



UNIVERSIDAD  
NACIONAL  
AUTÓNOMA  
DE MÉXICO

# PRACTICAL SESSION 4

## SIGNING, ENCRYPTING AND HASHING FILES

ASIGNATURA:  
Criptografía

GRUPO: 2

NOMBRE:

Reyes Mendoza  
Miriam Guadalupe

FECHA: 17/05/24



FACULTAD DE INGENIERÍA

# FIRMA, CIFRADO Y HASH DE ARCHIVOS

## FIRMA DE ARCHIVO

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.3593]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\miria>cd C:\Users\miria\Desktop\Cryptography\04 - Hashing Files

C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>type private_key.pem
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAQgSkAgEAAoIBAQDPiX5EDeDgldVY
dNzsyxGVhUsWEhD2ht9FU27TDgpI77n7Gx82a0y0lzS1zg0aHLZkdoCXVqBKBb9i
MJ5p3Hmmtutec5C31CrKSQK5EZH0743XAXYDntNhLVXqZ9Y6C/vjXhiEkIaWC8u
tFeHt5IrpHu5wHxDNR+QT82Q2MxIGA+g5ffVUiT751s0px0V8Ph6i5M3JcA7pemn
5KT6+asbYEXn80fsLDXQfIurgYmOEde5PhZtkYF0ywp7Tpy1w7JDIiy0V5wiGXM
bwMDQw68G0qDJgW+/K4hBGaRR10/7soA3JNCm5lif0sQ0c50nYjS1kYEiRC21y0z
YciWrKC1AgMBAECggEAHswPN++cS4+Ig2nan+v5gcfsLswDBuZYSwwYwcqPVaf3
YB6i2trip2DZ0LH0QIgau9LG905ETn/AWQSQeXKdfctk6H8PD+beLgWYPGKIGGAQ
Vdw6xMdPXl8/XtEdw55/AIwsjVcaxpiH9eU0qSXhU8cekH/BgOKo+qwQ+/nkBJ2C
9HszmLF2tXLEBng7JyyLLai+S/2ntSDNEGeHr07NjDSCAnj1MrgOkMdvmdE0SRjm
QU8Yf19qtRYM0TW4+YcEpQzRVY4FNIkNWWQDEP339KsgGGY5iqWnuHiefG6EgPDK
YRu3Z2Hk35SCjKRuNQ8GUvsrCcnJ04MdR6sttfiRxeQKBgQDyAmRMPBXzGGgekAsi
kGqQdsvMaCS96kbBPXjkch3LRjoCgC77GAADa4nmGK0ML79BfDOWz/mKofcPr09d
PXu8gLtQ5tSB4uqYpJJ3yjT4eP7h6BNB7d0hiRTRbASEutbI73XVUS4dRa0uS41L
8ai8TN+a45I26Jj7K5f9HjIiTQKBgQDbi042TGcSSIoWuJsLE9b1k105PHGsgNUP
pz6tg4WtGCKN8zBLEtrN9K3hZ/sfeN1dzg3x9hX98RBPBKobJPgRq1XLsV5Vq5QQT
ppyL7vLVHKF/nkGuWNqSmV/s5Ig0ejBwKnlSkXJfCTg06qC640bqCPRTdCyuCX3H
1uj8ok0cCQKBgQDA4LcfKyQzQtH4w0Scn4z05L824immUvlh0aUILQsh3y6Mn6zR
4ibgkuysvi3s8ID1qA9VuxufKcbGb4W2QjJD4jB0pvnnKvrd/Qve/QGKpqQ9V8FB
YlPDSAWXgwd24ZIdQvwFAh+Gp0mJZB2Q0WEx8fKkd4sox+hLBjF0QoSiYQKBgQDF
3lQ+UnywCPuVYQw1XstnikwC84zNW00uD6K5h4qrrqdQPA7rqX3MhIMC8IhLAK5KK
IteE1zWZLXmNqEPWDHS0yfAMPyCo0QSEpa4BngHSibyyJrSybNmM0nswLABrrZn1
3BZr+WgpakjvwPnfmEvpUurMcTB4KIva28QhODPh6QKBgFL+gsH1bRTMYeira4sP
qu/1hjGTPrfdm6PUynv+CCOC192eG+GydpRhR9nKvJXsm7a6ByT+sdon7ifL7L4s
Q488CROKFXj4UwT5R55vVwLDDfQz8aJgH/iNqDYXT9AtHVDfieRr4iwLQquarrzY
8UpHQFFPGOT1wruVqxy+Ra1
-----END PRIVATE KEY-----

