esta conexión securizada por métodos asimétricos, se intercambian una clave de tipo simétrico con la que realizarán el resto de la comunicación. De este modo, la penalización del rendimiento que supone el uso de claves asimétricas solo se da al inicio de la comunicación, cuando ambos agentes se ponen de acuerdo en la clave simétrica que van a usar.

Los criptosistemas híbridos tratan de aprovechar lo mejor de cada uno de los sistemas de cifrado de clave simétrica y asimétrica. Se trata de obtener un criptosistema rápido y eficiente que permita el intercambio de contraseñas en canales de comunicación inseguros.



4.7. Esquema de la transmisión de mensajes mediante sistemas de criptografía híbrida.

Los sistemas híbridos más importantes son: PGP, GnuPG y OpenPGP.

5.1 > PGP *(Pretty Good Privacy)*

Se trata de una herramienta sencilla, barata y potente desarrollada por Phil Zimmermann. Permite el cifrado de datos, archivos y mensajes mediante la utilización de codificación asimétrica junto con la simétrica. Su objetivo es proteger la información utilizando criptografía de clave pública y facilitar la autenticación de documentos mediante las firmas digitales.

Utiliza claves asimétricas que son almacenadas en el disco duro en ficheros llamados **llaveros**. Existe un llavero para las claves públicas utilizadas y otro para las claves privadas.

Sus aplicaciones más comunes son:

- Cifrado de ficheros, documentos y discos (PGPDisk).
- Firma digital y cifrado de correos electrónicos (PGPmail).
- Comunicaciones seguras (PGPNet).

El proceso de cifrado consiste en que la clave pública del receptor cifra la clave simétrica o clave de sesión con la que se cifra el mensaje a enviar. En el proceso de descifrado el receptor utiliza su clave privada para descifrar la clave de sesión secreta, la cual, a su vez, se utiliza para descifrar los datos comprimidos.

5.2 > OpenPGP

Su diseño está basado en la implementación PGP de Phil Zimmermann. El grupo IETF (*Internet Engineering Task Force*) creó el estándar de Internet OpenPGP basandose en el diseño de PGP. Se trata por tanto de un protocolo de encriptación de correo electrónico libre basado en criptografía de clave asimétrica.

Define formatos estándar para crear mensajes encriptados, firmas, certificados e intercambio de claves privadas. Actualmente, se considera el estándar a utilizar en la encriptación de correos electrónicos.

5.3 > GnuPG *(GNU Privacy Guard)*

Se trata de una herramienta de software libre y de código bajo licencia GPL utilizada para el cifrado y firmas digitales. Añade mejoras de seguridad y nuevas funcionalidades respecto a PGP.

Utiliza algoritmos no patentados, como son ElGamal, CAST5, TripleDES, AES y Blowfish. Se utiliza en algunos sistemas operativos, como FreeBSD, OpenBSD, GNU/Linux, Mac OS X o Windows, así como en algunos clientes de correo electrónico, como Kmail y en gestores de información personal, como Evolution.

Actividades propuestas

9·· ¿En qué se basan los criptosistemas híbridos?

10·· Indica algunas aplicaciones de los sistemas híbridos nombrados en el texto.

Actividades finales

.: CONSOLIDACIÓN :.

1·· ¿Qué es la criptografía y cuál es su finalidad? Explícalo con tus palabras.

2·· ¿De qué se compone un criptosistema?

3·· Indica en qué dos operaciones básicas se basan los sistemas criptográficos clásicos.

4·· ¿En qué consiste la transposición? Crea un mensaje utilizando dicha técnica.

5·· ¿En qué consiste la técnica de permutación o sustitución? Crea un mensaje usando dicha técnica.

6·· ¿Qué tipos de sistemas de cifrado se han visto en la unidad? Explica brevemente, utilizando tus propias palabras, en qué consiste cada uno de ellos.

7·· ¿Qué tipo de cifrado es el que se conoce como de clave secreta?

8·· ¿Qué desventaja crees que ofrece el método de cifrado simétrico?

9·· ¿Qué novedad aportan los sistemas de cifrado de clave asimétrica con respecto a los de clave simétrica?

10·· ¿Cuál de los dos sistemas de cifrado vistos es más rápido? ¿Por qué?

11·· ¿Qué garantizan los algoritmos de cifrado *hash*?

12·· Explica cómo se establece una comunicación entre dos interlocutores utilizando un sistema híbrido.

13·· ¿Qué ventajas ofrecen los sistemas híbridos?

