Seguridad de la Información

Una introducción con enfoque práctico

Ing. Mariano Aliaga

Universidad Católica de Córdoba - Facultad de Ingeniería

2021

Panorama General

Criptografía

Criptografía: ciencia que utiliza técnicas para ocultar (cifrar) información de modo que sólo pueda leerse (descifrarse) mediante la utilización de una llave o clave.

- **Confidencialidad:** al convertir un mensaje en otro cuyo contenido de información sólo puede ser accedido por las personas o sistemas autorizados.
- **Integridad:** permite detectar si el mensaje ha sido modificado en su totalidad o en parte.
- **Autenticidad y no repudio:** permite establecer en forma fehaciente la identidad del emisor.

Criptosistema

Criptosistema: lo definimos como una quíntupla (M, C, K, E, D) donde:

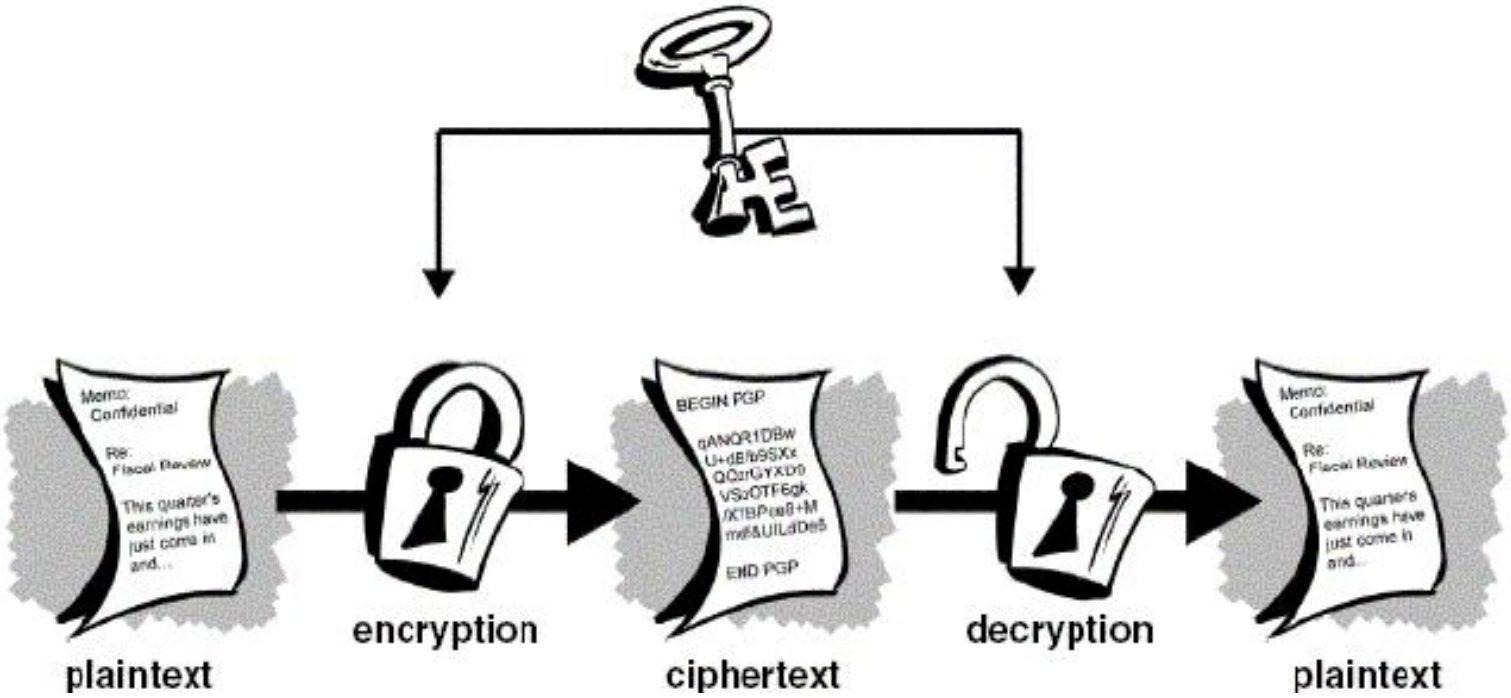
- M: representa el conjunto de todos los mensajes sin cifrar (lo que se denomina texto plano, o plaintext) que pueden ser enviados.
- C: representa el conjunto de todos los posibles mensajes cifrados.
- K: representa el conjunto de claves que se pueden emplear en el criptosistema.
- E: es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de M para obtener un elemento de C. Existe una transformación diferente E_k para cada valor posible de la clave k.
- D: es el conjunto de transformaciones de descifrado, análogo a E.

