Semana 13: Encriptación

Cifrado simétrico

Aprendizajes esperados

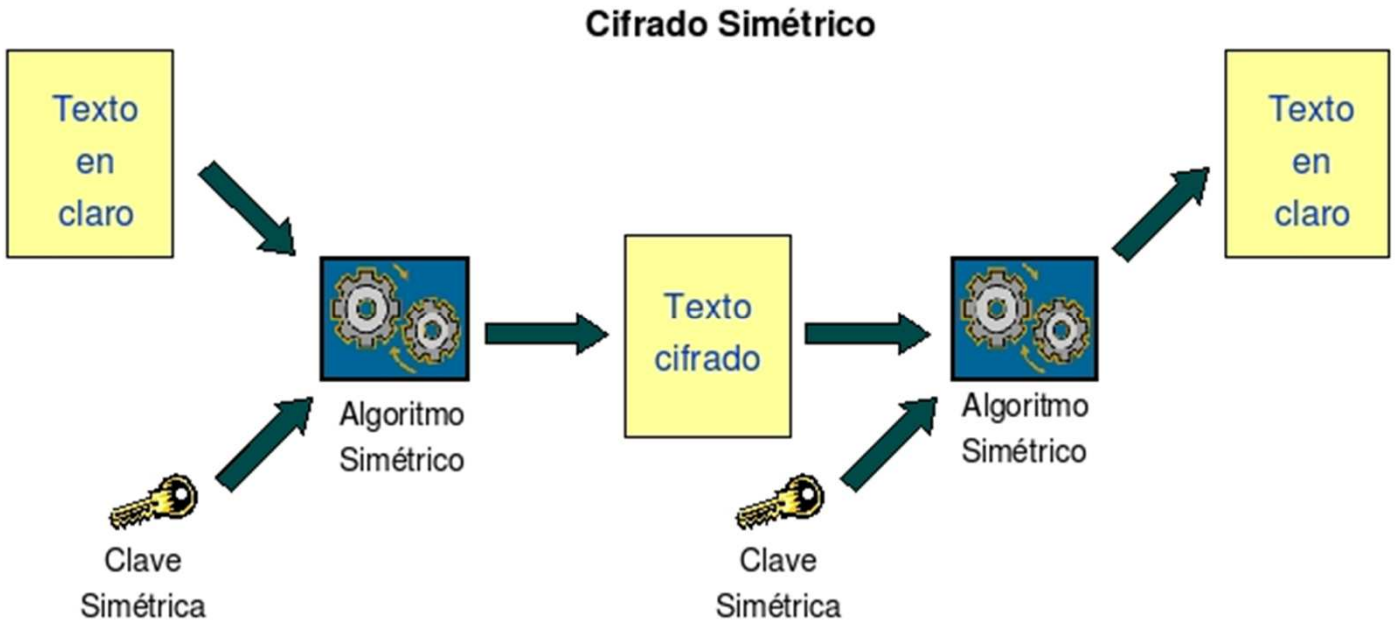
Contenidos:

- Características y principios del cifrado simétrico
- Algoritmos de cifrado simétrico

Encriptación Simétrica

- En la **encriptación simétrica** es utilizada una **clave única** (**clave privada**) tanto para encriptar como para desencriptar.
- Los **cifradores simétricos** también son llamados **cifradores de clave privada** o **de clave única**.

Encriptación Simétrica



Encriptación Simétrica

- Hay dos requerimientos para el uso seguro de la **encriptación simétrica**:
  - Un **algoritmo de encriptación fuerte**, este requerimiento significa que no debiera poderse descifrar el texto cifrado o descubrir la clave aun cuando se posean algunos textos cifrados junto con el texto plano que los produce.

Encriptación Simétrica

- Emisor y receptor deben haber obtenido copias de la clave secreta en forma segura y deben mantener la clave segura.

- La clave debe ser distribuida por un **canal seguro** entre el emisor y el receptor.

Encriptación Simétrica

- La seguridad del sistema depende de que nadie más conozca la clave. Esto tiene el problema de que hay que transmitirla al receptor en algún momento, y que puede ser interceptada.
- Algunos algoritmos de cifrado simétrico son: **DES**, **3DES**, **Blowfish**, **AES** y **CAST**.