.: APLICACIÓN :.

1·· Crea un mensaje utilizando la técnica de la transposición y, posteriormente, pásaselo a un compañero para que lo intente descrifrar.

2·· Cifra el mensaje "La máquina Enigma se usó en la segunda guerra mundial para cifrar mensajes" utilizando la técnica de la permutación o sustitución:

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

3·· Te han encargado establecer un mecanismo de seguridad para proteger las contraseñas de una aplicación. Dicha aplicación almacena la información de los usuarios en una base de datos, concretamente en una tabla con los campos *Usuario* y *Contraseña*. Ahora mismo la contraseña se almacena tal y como la introduce el usuario. ¿Cómo podrías proteger los valores introducidos en el campo *Contraseña* para que un acceso a dicha tabla no descubra las contraseñas?

4·· Dados los siguientes mensajes cifrados con el sistema MD5:

a) 9be2d8141fb9aba6aaafb6ddf22c5ed4
b) 4d186321c1a7f0f354b297e8914ab240

¿Hay forma de saber la longitud del mensaje original? ¿Y de saber cuál fue el mensaje original?

5·· ¿Se pueden cifrar los mensajes de cuentas de correo como Gmail o Hotmail?

6·· ¿Es seguro indicar a los navegadores que recuerden las contraseñas cuando las utilicemos en páginas web, formularios, etc.? Elige un navegador y averigua dónde y cómo se almacenan las contraseñas.

Caso final 2

Protección en el correo electrónico

•• Gema y David son dos amigos que frecuentemente se envían mensajes relativos a sus respectivos negocios y les gustaría utilizar para sus comunicaciones un gestor de correo que permitiera cifrar sus comunicaciones mediante técnicas de cifrado asimétrico basadas en GnuPG. Hablan con una amiga común, Rosa, y les recomienda que utilicen el gestor de correo electrónico Thunderbird, por ser libre y muy sencillo de configurar y que añadan al mismo la extensión de seguridad Enigmail que permite cifrar correos electrónicos.

a) ¿Cómo instalarían Gema y David el software necesario?
b) ¿Cómo se llevaría a cabo el proceso de comunicación segura entre David y Gema?

Solución ••

a) El primer paso que deben llevar a cabo Gema y David será instalar el gestor de correo y el soporte para GnuPG en sus dos equipos.

El soporte GnuPG se puede encontrar en la web siguiente: ftp://ftp.gnupg.org/gcrypt/ Para instalar en Windows habrá que ir a la carpeta *binary*. La instalación es rápida y sencilla.

Una vez instalado el soporte GnuPG, deben descargar e instalar el gestor de correo Thunderbird, lo cual pueden hacer desde de la página web http://www.mozilla.org/es-ES/thunderbird/

Una vez instalado, deben configurarlo para que trabaje con una cuenta de correo real y comprobar que funciona la cuenta de correo enviando un mensaje a la misma y desde ella.

A continuación, deben instalar el gestor de correo Enigmail, que es una extensión de seguridad para Mozilla Thunderbird que permite escribir y recibir mensajes firmados y/o cifrados mediante el estandar GnuPG. Descargan la extensión Enigmail para Thunderbird desde la dirección: http://enigmail.mozdev.org/download/index.php.html.

Seleccionan el sistema operativo y la versión de Thunderbird instalada y, a continuación, desde la interfaz de Thunderbird seleccionan *Herramientas / Complementos (Tools - Addons)*.

Haciendo clic sobre el símbolo de configuración (esquina superior izquierda) aparece un menú en el que elegirán la opción *Instalar complemento desde archivo* y buscarán el archivo recientemente descargado, que contiene la extensión Enigmail. Aparecerá una ventana en la que se advierte que solo se instalen complementos de autores de confianza y se hace clic en el botón *Instalar ahora*.

b) Una vez instalado el software necesario para llevar a cabo una comunicación segura, ya podrán enviar el mensaje cifrado. Para ello, si imaginamos que Gema va a enviar un mensaje a David, el proceso sería el siguiente:

- David debe crear un par de claves (una pública y otra privada).
- David debe pasarle su clave pública a Gema.
- Gema usará la clave pública de David para cifrar el mensaje.
- Cuando reciba el mensaje, David podrá descifrarlo con su clave privada.

Para crear el par de claves, David selecciona en el menú de Thunderbird *OpenPGP-Administración de claves (OpenPGP-Key management)*.