$$c_i = E_k(m_i)$$

$$m_i = D_k(c_i)$$

$$m_i = D_k(E_k(m_i))$$

Criptosistema

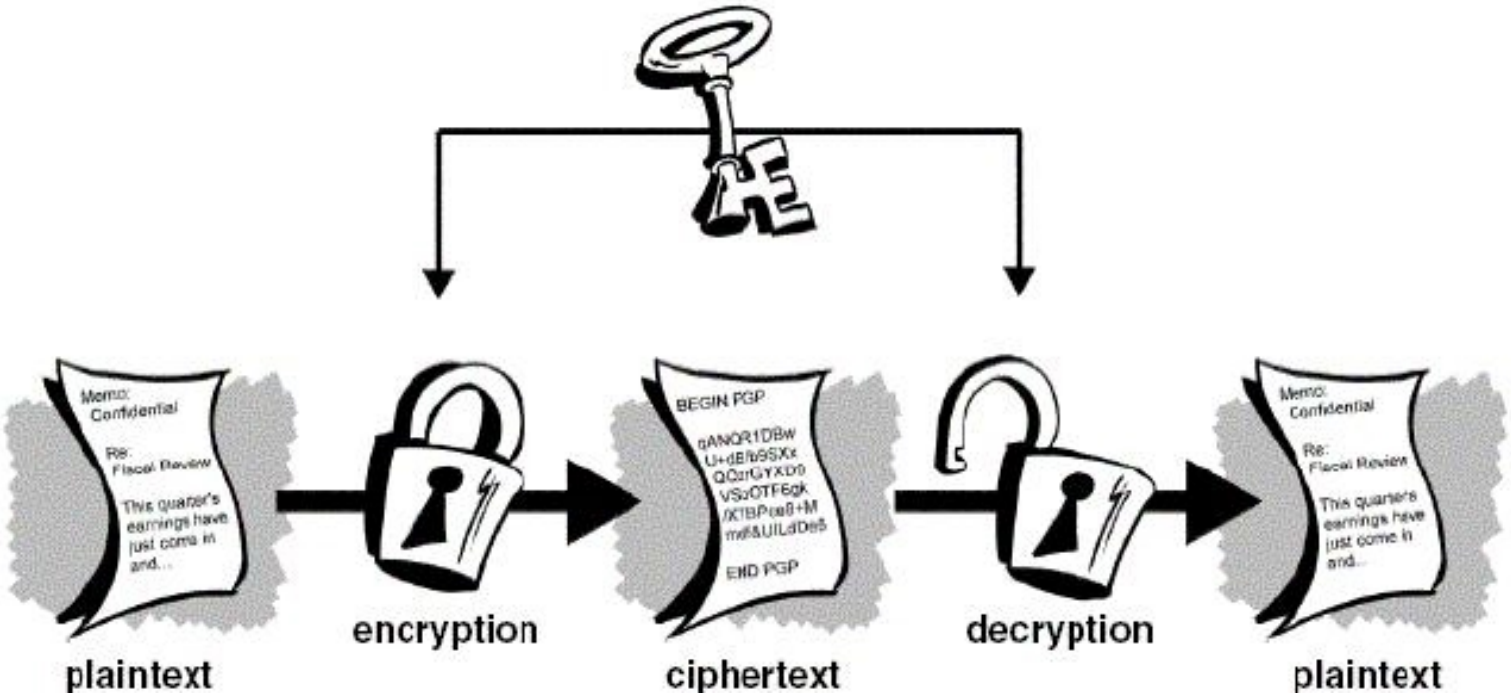


La "Clave" es la longitud de la "Llave"

$$2^8 = 256$$

$$2^{40} = 1.099.511.627.776$$

Criptosistema



La "Clave" es la longitud de la "Llave"

$$2^8 = 256$$

$$2^{40} = 1.099.511.627.776$$

Criterios de clasificación

- De acuerdo a las *técnicas* empleadas en los algoritmos.
  - Criptografía clásica.
  - Criptografía moderna.
- De acuerdo al *tipo de llaves* utilizadas en los algoritmos.
  - Criptografía simétrica.
  - Criptografía asimétrica.
- De acuerdo al *procesamiento* del mensaje.
  - Cifradores de bloque.
  - Cifradores de flujo.

Criptografía clásica y moderna

Criptografía Clásica: utiliza principalmente técnicas basadas en las operaciones de sustitución y transposición de caracteres. Su seguridad está basada solamente en el secreto de la transformación o del algoritmo empleado.

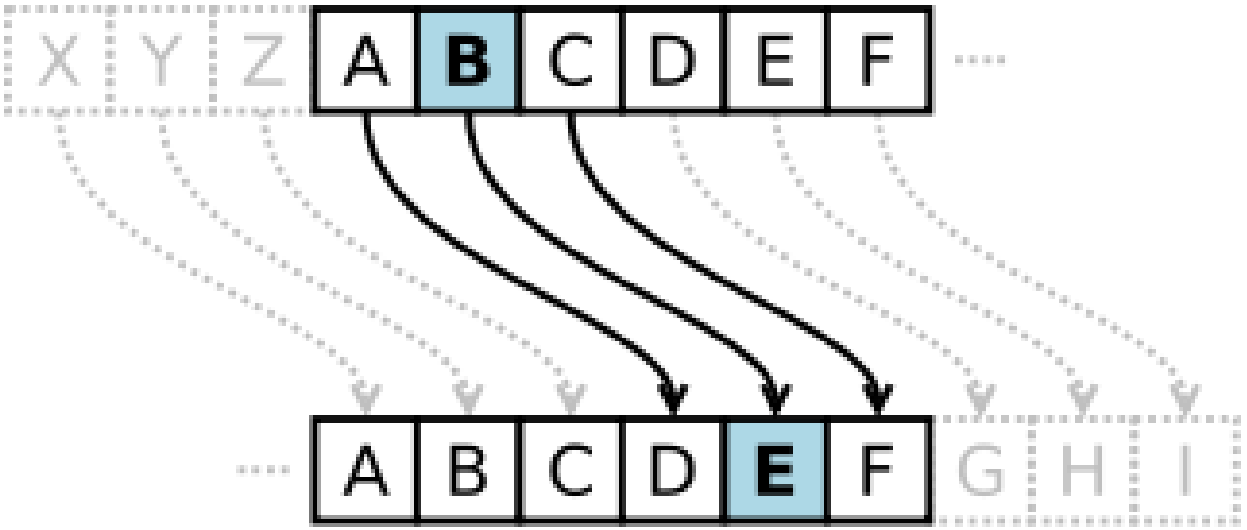
Criptografía Moderna: prácticamente toda la criptografía actual se basa en las teorías de la información y grandes números, la matemática discreta y la complejidad de los algoritmos, además de utilizar las operaciones clásicas de sustitución y transposición.

Criptografía clásica y moderna

Operaciones de Criptografía Clásica

- **Sustitución:** consiste en el *reemplazo* de las unidades del mensaje original según una determinada transformación.