Encriptación Simétrica: Tamaño de claves

- Hoy por hoy, los computadores pueden adivinar claves con extrema rapidez, y ésta es la razón por la cual el tamaño de la clave es importante en los criptosistemas modernos.
- El algoritmo de cifrado **DES** usa una clave de 56 bits, lo que significa que hay 2^{56} claves posibles.

Encriptación Simétrica: Tamaño de claves

- En este caso el espacio de posibilidades se puede comprobar en cuestión de días, si se trata de un computador de uso general a diferencia de una máquina especializada, que lo puede hacer en horas.
- Por otra parte, algoritmos de cifrado de diseño más reciente como **3DES**, **Blowfish** e **IDEA** usan todos claves de 128 bits, lo que significa que existen 2^{128} claves posibles.

Encriptación Simétrica: Tamaño de claves

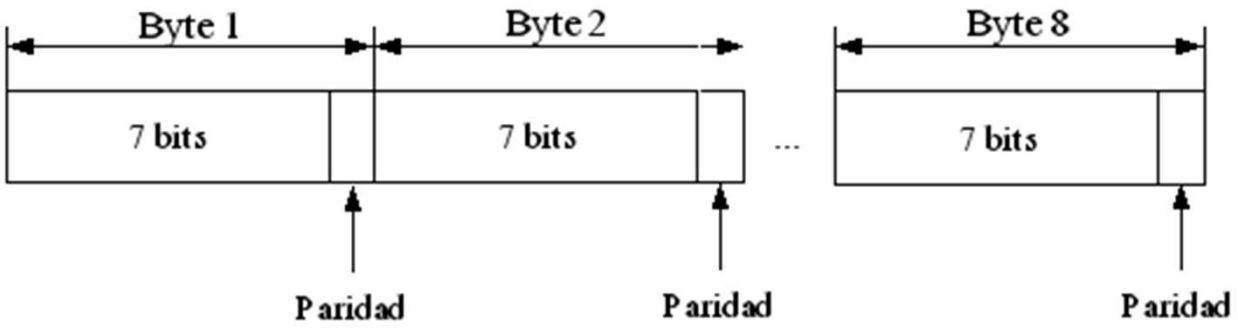
- Esto representa muchísimas claves, y aun en el caso de que todas las máquinas del planeta estuvieran cooperando, todavía tardarían más tiempo que la misma edad del universo en encontrar la clave.

DES

- Dentro de los cifradores de bloque, el más conocido es el **DES** o **Data Encryption Standar**.
- Este sistemas fue desarrollado a principio de los años 70 por un grupo de trabajo de IBM. En 1981 la ANSI aprobó el **DES** como estándar, el **X3.92**.
- Por su parte, la **ISO** hizo lo mismo en 1987 dándole el nombre de **DEA-1**.

DES

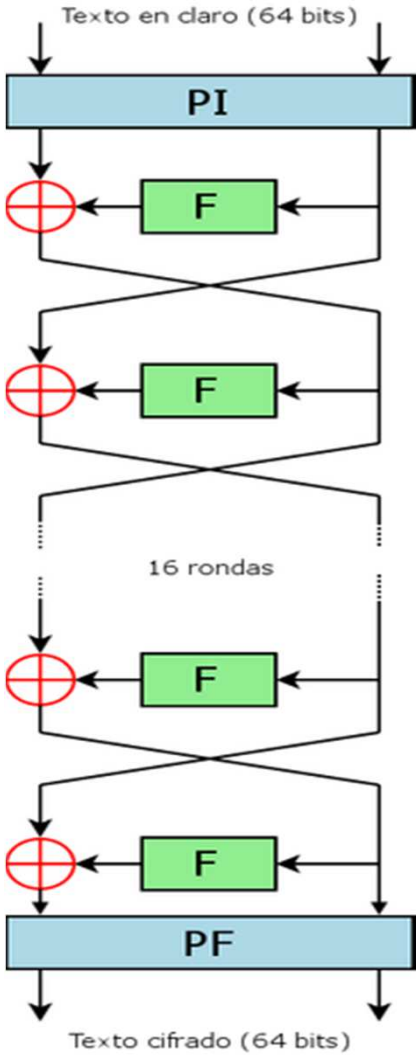
- El **DES** cifra y descifra bloques de 64 bits y lo somete a 16 rondas, con una clave de 64 bits (56 bits reales y 8 bits de control de paridad).