C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>
```

`type` es un comando de Windows que se utiliza para mostrar el contenido de un archivo de texto. En este caso, `type private_key.pem` imprimirá el contenido del archivo `private_key.pem` en la consola.

Este archivo `private_key.pem` es una clave privada generada por OpenSSL. Las claves privadas se utilizan en criptografía asimétrica para descifrar datos o firmar digitalmente documentos. Deberías tener cuidado al manejar claves privadas, ya que cualquier persona con acceso a tu clave privada puede descifrar tus datos cifrados o firmar documentos en tu nombre.

El texto es una clave privada en formato PEM (Privacy Enhanced Mail). Este formato se utiliza comúnmente para almacenar y transmitir claves criptográficas. La clave privada se utiliza en la criptografía de clave pública para descifrar los datos cifrados con la clave pública correspondiente o para firmar digitalmente los datos.

El contenido entre -----BEGIN PRIVATE KEY----- y -----END PRIVATE KEY----- es la clave privada codificada en base64. Esta codificación permite que los datos binarios se representen en un formato que se puede imprimir y enviar de manera segura a través de canales que están diseñados para manejar texto.

```
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>type public_key.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAz4l+RA3g4JXVcnTc7MsR
lYVLFhIQ9obfRVNu0w4KS0+5+4MfNmjstJc0tc4Dmhy2ZHaAl1agSgW/YjCeadx5
poLbrXn0Qt9QqykkCuRGR90+N1wF2A57TYS1V6mFW0gv7414YhJCGlgvLrRXh7eS
K6R7ucB8QzUfKE/NkNjMSBgPoOX31VIk++dbDqcTlfD4eouTNyXA06Xpp+Sk+vmr
G2BF5/NH7Cw10HyLq4GJjhHXnuT4WbZGBdMsKe06ctc0yQyIstFecIhlzG8DA0MO
vBtKgyYFvvyuIQRmkUdTv+7KANYTQpuZYnzrENH0dJ2I0tZGBIkQtctctM2HIlqyg
tQIDAQAB
-----END PUBLIC KEY-----

C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>
```

Este archivo `public_key.pem` es una clave pública generada por OpenSSL. Las claves públicas se utilizan en criptografía asimétrica para cifrar datos o verificar firmas digitales. A diferencia de las claves privadas, las claves públicas pueden ser distribuidas libremente sin comprometer la seguridad. Por lo tanto, es seguro compartir tu clave pública con otros.

La clave pública se utiliza en la criptografía de clave pública para cifrar los datos que solo pueden ser descifrados con la clave privada correspondiente o para verificar las firmas digitales creadas con la clave privada correspondiente.

```
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>type texto.txt
Reyes Mendoza Miriam Guadalupe
Criptografía - Grupo 02
Facultad de Ingeniería

Practical Session 4 - Signing, Encrypting and Hashing Files

Este es un archivo de texto de ejemplo para probar las utilidades de OpenSSL
.
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>
```

El comando `type` en el script de shell que has seleccionado es un comando de Windows que se utiliza para mostrar el contenido de un archivo de texto. En este caso, `type texto.txt` imprimirá el contenido del archivo `texto.txt` en la consola. En este ejemplo particular, el archivo contiene un texto de ejemplo que vamos a cifrar. Es una forma sencilla y efectiva de verificar el contenido del archivo, asegurarse de que el archivo correcto está siendo utilizado, o simplemente para leer el contenido sin modificarlo.