Ejemplo: El cifrado del César

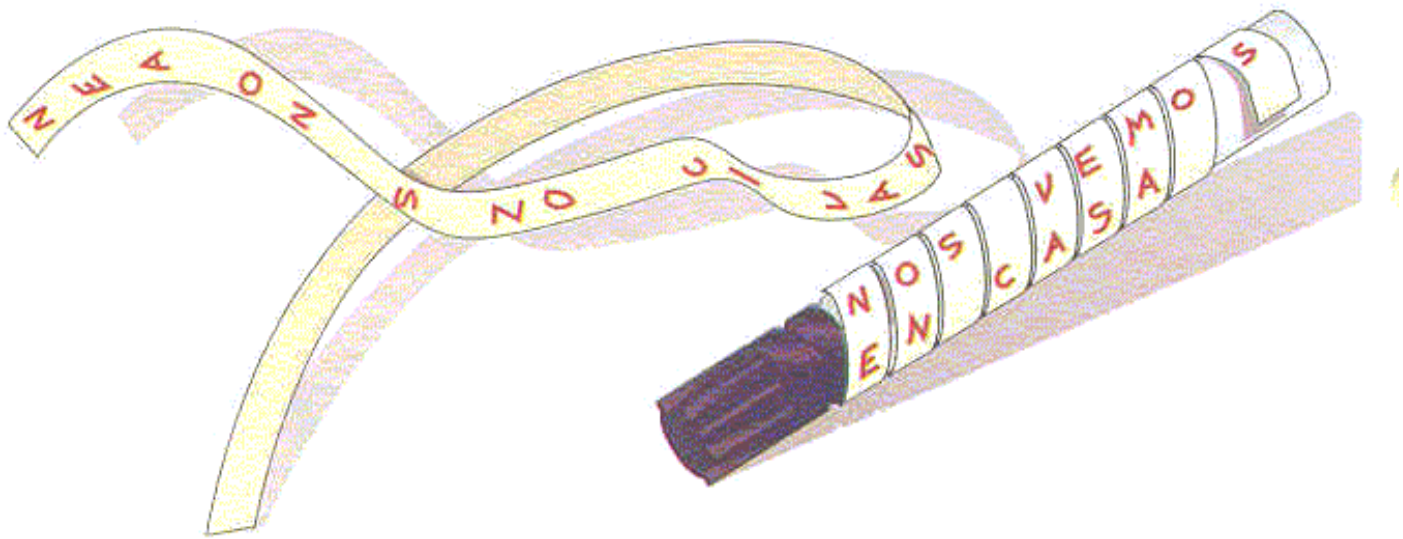


Criptografía clásica y moderna

Operaciones de Criptografía Clásica

- **Transposición:** consiste en el *reordenamiento* de las unidades del mensaje original según una determinada transformación.

Ejemplo: La Escítala

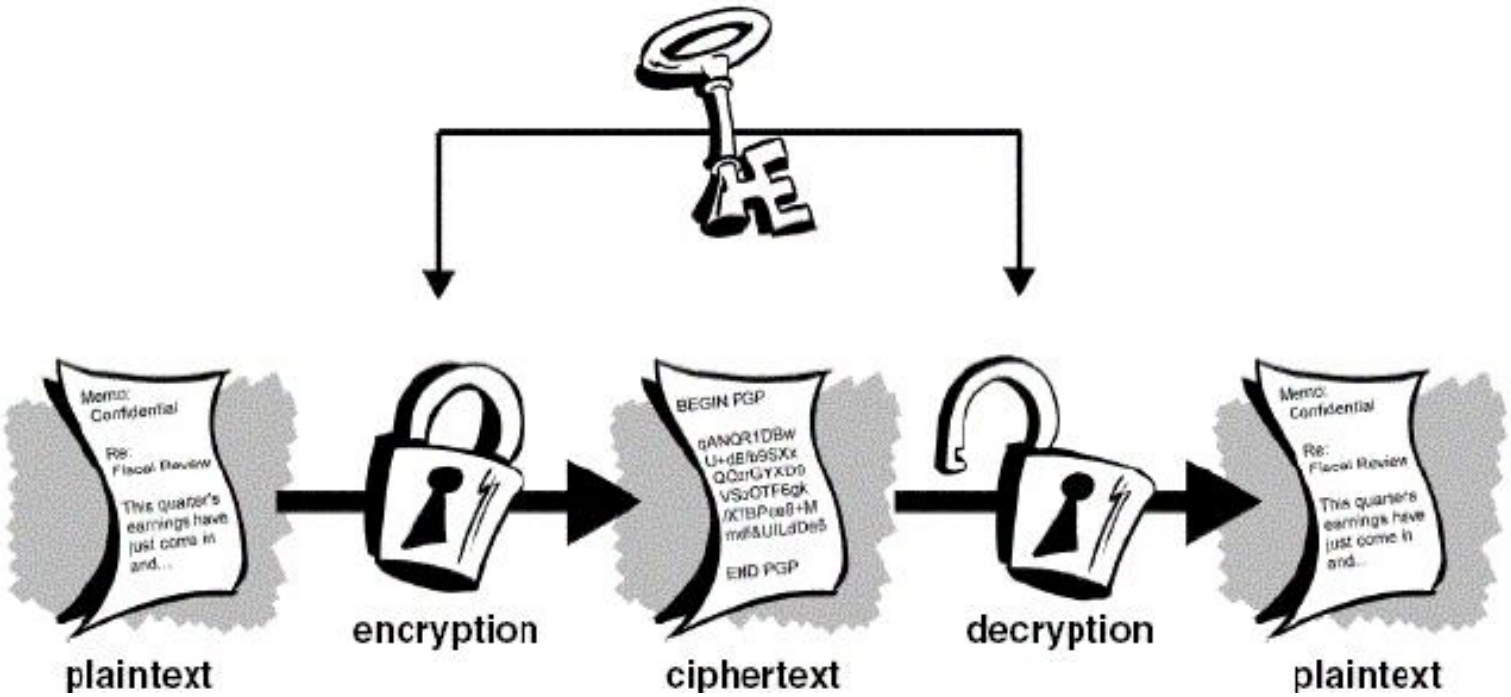


Criptografía Simétrica y Asimétrica

Criptografía Simétrica: utiliza la misma llave para cifrar y descifrar la información. Hablamos también de un algoritmo de cifrado simétrico o de clave privada.