DES

- En la siguiente figura podemos ver la estructura básica de funcionamiento del algoritmo de **DES**.
- Hay **16 fases** idénticas o **rondas**. También hay una permutación inicial y final denominadas **PI** y **PF**, que son funciones inversas por lo que no son criptográficamente significativas.

DES


Texto en claro (64 bits)
PI
F
F
16 rondas
F
F
PF
Texto cifrado (64 bits)

DES

- Antes de las rondas, el bloque es dividido en dos mitades de 32 bits y procesadas alternativamente. Este entrecruzamiento se conoce como **esquema Feistel** y asegura que cifrado y descifrado sean procesos similares (la diferencia es que las subclaves se aplican en orden inverso al descifrar).

DES

- Esto simplificó la implementación sobre hardware, al no necesitar algoritmos distintos para cifrar y descifrar. El símbolo rojo "⊕" representa la operación OR exclusivo (XOR).
- La **función F** mezcla la mitad del bloque con parte de la clave. La salida de la **función F** se combina entonces con la otra mitad del bloque, y los bloques son intercambiados antes de la siguiente ronda.

DES

- Tras la última ronda, las mitades no se intercambian; ésta es la característica de la estructura de Feistel que hace que el cifrado y el descifrado sean procesos parecidos.

3DES

- Existen varias implementaciones en hardware de **DES** o de algunas de sus variantes. Una de estas variantes, es conocida como el **triple DES**, **3DES**.
- El funcionamiento del **3DES** consiste, básicamente, en la aplicación de tres veces consecutivas de **DES**, con tres claves distintas, aumentando, de este modo, el tamaño del espacio de claves.

3DES

- **3DES** ha sido ampliamente reconocido como seguro por ahora, aunque es bastante lento.
- La variante más simple de **3DES** funciona de la siguiente manera:

$$C = E_{DES}^{k_3}\left(D_{DES}^{k_2}\left(E_{DES}^{k_1}(M)\right)\right)$$

donde M es el mensaje a cifrar y k_1, k_2 y k_3 las respectivas claves **DES**.

Modos de operación de DES

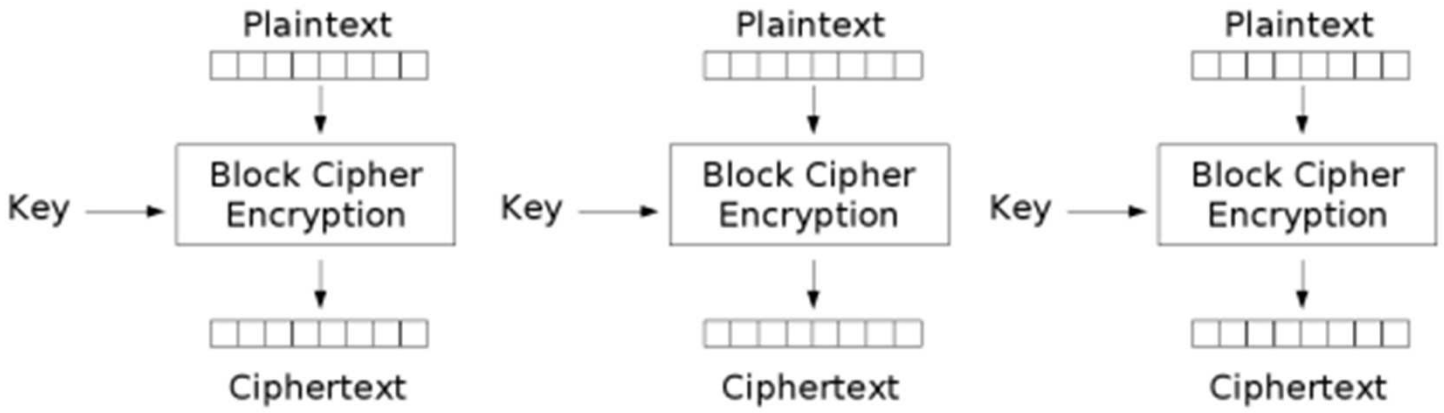
- Los cuatro modos que fueron diseñados por **DES** para aumentar la seguridad, pueden ser adaptados por cualquier cifrador en bloque.
  - **ECB** (**Electronic Codebook Mode**), para mensajes cortos.
  - **CBC** (**Cipher Block Chaining Mode**) para mensajes largos.
  - **CFB** (**Cipher Feedback Mode**) para cifrar bit por bit o byte por byte.
  - **OFB** (**Output Feedback Mode**) el mismo uso pero evitando propagacion.