```

C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>openssl dgst -sha256
-sign private_key.pem -out firma.sha256 texto.txt

C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>type firma.sha256
B:~pµÛ]ÚiqwÇ/¶|>í@ç³{ÃÇŕ>†^††^+û÷'8%ç Lŕì³-D,íŕó÷u¹

5ÜÐß@0JêMmFÍ°7°ç&~o·
Áİç±£ö/£$&f¶|WŦâ>±¹ÜlÓ_Dä¼×°F1ŕ©)¶+bú=0zè
0Bŕ:~*Ãİ'Dî(Yµa } )ÖöÖŕ+ðËN!NHæquŕCŕ»°æ
İ†ðŕŦŸ÷&Ld(uêŕotçR- 6ŸjIá]=µ#ÃJARóÃ@ÖŸŕİ
ŦŦD%<0BŕpŸ"1z_e`Gŕpî&"ôŕæŕe}3¹†İ""v7ŕ·
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>

```

El comando `openssl dgst -sha256 -sign private_key.pem -out firma.sha256 texto.txt` genera una firma digital del archivo `texto.txt` utilizando la clave privada almacenada en `private_key.pem`.

- `openssl`: Es la herramienta de línea de comandos para la biblioteca de criptografía OpenSSL. Se utiliza para realizar diversas operaciones criptográficas.
- `dgst -sha256`: `dgst` es un comando de OpenSSL para calcular el resumen de un mensaje (también conocido como hash). `-sha256` especifica que se debe usar el algoritmo SHA-256 para calcular el resumen del mensaje.
- `-sign private_key.pem`: `-sign` es una opción que indica que OpenSSL debe firmar el resumen del mensaje utilizando la clave privada especificada. En este caso, la clave privada está en el archivo `private_key.pem`.
- `-out firma.sha256`: `-out` es una opción que especifica el archivo de salida donde se debe guardar la firma digital. En este caso, la firma se guardará en el archivo `firma.sha256`.
- `texto.txt`: Este es el archivo de entrada del que se calculará el resumen del mensaje. En este caso, el resumen se calculará del contenido del archivo `texto.txt`.

En resumen, este comando calcula el resumen SHA-256 del archivo `texto.txt`, firma ese resumen con la clave privada en `private_key.pem` y guarda la firma en `firma.sha256`. El comando `type firma.sha256` muestra el contenido del archivo `firma.sha256` que contiene la firma generada. El texto que resulta es la firma digital del archivo `texto.txt`.

Esta firma puede ser verificada por alguien que tenga la clave pública correspondiente a la clave privada utilizada para firmar el archivo. La verificación confirmará que el archivo `texto.txt` no ha sido alterado desde que fue firmado y que la firma fue creada por alguien que posee la clave privada.

## VERIFICACIÓN DE ARCHIVO

```
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>openssl dgst -sha256
-verify public_key.pem -signature firma.sha256 texto.txt
Verified OK
```

El comando `openssl dgst -sha256 -verify public_key.pem -signature firma.sha256 texto.txt` verifica una firma digital del archivo `texto.txt` utilizando la clave pública almacenada en `public_key.pem`.

- `dgst -sha256`: `dgst` es un comando de OpenSSL para calcular el resumen de un mensaje (también conocido como hash). `-sha256` especifica que se debe usar el algoritmo SHA-256 para calcular el resumen del mensaje.
- `-verify public_key.pem`: `-verify` es una opción que indica que OpenSSL debe verificar la firma del resumen del mensaje utilizando la clave pública especificada. En este caso, la clave pública está en el archivo `public_key.pem`.
- `-signature firma.sha256`: `-signature` es una opción que especifica el archivo que contiene la firma digital que se va a verificar. En este caso, la firma está en el archivo `firma.sha256`.
- `texto.txt`: Este es el archivo de entrada del que se calculará el resumen del mensaje. En este caso, el resumen se calculará del contenido del archivo `texto.txt`.

En resumen, este comando calcula el resumen SHA-256 del archivo `texto.txt`, verifica la firma en `firma.sha256` con la clave pública en `public_key.pem` y devuelve si la firma es válida o no. Al ser la firma es correcta, se indica en la salida del comando.

## CIFRADO DE ARCHIVO AES-256