Criptografía Asimétrica: se utiliza una llave para descifrar distinta de la empleada para cifrar. Ambas llaves se encuentran relacionadas matemáticamente, pero es computacionalmente imposible obtener una a partir de la otra. También se la llama criptografía de llave pública.

Criptografía Simétrica



Criptografía Simétrica

Algoritmos más utilizados

- **DES:** Es un algoritmo cifrador de bloques de 64 bits. La longitud de la llave es fija y de solamente 56 bits lo que permite encontrar la misma mediante fuerza bruta en cuestión de horas. No posee patentes.
- **Triple DES:** consiste en la aplicación del algoritmo DES 3 veces con 3 llaves distintas. Al emplear 3 llaves DES distintas, la longitud de la llave en TDES es fija y de 168 bits. La longitud efectiva de la misma es de 112 bits. No posee patentes.
- **AES (Advanced Encryption Standard):** Es un algoritmo cifrador de bloques de 128 bits con una llave de longitud variable de 128, 192 o 256 bits. Deriva de un algoritmo llamado Rijndael. No posee patentes.

Criptografía Simétrica

Algoritmos más utilizados

- **IDEA (International Data Encryption Algorithm):** Es un algoritmo cifrador de bloques de 64 bits, que utiliza una llave de longitud fija de 128 bits. Es un algoritmo sencillo, rápido y fácil de programar, sólo es de utilización libre para uso no comercial.
- **Blowfish:** Este algoritmo cifrador de bloques de 64 bits se diferencia de los anteriores en que utiliza una llave de longitud variable desde 32 hasta 448 bits (en saltos de 8 bits). La longitud por defecto es de 128 bits. Es más rápido que DES e inclusive IDEA. No posee patentes.
- **Twofish**: Cifrador de bloques de 128 bits que también utiliza llaves de longitud variable de 8 a 256 bits en múltiplos de 8 bits (128 bits por defecto). No posee patentes.

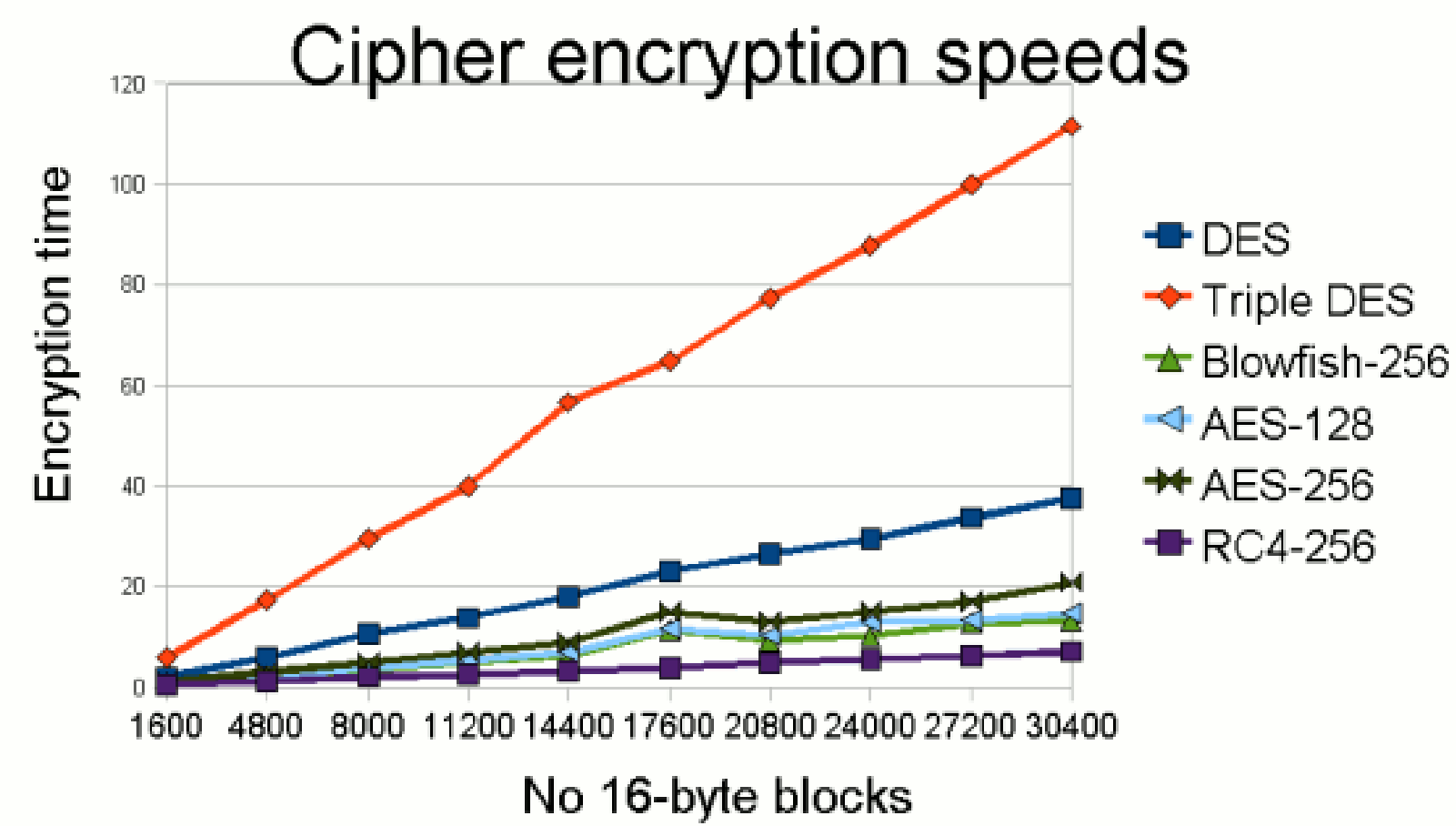
Criptografía Simétrica

Algoritmos más utilizados

- **CAST-256:** Cifrador de bloques de 128 bits que utiliza una llave de longitud variable de 128 a 256 bits en múltiplos de 32 bits. Está patentado y es de uso libre.
- **RC2:** Cifrador de bloques de 64 bits que soporta de 0 a 1024 bits de longitud de llave (128 bits por defecto) en múltiplos de 8 bits.
- **RC4:** Cifrador de flujo con longitudes de llave 8 a 2048 bits (128 bits por defecto) en múltiplos de 8 bits. Puede ser inseguro dependiendo del uso que se le dé.
- **ChaCha20:** Cifrador de bloques de 256 bits. Hasta 3 veces más rápido que AES, utilizado por Google, OpenSSH, etc. Disponible en el dominio público.

