

# SAD - U2 Criptografía

[Descargar estas actividades](#)

## Índice

### ▼ Actividades Criptografía

- [Actividad 1. Esteganografía](#)
- [Actividad 2. Criptografía híbrida](#)
- [Actividad 3. Funciones resumen](#)
- [Actividad 4. Algoritmos simétricos](#)
- [Actividad 5. Algoritmos asimétricos](#)
- [Actividad 6. Algoritmos](#)
- [Actividad 7. Cifrado simétrico con OpenSSL](#)
- [Actividad 8. Funciones resumen con OpenSSL](#)
- [Actividad 9. Comparativa de algoritmos de cifrado simétrico](#)
- [Actividad 10. Interpretación de errores de cifrado](#)
- [Actividad 11. Interpretación de errores de funciones resumen](#)
- [Actividad 12. Propuesta de mejora de seguridad](#)
- [Actividad 13. Cifrado asimétrico con OpenSSL](#)
- [Actividad 14. Esquema híbrido: cifrar simétricamente y firmar \(qué se protege\)](#)
- [Actividad 15. Esquema híbrido práctico: cifrar con AES, proteger la clave con RSA y firmar el paquete](#)

### ▼ Casos prácticos

- [Caso práctico 1. Cifrado de unidades de almacenamiento](#)
- [Caso práctico 2. Uso de OpenSSL en entornos reales](#)

## Actividades Criptografía

### Actividad 1. Esteganografía

La esteganografía es la práctica de ocultar un mensaje secreto dentro (o incluso encima) de algo que no es secreto. Ese algo puede ser casi cualquier cosa que desees. Por ejemplo; ocultar un mensaje secreto o un script dentro de un documento de Word o Excel.

Hay tres aspectos principales para la ocultación de la información:

- **Capacidad:** cantidad de información que se puede ocultar.
- **Seguridad:** Dificultad para detectar información oculta.
- **Solidez:** modificaciones que el medio de cobertura puede resistir antes de que la información oculta se corrompa.

Los principales tipos de esteganografía son:

- **Pura:** no requiere el intercambio de un cifrado como un stego-key. Se asume que ninguna otra parte tiene conocimiento de la comunicación.
- De **clave secreta:** la clave secreta (stego) se intercambia antes de la comunicación. Esto es más susceptible a la interceptación. Solo las partes que conocen la clave secreta pueden revertir el proceso y leer el mensaje secreto.

- De **clave pública**: se utiliza una clave pública y una clave privada para una comunicación segura. El remitente utilizará la clave pública durante el proceso de codificación y solo la clave privada, que tiene una relación matemática directa con la clave pública, puede descifrar el mensaje secreto.

En [StegOnline](#) puedes encontrar una guía de las diferentes técnicas esteganográficas y ejemplos de cómo se pueden utilizar.

- a) Descubre los secretos ocultos en las técnicas que tienen una "Custom Image para probar"

## Actividad 2. Criptografía híbrida

La criptografía simétrica es más insegura ya que el hecho de pasar la clave es una gran vulnerabilidad, pero se puede cifrar y descifrar muy rápidamente grandes cantidades de datos. Ese es el principal inconveniente de la criptografía asimétrica. Para aprovechar lo mejor de las dos técnicas, existe la criptografía híbrida.

Para ello, la comunicación se divide en dos partes: **intercambio de claves** y **intercambio de mensajes**.

El proceso para usar un sistema criptográfico híbrido es el siguiente (para enviar un archivo cifrado a un compañero):

### Intercambio de claves

1. Generar una clave pública y otra privada (en el receptor): `privHibrida_tunombre.pem` y `pubHibrida_tunombre.pem`.
2. Guardar una clave simétrica en un archivo: `claveSimetrica.txt`.
3. Cifrar el archivo `claveSimetrica.txt` con la clave pública de un compañero: `claveSimetrica.aes`.
4. Enviar el archivo cifrado `claveSimetrica.aes` al compañero al que pertenece la clave pública que hemos usado para cifrar.

### Intercambio de mensajes

1. Descifrar el archivo recibido `claveSimetrica.aes` con nuestra clave privada.
2. Obtener la clave simétrica `claveSimetrica.txt`. **Ahora los dos sabemos la clave simétrica.**
3. Cifrar el archivo `claveSimetrica.txt` con un algoritmo de clave simétrica, usando la clave recibida: `claveSimetrica2.aes`.
4. Enviar el archivo cifrado `claveSimetrica2.aes` al compañero que generó la clave.
5. El compañero descifra `claveSimetrica2.aes` con la clave que creó al principio: `claveSimetrica2.txt`.
6. Compara la clave obtenida con la que generó inicialmente.
7. Si son iguales, ya puede realizar un intercambio de archivos, sin importar el tamaño de los mismos, usando criptografía simétrica, con la clave intercambiada.

a) Realiza y documenta gráficamente este proceso con un compañero de clase. Incluye capturas de pantalla de los archivos generados y de los pasos realizados.