```
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>openssl enc -aes-256-cbc -salt -in texto.txt -out cifrado.txt -k miriam1234 -pbkdf2

C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>type cifrado.txt
Salted__d5j5ó}-YMLrBuĐiHöRÄ%áA7Óik~@V,C[ãÓ÷Äü=_a,%+hÎ^ú|ÊmL£H
îîé«êK7-ú'÷Êââø
LWS=D-ÖJENüÿ
òîl$Z@t-;
Ôâ%y`39u|°smD!vTxEÄãëøYÔBèÉm■3-H!||7ê7^3l9öi5èPS6nxãóÙDE~TuisTÀ<¤xÂq|ääâûrä
1ÊÉTPÁE0^~V||v-Ê-TòøôM%_Tç
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>
```

El comando `openssl enc -aes-256-cbc -salt -in texto.txt -out cifrado.txt -k miriam1234 -pbkdf2` cifra el archivo `texto.txt` utilizando el algoritmo AES-256-CBC y la contraseña `miriam1234`.

- `enc -aes-256-cbc`: `enc` es un comando de OpenSSL para cifrar o descifrar archivos. `-aes-256-cbc` especifica que se debe usar el algoritmo AES (Advanced Encryption Standard) con una clave de 256 bits y en modo CBC (Cipher Block Chaining) para el cifrado.
- `-salt`: Esta opción indica que se debe usar un "salt" para mejorar la seguridad del cifrado. Un "salt" es un valor aleatorio que se utiliza como entrada adicional al cifrado para evitar ataques de diccionario y de tabla de arco iris.
- `-in texto.txt`: `-in` es una opción que especifica el archivo de entrada que se va a cifrar. En este caso, el archivo de entrada es `texto.txt`.
- `-out cifrado.txt`: `-out` es una opción que especifica el archivo de salida donde se debe guardar el texto cifrado. En este caso, el texto cifrado se guardará en el archivo `cifrado.txt`.
- `-k miriam1234`: `-k` es una opción que especifica la contraseña que se utilizará para generar la clave de cifrado. En este caso, la contraseña es `miriam1234`.
- `-pbkdf2`: Esta opción indica que se debe usar PBKDF2 (Password-Based Key Derivation Function 2) para derivar la clave de cifrado de la contraseña. PBKDF2 es una función que se utiliza para derivar una clave de una contraseña y un "salt".

En resumen, este comando cifra el archivo `texto.txt` utilizando AES-256-CBC con la contraseña `miriam1234` y PBKDF2 para la derivación de la clave, y guarda el texto cifrado en `cifrado.txt` cuyo contenido se muestra posteriormente. Lo siento, pero el texto proporcionado parece ser una salida cifrada o codificada y no es legible como texto plano. El texto cifrado resultante se guarda en el archivo `cifrado.txt`. Este archivo contiene datos binarios, por lo que, al intentar abrirlo se verán caracteres no legibles.

## DESCIFRADO DE ARCHIVO AES-256

```
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>openssl enc -d -aes-256-cbc -in cifrado.txt -out descifrado.txt -k miriam1234 -pbkdf2

C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>type descifrado.txt
Reyes Mendoza Miriam Guadalupe
Criptografía - Grupo 02
Facultad de Ingeniería