Criptografía Simétrica

Comparación de algoritmos

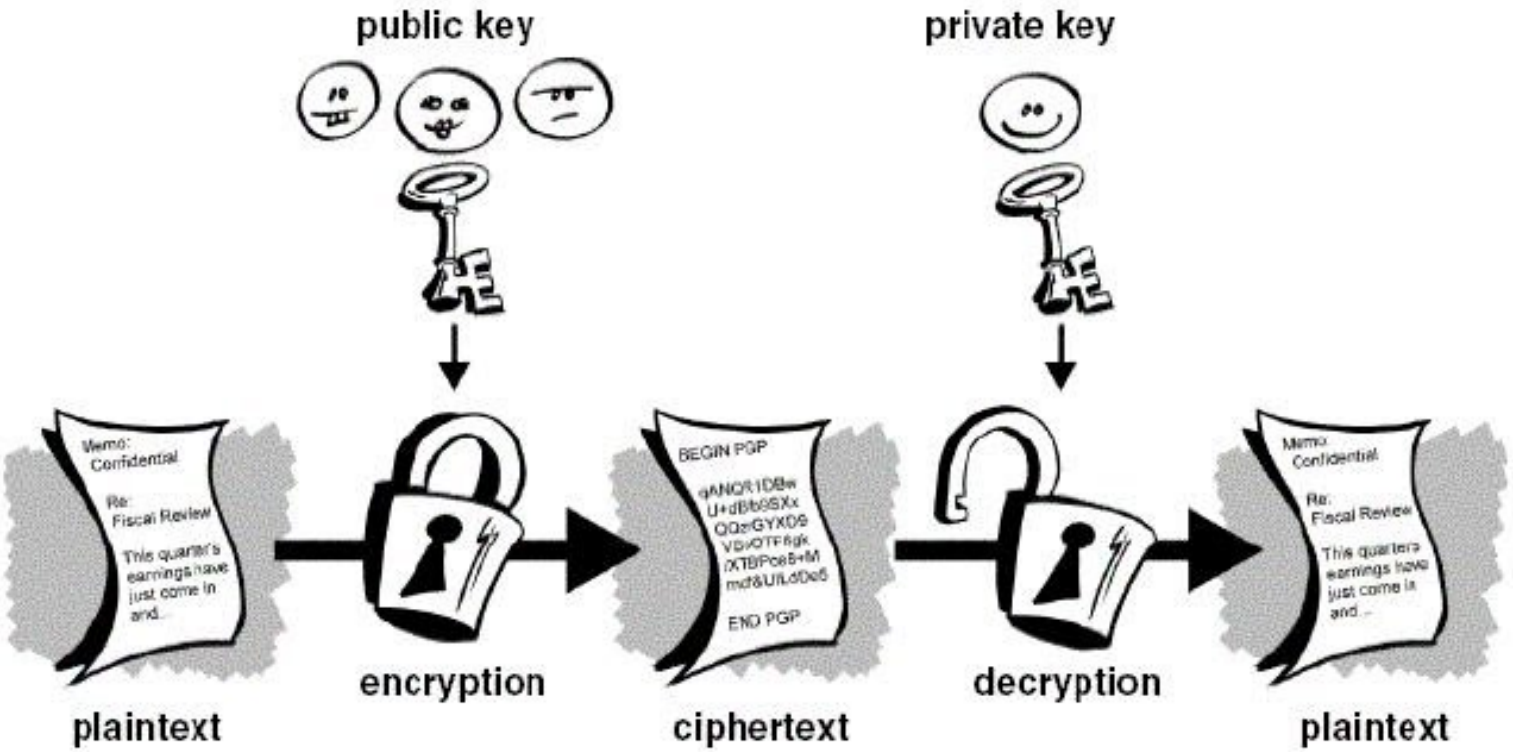


Criptografía Simétrica

Ventajas y desventajas

- **Ventajas:**
  - Los algoritmos simétricos son más rápidos.
  - El tamaño del mensaje cifrado es prácticamente el mismo que el del mensaje original.
  - Ideal para cifrar información que permanece almacenada y no necesita ser transmitida.
- **Desventajas:**
  - Intercambio de llaves en forma segura.
  - Cantidad de llaves necesarias cuando intervienen muchas personas: $n(n-1)/2$ llaves para cada par de personas.
  - No garantizan autenticidad y no repudio.

Criptografía Asimétrica



Criptografía Asimétrica
Usos

- **Confidencialidad:** utilizamos la **llave pública** del receptor para cifrar y luego él utilizará su propia llave privada para descifrarlo. De esta forma nos aseguramos que sólo el receptor autorizado pueda acceder nuevamente al texto plano.
- **Autenticación:** ciframos el mensaje utilizando la **llave privada** del emisor y cualquiera que posea la llave pública correspondiente podrá descifrarlo. El receptor se asegura de la integridad, autenticidad y no repudio del mensaje recibido.