Modos de operación de DES

- En el **modo ECB** el texto claro es dividido en bloques de 128 bits que se cifran uno a uno y por separado usando el AES. La concatenación de los bloques cifrados da lugar al texto cifrado.
- Este modo de funcionamiento tiene la ventaja de que funciona bien en canales con ruido. Un fallo en la transmisión tan solo afecta a un bloque de 128 bits, no al mensaje completo.

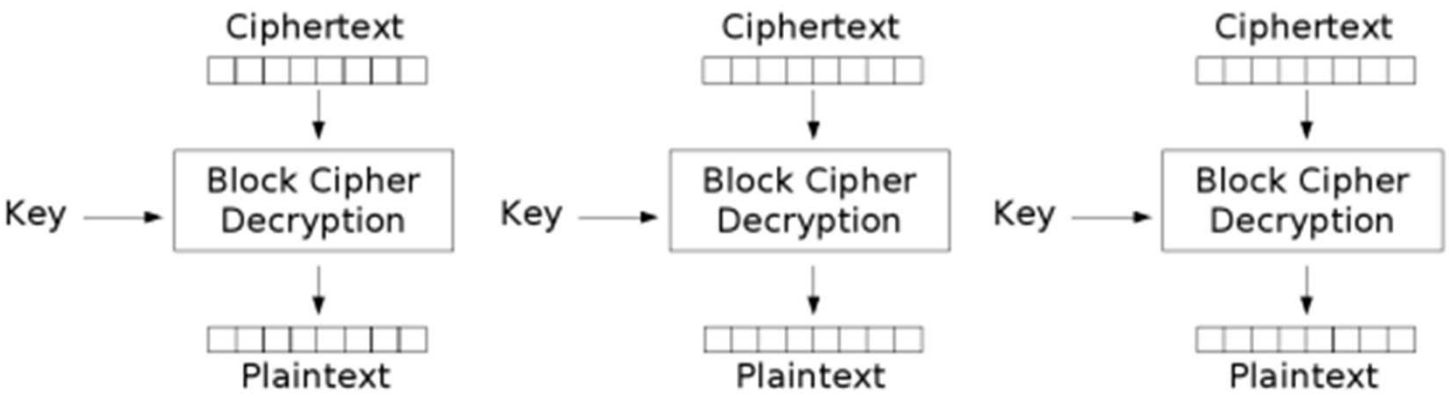
Modos de operación de DES

- Por otro lado, es susceptible a ataques estadísticos y/o ataques sobre la clave con un texto en claro conocido.



Electronic Codebook (ECB) mode encryption

Modos de operación de DES



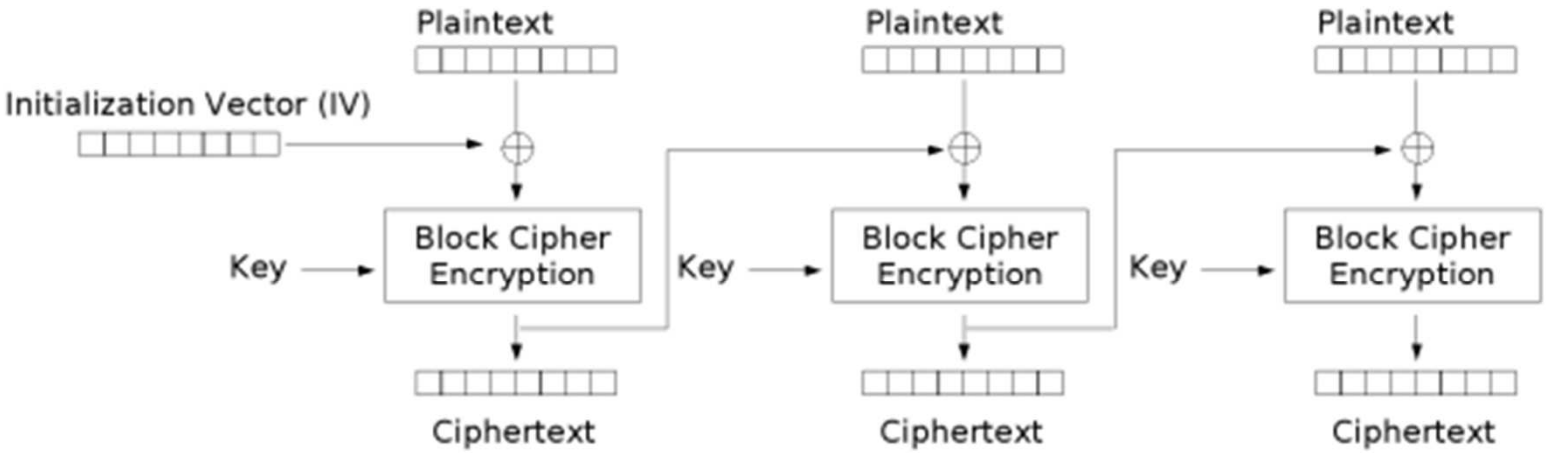
Electronic Codebook (ECB) mode decryption