»

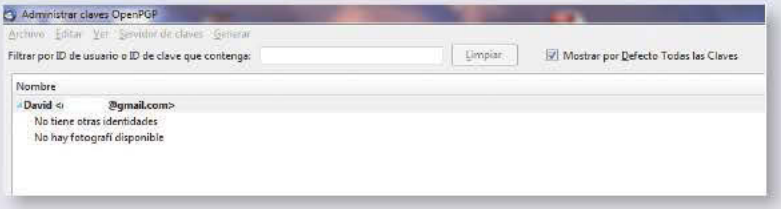
2

Se muestra la ventana *Administrar claves OpenPGP*. En dicha ventana, selecciona la opción de menú *Generar / Nuevo par de claves*.

En la pestaña *Expiración de la clave*, selecciona *1 año*. A continuación, en la pestaña *Avanzadas* elige como tamaño de clave *1024 bits* y como tipo de clave *DSA*. Se pide además que se introduzca una "frase" (en realidad se trata de la clave). Esta frase o clave servirá para proteger la clave privada generada y deberá tener una longitud adecuada (8 caracteres o más), que incluya números, letras y caracteres especiales. Además se recomienda que no sea una palabra que exista en el diccionario ni una cadena de caracteres previsible (por ejemplo, "1234" no será una buena elección como frase). Una vez elegida la frase, selecciona el botón *Generar clave*.

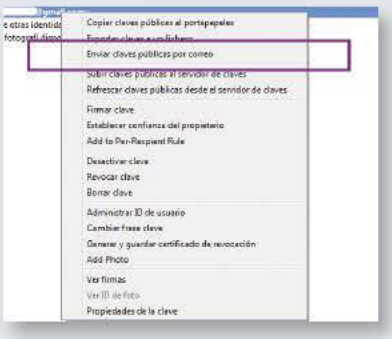
Una vez creados el par de claves, se indicará que es recomendable crear un certificado de revocación para esta clave. Por tanto, selecciona *Generar certificado*.

Si todo ha funcionado bien, en la ventana principal aparecen las claves creadas (*Administrar claves OpenPGP*). Si no es así, marca la opción *Mostrar por defecto todas las claves*.



Una vez que ha generado las claves, David deberá exportar su clave pública y pasársela a Gema, quien le enviará un correo cifrado. David podrá enviarle la clave pública por correo electrónico, colgarla en un servidor de claves, etc.

Si desea enviarla por correo electrónico bastará con que la adjunte a un mensaje mediante el botón *OpenPGP*. Para ello, irá a la ventana de *Administración de las claves OpenPGP* y, sobre su ID de usuario, hará clic con el botón secundario del ratón y seleccionará *Enviar claves públicas por correo*.

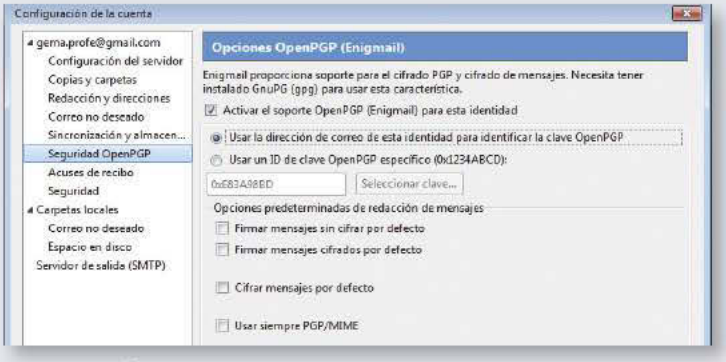


2

A continuación, escribirá el cuerpo de mensaje de correo y seleccionará *Enviar*. Tras hacer clic en este botón, la aplicación le pedirá que inserte la frase o clave. Una vez insertada y tras hacer clic en *Aceptar*, el mensaje de correo electrónico se enviará con la clave pública.

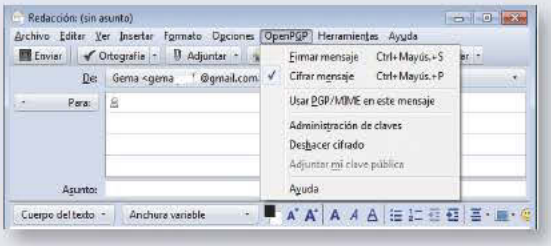
Para que Gema pueda enviarle un mensaje cifrado a David, tendrá que escribirlo y cifrarlo utilizando la clave pública de David, pero antes deberá importar la clave pública de este.