Las capturas de pantalla deben incluir elementos que permitan identificar que tú has realizado el proceso, como el nombre de la máquina, un nombre de directorio, etc.

## Actividad 3. Funciones resumen

Crea una tabla con las funciones resumen más utilizadas en criptografía.

La tabla debe tener al menos la siguiente información:

- Nombre del algoritmo
- Fecha de creación

- Tamaños de salida
- Uso más común
- Segura: Si se ha encontrado alguna vulnerabilidad en el algoritmo y cuándo se descubrió
- Ejemplo de uso con un texto de ejemplo (el mismo texto para todas las funciones) con OpenSSL
- Otros datos que consideres relevantes

## Actividad 4. Algoritmos simétricos

Crea una tabla con los algoritmos simétricos más utilizados en criptografía.

La tabla debe tener al menos la siguiente información:

- Nombre del algoritmo
- Fecha de creación
- Tamaños de clave
- Uso más común
- Seguro: Si se ha encontrado alguna vulnerabilidad en el algoritmo y cuándo se descubrió
- Ejemplo de uso con un texto de ejemplo (el mismo texto para todos los algoritmos) con OpenSSL
- Otros datos que consideres relevantes

## Actividad 5. Algoritmos asimétricos

Crea una tabla con los algoritmos asimétricos más utilizados en criptografía.

La tabla debe tener al menos la siguiente información:

- Nombre del algoritmo
- Fecha de creación
- Tamaños de clave
- Uso más común
- Seguro: Si se ha encontrado alguna vulnerabilidad en el algoritmo y cuándo se descubrió
- Ejemplo de uso con un texto de ejemplo (el mismo texto para todos los algoritmos) con OpenSSL
- Otros datos que consideres relevantes

## Actividad 6. Algoritmos

Completa la siguiente tabla marcando una X en la columna correspondiente a las funcionalidades que proporciona cada algoritmo.

Algoritmo	C. Simétrico	C. Asimétrico	F. Resumen	Passwd Linux	Passwd Windows	Firma digital
RSA						
AES						
SHA-256						
MD5						
3DES						
DSA						

Algoritmo	C. Simétrico	C. Asimétrico	F. Resumen	Passwd Linux	Passwd Windows	Firma digital
Blowfish						
El-Gamal						
ECC						

## Actividad 7. Cifrado simétrico con OpenSSL

a) Define con tus propias palabras qué es el cifrado simétrico y menciona un algoritmo que se puede usar con OpenSSL para este tipo de cifrado.

Pon un ejemplo de uso de ese algoritmo con OpenSSL.

b) Escribe el comando básico que utilizarías en OpenSSL para cifrar un archivo usando AES-256-CBC y explica para qué sirve cada parámetro del comando.

c) Escribe el comando básico que utilizarías en OpenSSL para descifrar el archivo anterior y explica para qué sirve cada parámetro del comando.

d) Pásale a tu compañero de la derecha (si eres el último al primero de tu fila) el archivo cifrado para que intente descifrarlo sin pasarle la clave.

e) Si no ha podido descifrarlo, pásale ahora la clave que has usado para cifrarlo. Pídele que lo descifre y te muestre el archivo descifrado. Comprueba que el archivo descifrado es igual al original.

## Actividad 8. Funciones resumen con OpenSSL

a) Explica qué función tiene el comando openssl dgst -sha256 cuando se aplica a un archivo.

b) ¿Para qué sirve calcular un hash de un archivo?

c) Calcula el hash SHA-256 y MD5 de un archivo de texto (actividad8.txt) con OpenSSL. Compara los hashes generados. Guarda los resultados en los archivos actividad8.sha256 y actividad8.md5 respectivamente.

d) Modifica el archivo de texto (actividad8\_d.txt) y vuelve a calcular los hashes.

Guarda los resultados en los archivos actividad8\_d.sha256 y actividad8\_d.md5 respectivamente.

¿Qué diferencias encuentras respecto a los hashes generados en el paso anterior?