Practical Session 4 - Signing, Encrypting and Hashing Files

Este es un archivo de texto de ejemplo para probar las utilidades de OpenSSL

C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>
```

El comando `openssl enc -d -aes-256-cbc -in cifrado.txt -out descifrado.txt -k miriam1234 -pbkdf2` descifra el archivo `cifrado.txt` que ha sido cifrado previamente utilizando el algoritmo AES-256-CBC y la contraseña `miriam1234`.

- `enc -d -aes-256-cbc`: `enc` es un comando de OpenSSL para cifrar o descifrar archivos. `-d` es una opción que indica que OpenSSL debe descifrar el archivo de entrada. `-aes-256-cbc` especifica que se debe usar el algoritmo AES (Advanced Encryption Standard) con una clave de 256 bits y en modo CBC (Cipher Block Chaining) para el descifrado.
- `-in cifrado.txt`: `-in` es una opción que especifica el archivo de entrada que se va a descifrar. En este caso, el archivo de entrada es `cifrado.txt`.
- `-out descifrado.txt`: `-out` es una opción que especifica el archivo de salida donde se debe guardar el texto descifrado. En este caso, el texto descifrado se guardará en el archivo `descifrado.txt`.

- `-k miriam1234`: `-k` es una opción que especifica la contraseña que se utilizará para generar la clave de descifrado. En este caso, la contraseña es `miriam1234`.
- `-pbkdf2`: Esta opción indica que se debe usar PBKDF2 (Password-Based Key Derivation Function 2) para derivar la clave de descifrado de la contraseña. PBKDF2 es una función que se utiliza para derivar una clave de una contraseña.

En resumen, este comando descifra el archivo `cifrado.txt` utilizando AES-256-CBC con la contraseña `miriam1234` y PBKDF2 para la derivación de la clave, y guarda el texto descifrado en `descifrado.txt`. El texto de salida proporcionado indica que el proceso de descifrado fue exitoso y el texto original ha sido recuperado.

## HASH DE ARCHIVO SHA2 DE 512 BITS

```
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>openssl dgst -sha512
-hex texto.txt
SHA2-512(texto.txt)= 2c81fabfe7cb733c99922abb1abbce559836ba0684e932506c6e88d
fe933aab09bd9c701ebe19f3a4579b6538fde2dd9cd4306cec58588da9f3b9fa5ad3b8da0
```

El comando `openssl dgst -sha512 -hex texto.txt` de OpenSSL calcula el resumen criptográfico (también conocido como hash) de un archivo utilizando el algoritmo SHA-512.

- `dgst -sha512`: `dgst` es un comando de OpenSSL para calcular resúmenes criptográficos. `-sha512` especifica que se debe usar el algoritmo SHA-512 para calcular el resumen. SHA-512 es un algoritmo de la familia SHA-2 que produce un resumen de 512 bits.
- `-hex`: Esta opción indica que el resumen calculado debe mostrarse en formato hexadecimal.

En resumen, este comando calcula el resumen SHA-512 del archivo `texto.txt` y muestra el resultado en formato hexadecimal. Este hash es único para el contenido del archivo `texto.txt` en el momento en que se calculó el hash. Cualquier cambio, incluso el más mínimo, en el contenido del archivo `texto.txt` resultará en un hash completamente diferente si se vuelve a calcular.



Los hashes son útiles para verificar la integridad de los datos, ya que puedes calcular el hash de los datos en dos momentos diferentes y comparar los hashes para ver si los datos han cambiado.

```
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>openssl dgst -sha512
-sign private_key.pem -out hash.sha512 texto.txt

C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>type hash.sha512
S^_Ty%y!f÷òuj-AXQ°T±ù]·7N²ÖhÊ)¾-U!¿Rf±ûÐ|#i6U«ð9»Â£oaJ̄U0nŋjqC9q'ûá|f-õô-Ä1ÃÈ
Ä"FuówhÚúWVSQç@æû²■'l2J_e;nè£Ä||TJ\ ½\j!Ö5=ô+Z¥½ñT
x  £éYãð5vIð0ßL̄Nð¶±9ø¿ŋ¹i
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>|
```

El comando `openssl dgst -sha512 -sign private_key.pem -out hash.sha512 texto.txt` genera una firma digital de un archivo utilizando una clave privada y el algoritmo SHA-512.

- `dgst -sha512`: `dgst` es un comando de OpenSSL para calcular resúmenes criptográficos. `-sha512` especifica que se debe usar el algoritmo SHA-512 para calcular el resumen. SHA-512 es un algoritmo de la familia SHA-2 que produce un resumen de 512 bits.
- `-sign private_key.pem`: `-sign` es una opción que indica que OpenSSL debe firmar el resumen calculado utilizando la clave privada especificada. En este caso, la clave privada se encuentra en el archivo `private_key.pem`.
- `-out hash.sha512`: `-out` es una opción que especifica el archivo de salida donde se debe guardar la firma digital. En este caso, la firma se guardará en el archivo `hash.sha512`.

En resumen, este comando calcula el resumen SHA-512 del archivo `texto.txt`, firma el resumen con la clave privada en `private_key.pem` y guarda la firma en `hash.sha512`.

El texto de salida después del comando `type`, es la firma digital del archivo `texto.txt`. Esta firma puede ser verificada por alguien que tenga la clave pública correspondiente a la clave privada utilizada para firmar el archivo. La verificación confirmará que el archivo `texto.txt` no ha sido alterado desde que fue firmado y que la firma fue creada por alguien que posee la clave privada.

## VERIFICACIÓN HASH