Antes de que Gema cree el correo para enviar a David, es conveniente que seleccione *Herramientas / Configuración de la cuenta* y, en el panel lateral, marque la opción *Seguridad OpenPGP*. En el panel de la derecha, marcará la casilla *Activar el soporte OpenPGP (Enigmail)* para esta identidad, si es que ya no lo estaba antes, y seleccionará la opción *Usar la dirección de correo de esta identidad para identificar la clave OpenPGP*.



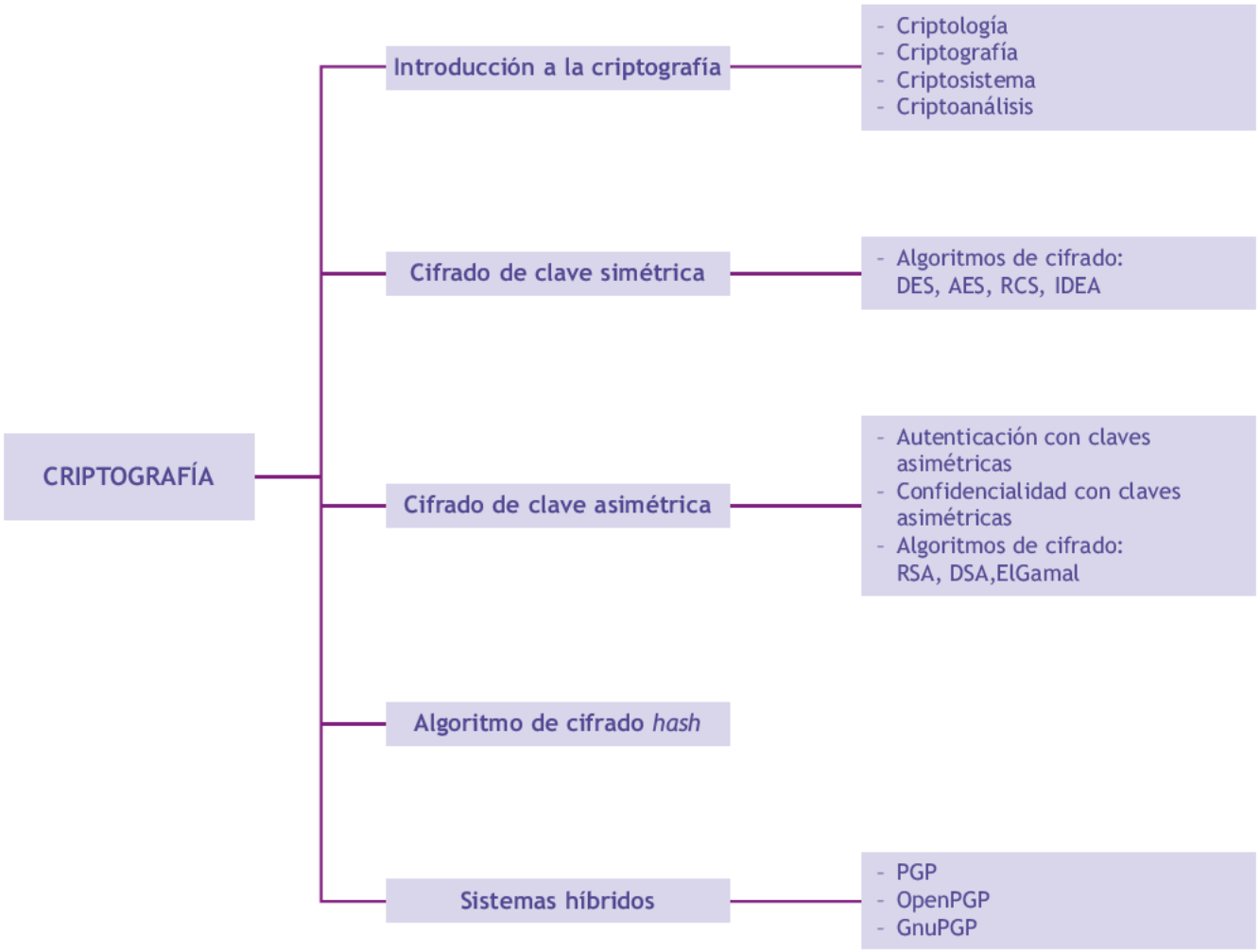
El proceso de importación de la clave de David lo realizará seleccionando *Administrar claves* y seleccionando *Archivo / Importar la clave desde un fichero* que le ha enviado David.