Criptografía Asimétrica

Algoritmos más utilizados

- **DH (Diffie-Hellman):** fue el primer algoritmo asimétrico (1976). Es un método para intercambio de llaves que permite a dos extremos que no se conocen previamente, establecer una llave secreta compartida a través de un canal de comunicaciones inseguro. Esta llave puede ser luego utilizada para cifrar comunicaciones subsiguientes usando un algoritmo de cifrado simétrico.
- **RSA (Rivest, Shamir y Adelman):** data del año 1977. Este algoritmo sirve tanto para cifrar como para autentificar. Permite utilizar llaves de longitud variable, pero actualmente se aconsejan llaves mayores a 1024 bits.
- **Elgamal:** desarrollado por el Dr. Taher Elgamal en 1985. Este algoritmo se utiliza tanto para cifrar un mensaje como para firmar. El mensaje cifrado puede llegar a ocupar el doble de espacio que el mensaje original por lo que no se lo suele utilizar. No está patentado.
- **ECC (Elliptic Curve Cryptography):** algoritmo que utiliza puntos en una curva elíptica para definir las llaves pública/privada. Una llave de 256 bits ECC equivale a una de 3072 bits de RSA. Ideal para conexiones rápidas, seguras y con bajo uso de poder de cómputo (smartphones, tablets, etc).

Criptografía Asimétrica
Ventajas y desventajas

- **Ventajas:**
  - Los algoritmos asimétricos simplifican el intercambio de llaves en forma segura.
  - Se garantiza la integridad, autenticidad y no repudio.
- **Desventajas:**
  - Son mucho más lentos que los algoritmos simétricos.
  - Un mensaje cifrado con un algoritmo asimétrico ocupa más espacio que el original.

Cifradores de bloque y de flujo

Cifradores de Bloque: aplicamos la operación de cifrar (o descifrar) sobre bloques de tamaño fijo del mensaje. Por ejemplo, estos pueden ser bloques de 32, de 64 o de 128 bits.

Cifradores de Flujo: aplicamos la operación de cifrar (o descifrar) sobre cada elemento o caracter del mensaje. Normalmente por cada bit.

Cifradores de bloque y de flujo

Cifradores de bloque

- **Electronic Code Book (ECB):** los mensajes se dividen en bloques y cada uno de ellos es cifrado por separado. La desventaja de este método es que a bloques de texto plano idénticos les corresponde bloques idénticos de texto cifrado, de manera que se pueden utilizar estos patrones como guía para descubrir el texto plano a partir del texto cifrado.
- **Cipher Block Chaining (CBC):** a cada bloque de texto plano se le aplica la operación XOR con el bloque cifrado anterior y luego es cifrado. De esta forma, cada bloque de texto cifrado depende de todo el texto plano procesado hasta este punto. Esto evita la sustitución de un bloque individual dentro del mensaje.

Cifradores de bloque y de flujo

Cifradores de flujo

- **RC4:** Cifrador de flujo que soporta de 8 a 2048 bits de longitud de llave (128 bits por defecto) en múltiplos de 8 bits. Dependiendo del uso que se le dé, puede presentar algunas vulnerabilidades.
- **SEAL:** Cifrador de flujo muy veloz diseñado para equipos de 32 bits. Se encuentra patentado.

Funciones criptográficas de Hash

Función Hash: una función criptográfica de hash es un procedimiento determinístico que toma un bloque arbitrario de datos y devuelve una cadena de longitud fija (en bits), llamada valor hash, de modo tal que cualquier modificación a los datos hará que el valor hash cambie.

Funciones criptográficas de Hash

Propiedades de las funciones hash

1. **Unidireccionalidad:** dado un resumen $H(m_i)$, debe ser computacionalmente imposible encontrar m_i a partir de dicho resumen.
2. **Compresión:** a partir de un mensaje de cualquier longitud, el resumen $H(m_i)$ debe tener una longitud fija. Lo normal es que la longitud de $H(m_i)$ sea menor que el mensaje m_i.
3. **Facilidad de cálculo:** debe ser fácil calcular $H(m_i)$ a partir de un mensaje m_i.
4. **Difusión:** el resumen $H(m_i)$ debe ser una función compleja de todos los bits del mensaje m_i: si se modifica un solo bit del mensaje m_i, el hash $H(m_i)$ debería cambiar la mitad de sus bits aproximadamente.

Funciones criptográficas de Hash

Funciones hash más utilizadas

- **MD2 (Message Digest 2):** devuelve una salida fija de 128 bits para cualquier longitud del mensaje de entrada. La implementación de MD2 está optimizada para equipos de 8 bits. Es el más lento y no se recomienda su utilización en nuevas implementaciones.
- **MD4 (Message Digest 4):** devuelve una salida fija de 128 bits para cualquier longitud del mensaje de entrada. Es el más rápido de la familia. Es considerado actualmente inseguro y no se recomienda su utilización.
- **MD5 (Message Digest 5):** también devuelve una salida de longitud fija de 128 bits. Procesa los datos de entrada en bloques de 512 bits. Es uno de los algoritmos más utilizados aunque actualmente está en duda su seguridad y no se recomienda su utilización en nuevas implementaciones.