Modos de operación de DES

- Una alternativa al **modo ECB** es el **modo CBC**, se divide el texto en bloques y se hace depender el bloque i-ésimo del texto cifrado/descifrado del (i-1)-ésimo:

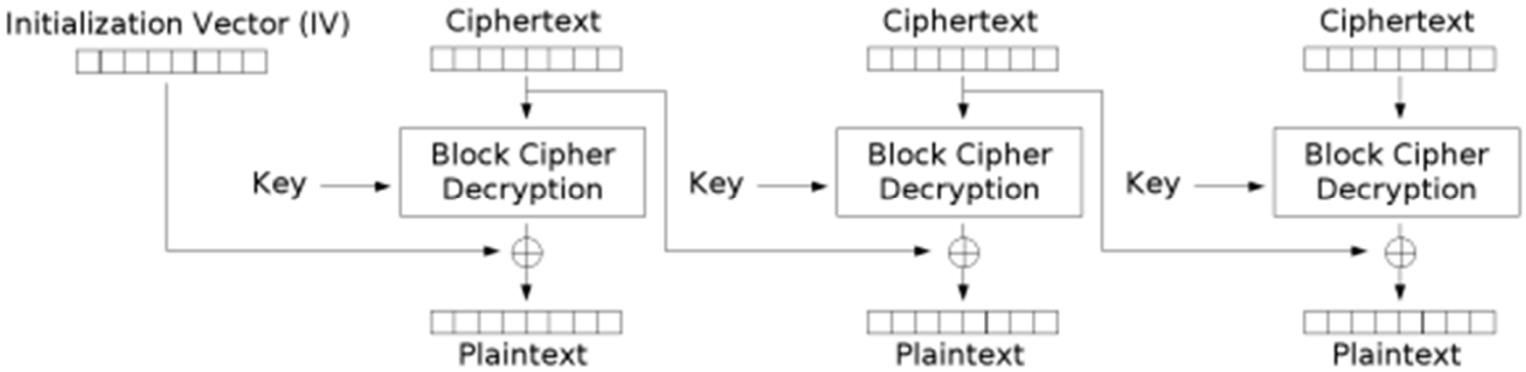
$$y_i = E_K(x_i \oplus y_{i-1}), \quad x_i = D_K(y_i) \oplus y_{i-1}$$