Ahora ya podrá crear y cifrar un correo electrónico mediante GnuPG. Para ello deberá ir a *Redactar* para seleccionar el destinatario y escribir el nuevo correo. Ahora aparece una nueva opción: *GnuPG*. Deberá seleccionar esta opción marcando que desea cifrar el mensaje y ya solo le faltará agregar el *Destinatario* y un *Asunto* y podrá enviarlo.



Cuando David reciba el mensaje cifrado y lo intente abrir, le aparecerá una ventana solicitando la inserción de la clave privada (la frase o clave de paso OpenPGP). Si David no introduce la clave privada, el *mail* se verá sin descifrar, como muestra la imagen de la izquierda; en cambio, al insertar la clave, el mensaje se puede leer ya descifrado, como se ve a la derecha.

Ideas clave


CRIPTOGRAFÍA
Introducción a la criptografía
- Criptología
- Criptografía
- Criptosistema
- Criptoanálisis
Cifrado de clave simétrica
- Algoritmos de cifrado: DES, AES, RCS, IDEA
Cifrado de clave asimétrica
- Autenticación con claves asimétricas
- Confidencialidad con claves asimétricas
- Algoritmos de cifrado: RSA, DSA,ElGamal
Algoritmo de cifrado hash
Sistemas híbridos
- PGP
- OpenPGP
- GnuPGP

El cifrado AES, ¿está roto o no?

Un equipo de investigadores ha encontrado la primera vulnerabilidad en el estándar de cifrado AES reduciendo la longitud efectiva de la clave en 2 bits. Esto implica que las longitudes habituales de 128, 192 y 256 bits se han visto reducidas a 126, 190 y 254 bits. ¿Significa que está roto?

AES *(Advanced Encryption Standard)* es en realidad el algoritmo Rijndael, que pasó a ser un estándar de cifrado aprobado por el gobierno de los Estados Unidos en 2003 para cifrar información clasificada. Su versión de 128 bits está permitido para información secreta mientras que para información *top secret* requiere claves de 192 o 256 bits. De hecho, fue el primer algoritmo público usado para cifrar información *top secret* gubernamental.

Andrey Bodgdanov de la Universidad Católica de Leuven, Christian Rechberger del ENS de París y Dmitry Khovratovich del departamento de investigación de Microsoft ya apuntan que el ataque no tiene una gran relevancia práctica. Aunque el descubrimiento es considerado un avance importante en la investigación de la seguridad del algoritmo AES, puesto que la experiencia dice que en ciertos algoritmos se avanza despacio hasta romperlos. Esta vulnerabilidad ha sido confirmada por los desarrolladores de AES, Joan Daemen y Vincent Rijmen.

Los investigadores emplearon un ataque *meet-in-the-middle,* una aproximación que ha sido principalmente empleada con algoritmos de *hashing*, combinándolo con un ataque *biclique*. Este método ha permitido a los investigadores calcular la clave de un par texto plano/texto cifrado más rápidamente que empleando un ataque de fuerza bruta en el espacio total de la llave. O sea, se ha reducido el número de claves que deben ser probadas. La fuerza bruta total con clave de 128 bits serían 2^{128} posibilidades. Con este ataque serían necesarias solo 2^{126}.

En el sentido estrictamente académico, en algoritmo está roto puesto que se ha reducido (aunque sea en 2 bits) el espacio de claves necesario para calcular la clave por fuerza bruta. Sin embargo, roto no significa que no pueda ser usado con seguridad todavía. Por ejemplo, un ataque contra una llave de 128 bits requiere diez millones de años empleando un parque de un billón de equipos probando cada

uno de ellos un billón de claves. Al reducir en dos bits dicha clave, el tiempo se reduciría a 3 millones de años.

Hasta 2005, el único ataque que se conocía contra AES contemplaba una reducción del número de rondas de cifrado, o sea, una modificación artificial en su implementación que no suele encontrarse habitualmente. Los detalles del ataque fueron presentados en la conferencia CRYPTO 2011 y pueden ser descargados desde la web de investigación de Microsoft.

Fuente: Borja Luaces y Sergio de los Santos. 24/04/2011. http://unaaldia.hispasec.com.

Actividades

1·· El algoritmo de cifrado AES, ¿se utiliza en criptosistemas de clave simétrica o asimétrica?

2·· ¿Qué longitudes de clave se utilizan para cifrar información secreta? ¿Y para cifrar información de tipo *top secret*?

3·· ¿Por qué se considera que el algoritmo está roto?

4·· A partir de la información del texto, ¿consideras que hay que dejar de utilizar este algoritmo de cifrado?