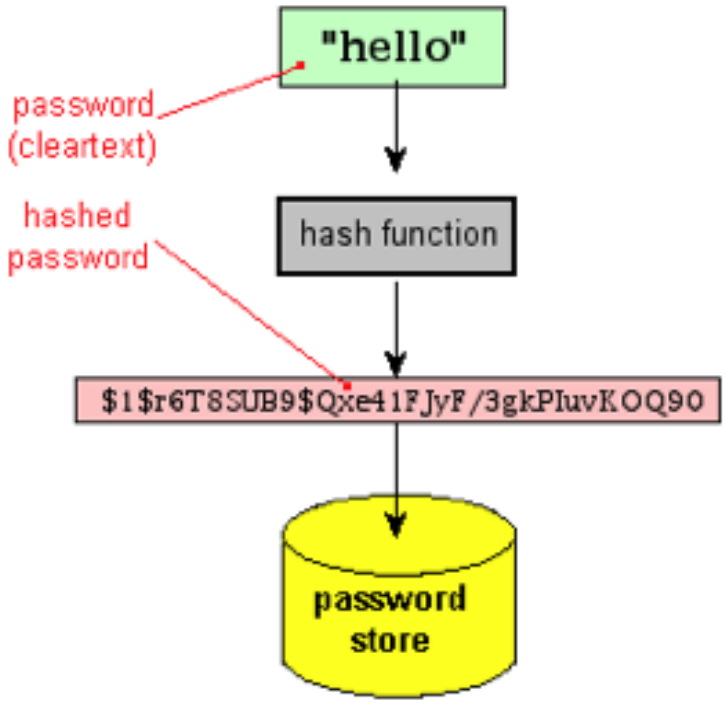
Funciones criptográficas de Hash

Funciones hash más utilizadas

- **MDC2 (Modification Detection Code 2):** desarrollado por IBM. Devuelve un compendio de 128 bits.
- **SHA-1 (Secure Hash Algorithm):** es un algoritmo que se utiliza en el SHS (Secure Hash Standard) y devuelve un compendio de 160 bits. Procesa los datos de entrada en bloques de 512 bits. Actualmente su seguridad está puesta en duda. Otros algoritmos de la misma familia son SHA-224, SHA-256, SHA-384 Y SHA-512, con salidas de 224, 256, 384 y 512 bits respectivamente.
- **RIPEMD-160 (RACE Integrity Primitives Evaluation Message Digest):** este es un algoritmo europeo que devuelve una salida de 160 bits. Existe otras versiones de este algoritmo de 128, 256 y 320 bits. RIPEMD-160 es más lento que SHA-1. No posee ninguna patente.
- **Tiger:** Devuelve una salida de 192 bits. Este algoritmo está diseñado para ser utilizado en plataformas de 64 bits.