```
C:\Users\miria\Desktop\Cryptography\04 - Hashing Files>openssl dgst -sha512  
-hex -verify public_key.pem -signature hash.sha512 texto.txt  
Verified OK
```

El comando `openssl dgst -sha512 -hex -verify public_key.pem -signature hash.sha512 texto.txt` verifica una firma digital de un archivo utilizando una clave pública y el algoritmo SHA-512.

- `dgst -sha512`: `dgst` es un comando de OpenSSL para calcular resúmenes criptográficos. `-sha512` especifica que se debe usar el algoritmo SHA-512 para calcular el resumen. SHA-512 es un algoritmo de la familia SHA-2 que produce un resumen de 512 bits.
- `-hex`: Esta opción indica que el resumen calculado debe mostrarse en formato hexadecimal.
- `-verify public_key.pem`: `-verify` es una opción que indica que OpenSSL debe verificar la firma del resumen utilizando la clave pública especificada. En este caso, la clave pública se encuentra en el archivo `public_key.pem`.
- `-signature hash.sha512`: `-signature` es una opción que especifica el archivo que contiene la firma digital que se va a verificar. En este caso, la firma se encuentra en el archivo `hash.sha512`.

En resumen, este comando calcula el resumen SHA-512 del archivo `texto.txt`, verifica la firma en `hash.sha512` con la clave pública en `public_key.pem` y muestra el resultado en formato hexadecimal. Si la verificación es exitosa, OpenSSL imprimirá *"Verified OK"* en la consola, como se ve. Si la verificación falla, imprimirá *"Verification Failure"*.

## CONCLUSIÓN

Cada uno de estos comandos cumple una función específica en el manejo de archivos y criptografía usando OpenSSL en Windows. Estos comandos permiten firmar y verificar archivos, cifrar y descifrar textos, y generar y verificar hashes, proporcionando una solución completa para la seguridad y la integridad de datos.

Durante esta práctica, se aprendió a firmar un archivo de texto con una clave privada y verificarlo con una clave pública, asegurando que el contenido no ha sido alterado y proviene de una fuente autenticada. La firma digital asegura la autenticidad del archivo, mientras que la verificación con la clave pública confirma la integridad del archivo firmado.

Además, se exploró el cifrado de un archivo de texto usando el algoritmo AES-256, haciéndolo ilegible para cualquier persona que no posea la clave correcta. Posteriormente, se descifró el archivo cifrado para devolverlo a su forma original, demostrando cómo se puede proteger la confidencialidad de la información durante su almacenamiento o transmisión.

También se comprendió cómo generar un hash SHA-512 de un archivo, creando una huella digital única que puede utilizarse para verificar la integridad del contenido. La firma del hash con una clave privada proporciona una capa adicional de seguridad, garantizando que el hash no ha sido alterado. La verificación del hash firmado con una clave pública asegura la autenticidad y la integridad del archivo.

Esta práctica proporciona una comprensión profunda de los mecanismos fundamentales de la criptografía aplicada a archivos de texto, esenciales para asegurar datos en entornos corporativos y de desarrollo, y para cualquier persona que trabaje con información sensible o requiera autenticación y verificación de datos. Con estas habilidades, se garantiza un nivel adicional de seguridad y confianza en el manejo de información digital.

## REFERENCIAS

- OpenSSL Wiki. *Command Line Utilities*.  
[https://wiki.openssl.org/index.php/Command Line Utilities](https://wiki.openssl.org/index.php/Command_Line_Uutilities)