Modos de operación de DES



Cipher Block Chaining (CBC) mode encryption

Modos de operación de DES

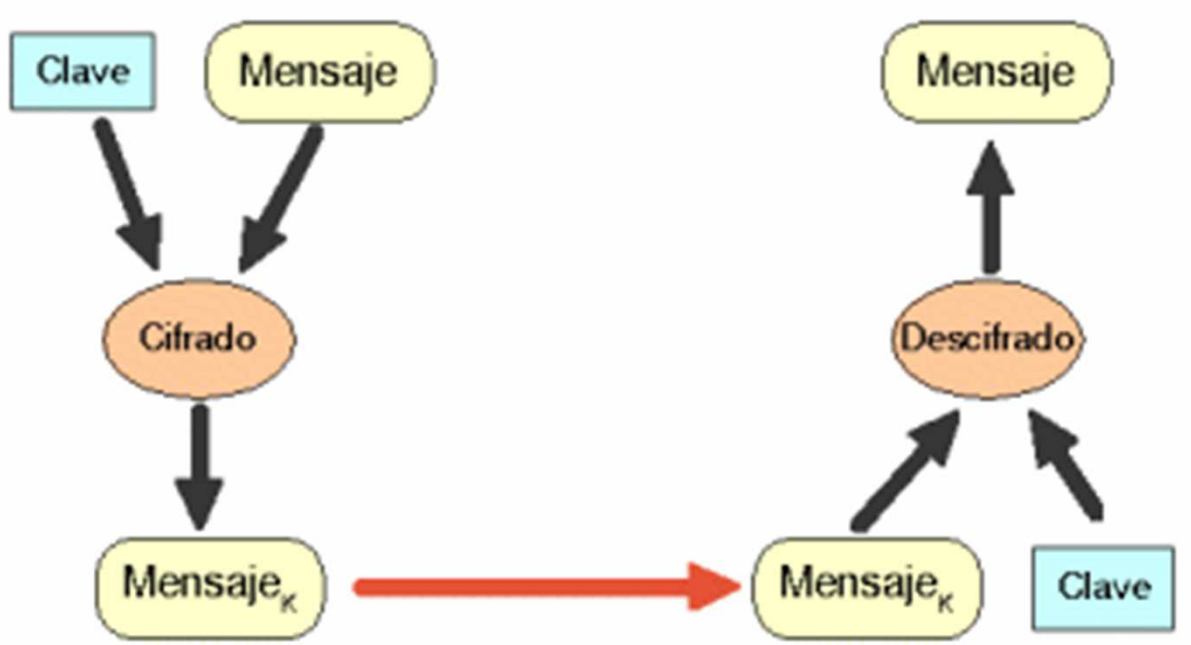


Cipher Block Chaining (CBC) mode decryption

Cifrado Simétrico con OpenSSL

- **OpenSSL** permite encriptación simétrica de datos a través de varios algoritmos y sus variantes.
- **OpenSSL** soporta varios algoritmos de clave simétrica como: **DES**, **3DES**, **Blowfish**, **AES** y **CAST**.
- Cada algoritmo de clave simétrica puede ser invocado desde la línea de comandos pasando el nombre del algoritmo en el comando **OpenSSL**.

Cifrado Simétrico con OpenSSL



Cifrado Simétrico con OpenSSL

- En la **encriptación simétrica** es utilizada una clave única (privada) tanto para encriptar como para desencriptar.
- La clave debe ser distribuida por un **canal seguro** entre el emisor y el receptor.
- La seguridad del sistema depende de que nadie más conozca la clave. Esto tiene el problema de que hay que transmitirla al receptor en algún momento, y que puede ser interceptada.

Encriptación Simétrica con OpenSSL

```
openssl enc [operación] [cifrador] -in
file.input -out file.output [campo clave]
[campo salt] [otras opciones]
```

Encriptación Simétrica con OpenSSL

- **Operación** hace referencia a encriptar (**-e**) y desencriptar (**-d**).
- **Cifrador** permite seleccionar el algoritmo a usar para la encriptación.
- **-in** file.input es el archivo a encriptar (texto claro).
- **-out** file.output es el resultado de la encriptación (texto cifrado).

Encriptación Simétrica con OpenSSL

- El campo "**salt**" es importante de utilizar pues previene el éxito de ataques de fuerza bruta y de diccionario.
- Usualmente aquí se utiliza la opción **-salt**.

```
# openssl   enc  -aes-192-ecb  -in
  fichero.txt -out encriptado.txt
```

Encriptación Simétrica con OpenSSL

- Para el caso de desencriptar un archivo cifrado es necesario conocer los detalles del cifrado que fueron seleccionados durante el proceso de encriptación.
- Por ejemplo, será necesario conocer: el cifrador utilizado, contraseña o clave, si el archivo incluye salt, el formato, etc.

Encriptación Simétrica con OpenSSL

```
# openssl enc  -d   -aes-192-ecb  -in
  cifrado.txt  -out fichero.txt
```

```
# openssl enc  -e  -aes-256-ecb  -salt  -
  in fichero.txt -out cifrado.txt
# openssl enc  -d   -aes-256-ecb  -in
  cifrado.txt  -out fichero.txt
```

Resumen

- La