Funciones criptográficas de Hash

Utilización en contraseñas de Sistemas Operativos



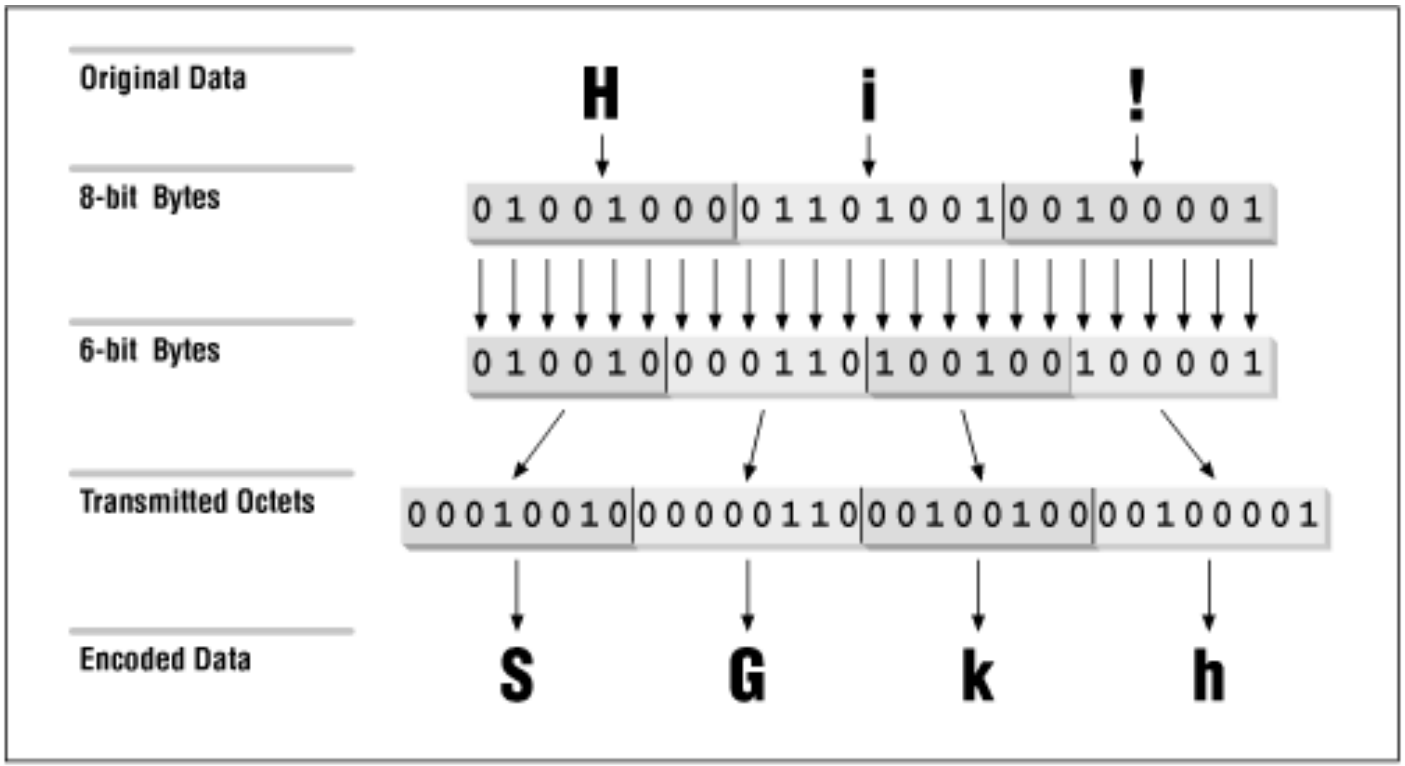
Funciones criptográficas de Hash

Utilización en contraseñas de Sistemas Operativos

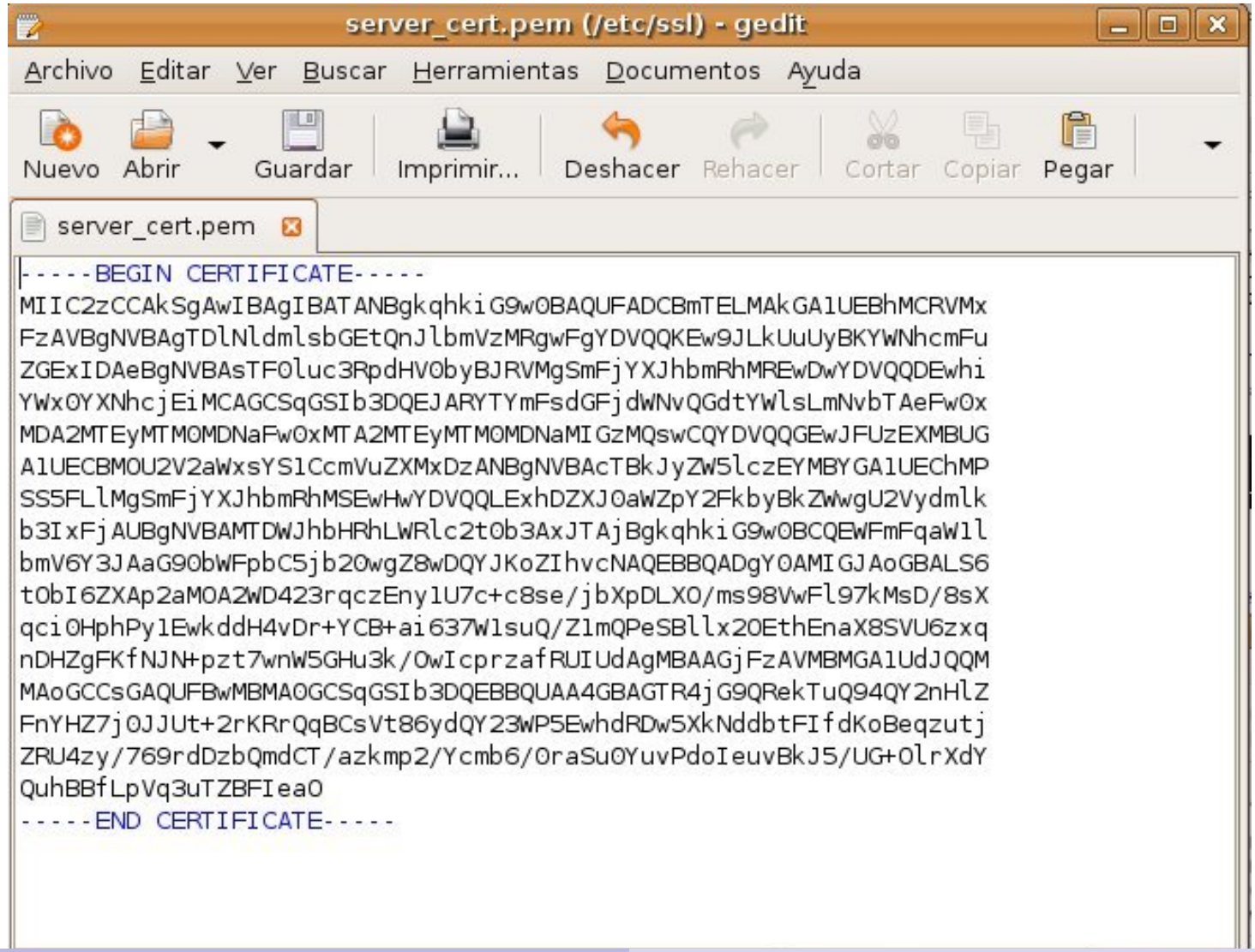
Sistema Operativo	**Algoritmo**	**Ubicación**
Linux (shadow)	Basado en MD5/SHA	/etc/shadow
Linux (no shadow)	Basado en DES	/etc/passwd
OpenBSD	Blowfish	/etc/master.passwd
Windows 3.x	Lan Manager (Basado en DES)	
Windows NT/2000/XP/2003	NTLM (Basado en MD4)	%SystemRoot%\ config\SAM

Codificación Base64

Base64: es un esquema de codificación para datos binarios que puede ser representado usando únicamente los caracteres imprimibles de ASCII.



Codificación Base64



OpenSSL

OpenSSL: es un proyecto colaborativo para desarrollar un kit Open Source de herramientas robusto, completo y de calidad empresarial que implemente los protocolos SSL (Secure Socket Layer) y TLS (Transport Layer Security) y un conjunto de librerías criptográficas de propósito general.

OpenSSL
Algoritmos y funciones implementados

Algoritmos simétricos

- AES128, AES192, AES256
- Blowfish
- CAST, CAST5
- DES, 3DES
- IDEA (Patentado)
- RC2, RC4, RC5 (Patentado)

Funciones de HASH:

- MD2, MD5
- MDC2 (Patentado)
- SHA, SHA-1
- RIPEMD-160

OpenSSL
Algunos ejemplos

Cifrar un archivo utilizando el algoritmo simétrico DES

```
$ openssl enc -e -in mensaje.txt -out mensaje.des -des
```

Cifrar un archivo utilizando el algoritmo simétrico AES128

```
$ openssl enc -e -in mensaje.txt -out mensaje.aes -aes128
```

Descifrar un archivo cifrado con el algoritmo DES

```
$ openssl enc -d -in mensaje.des -out mensaje.txt -des
```

Calcular el resumen de un archivo empleando el algoritmo MD5

```
$ openssl dgst -md5 mensaje.txt
```

Codificar en base 64 un archivo

```
$ openssl base64 -in mensaje.txt
```

Más información

- **LUCENA LÓPEZ, Manuel J.**. *Criptografía y Seguridad en Computadores*. https://blog.segu-info.com.ar/2020/01/libro-criptografia-y-seguridad-en.html
- **Wikipedia**. *Cryptography*. http://en.wikipedia.org/wiki/Cryptography
- **LYONS, James**. *Practical Cryptography. Ciphers.* http://practicalcryptography.com/ciphers/
- **Wikipedia**. *Base64*. http://en.wikipedia.org/wiki/Base64
- **GURUPRASAD, Gokul**. Cryptography and Data Security. Google Knol. http://knol.google.com/k/gokul-guruprasad/cryptography-and-data-security/26uzst0ozey3j/8
- **HEINLEIN, Paul.** *OpenSSL Command-Line HOWTO*. http://www.madboa.com/geek/openssl/
- **RAMIÓ AGUIRRE, Jorge.** *Libro Electrónico de Seguridad Informática y Criptografía.* Universidad Politécnica de Madrid. http://www.criptored.upm.es/guiateoria/gt_m001a.htm