## Actividad 9. Comparativa de algoritmos de cifrado simétrico

Cifra el mismo archivo usando dos algoritmos diferentes (por ejemplo AES-256-CBC y DES-CBC) y realiza las siguientes tareas:

1. Compara el tamaño de los archivos cifrados resultantes
2. Identifica las diferencias en los comandos utilizados para cada algoritmo
3. Explica las ventajas y desventajas que observas entre ambos algoritmos

## Actividad 10. Interpretación de errores de cifrado

Intenta descifrar el archivo cifrado con AES-256-CBC en la actividad anterior realizando las siguientes modificaciones y explica qué sucede en cada caso:

- Usar una contraseña incorrecta
- Modificar un byte del archivo cifrado antes de descifrarlo

Explica por qué ocurren los errores observados y qué implican para la seguridad del cifrado

## Actividad 11. Interpretación de errores de funciones resumen

Dados dos archivos diferentes pero con contenido similar (por ejemplo, el mismo texto con un espacio extra), realiza las siguientes tareas:

- Calcula el hash SHA-256 de ambos archivos
- a) Explica por qué los hashes son diferentes aunque el contenido sea casi idéntico
- b) Describe qué características de las funciones hash se evidencian en este ejemplo

## Actividad 12. Propuesta de mejora de seguridad

Imagina que necesitas proteger un documento importante con cifrado simétrico y asegurarte además de que no ha sido alterado.

- a) Describe qué técnicas usarías para conseguir tu objetivo.
- b) Especifica los comandos OpenSSL que necesitas, explicando cada paso y comando.

## Actividad 13. Cifrado asimétrico con OpenSSL

- a) Generar par RSA y firmar (SHA-256), firmar el fichero con la clave privada y verificar la firma con la clave pública.
- b) Verificar que la firma falla si se modifica el mensaje

Verificar firma con el fichero original, modificar el fichero y volver a verificar (debe fallar).

Realiza cada actividad, guarda los ficheros generados (clave\_privada.pem, clave\_publica.pem, mensaje.sig) anota los comandos y captura la salida de los distintos comandos.

## Actividad 14. Esquema híbrido: cifrar simétricamente y firmar (qué se protege)

Pasos:

1. Generar par de claves asimétricas (RSA 2048 bits).
2. Cifra un fichero de imagen (jpg, png) con AES usando una contraseña/clave y firma tanto el archivo original como el cifrado en archivos separados.
3. Verifica ambas firmas y explica qué garantiza cada firma (firmar plaintext vs firmar ciphertext) y cuándo es preferible cada opción.

Documenta los comandos usados y los ficheros generados con capturas de pantalla de la salida de los comandos usados.

## Actividad 15. Esquema híbrido práctico: cifrar con AES, proteger la clave con RSA y firmar el paquete

**Objetivo:** aplicar un flujo real (clave simétrica protegida con clave pública; firma del ciphertext) y reproducir el proceso inverso.

Pasos:

1. **Pasos emisor:** generar clave simétrica (texto) y cifrar con AES (password desde archivo)
2. Cifrar la clave simétrica con la clave pública del receptor
3. Firmar el archivo cifrado con la clave privada del emisor
4. **Pasos receptor:** Descifrar la clave simétrica
5. Verificar la firma del archivo cifrado
6. Descifrar el mensaje usando la clave recuperada
7. Comprobar que el mensaje descifrado es igual al original

Archivos generados:

- clave.txt (original)
- mensaje.txt (original)
- emisor\_priv.pem (clave privada emisor)
- emisor\_pub.pem (clave pública emisor)
- mensaje.enc (cifrado)
- clave.txt.enc (clave simétrica cifrada)
- mensaje.enc.sig (firma del mensaje cifrado)
- key.txt (clave simétrica recuperada)
- mensaje.dec (mensaje descifrado)

Documenta los comandos usados y los ficheros generados con capturas de pantalla de la salida de los comandos usados.

## Casos prácticos

### Caso práctico 1. Cifrado de unidades de almacenamiento

**Bitlocker** es una herramienta de cifrado de unidades de almacenamiento que se utiliza en sistemas Windows.

En sistemas Linux, se puede utilizar **LUKS** (Linux Unified Key Setup) para cifrar unidades de almacenamiento o bien **Dislocker** para acceder a unidades cifradas con Bitlocker en sistemas Linux.

Otra herramienta interesante es **VeraCrypt**, una herramienta de cifrado de código abierto que se puede utilizar en Windows, Linux y macOS.

a) Investiga cómo usar Dislocker para cifrar una unidad USB con Bitlocker y cómo montarla en un sistema Linux para acceder a los datos de la unidad cifrada. Este es un mecanismo muy útil para compartir datos entre sistemas Windows y Linux de forma segura.

b) Documenta los pasos necesarios para crear un contenedor cifrado con VeraCrypt en Windows y montarlo en una unidad del sistema.

### Caso práctico 2. Uso de OpenSSL en entornos reales

OpenSSL es una herramienta muy utilizada en entornos reales para gestionar certificados digitales, cifrar comunicaciones y proteger datos sensibles.

a) Investiga cómo se utiliza OpenSSL para generar certificados digitales autofirmados y cómo se pueden utilizar estos certificados para cifrar comunicaciones en un servidor web (por ejemplo, un servidor Apache o Nginx).

- b) Documenta los pasos necesarios para generar un certificado autofirmado con OpenSSL y configurarlo en un servidor web para habilitar HTTPS.