

Actividad 16. Cifrado con Openssl

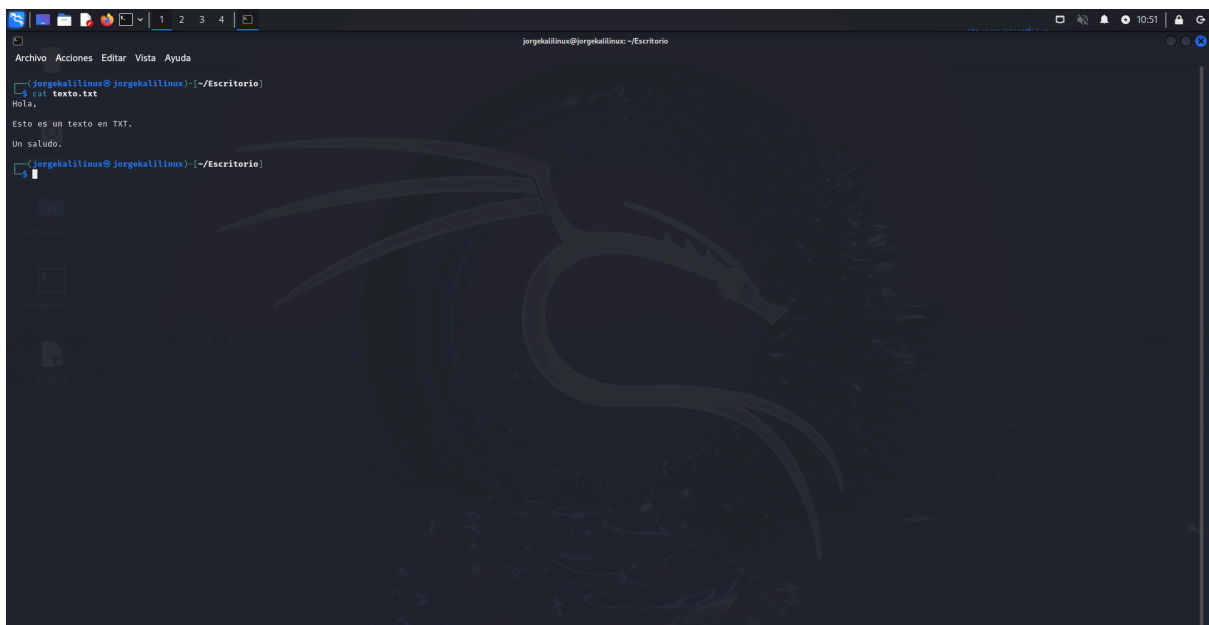
[1. Cifrado y Descifrado simétrico](#)

[2. Cifrado y Descifrado asimétrico](#)

[3. Función hash](#)

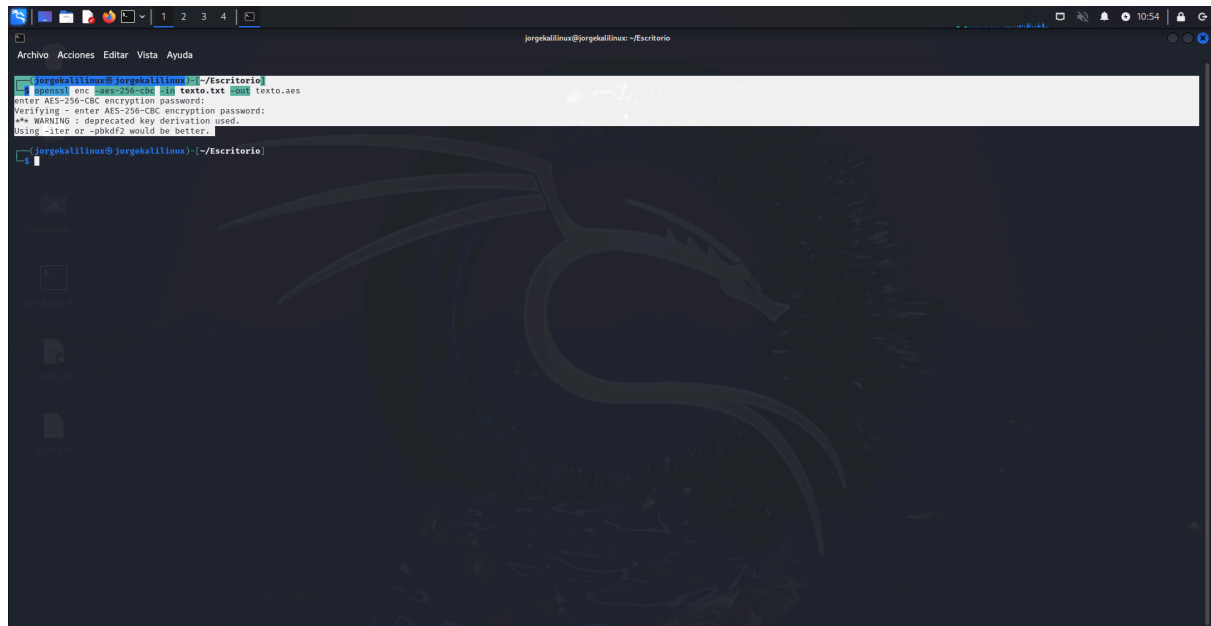
1. Cifrado y Descifrado simétrico

Vamos a crear un documento de texto plano:

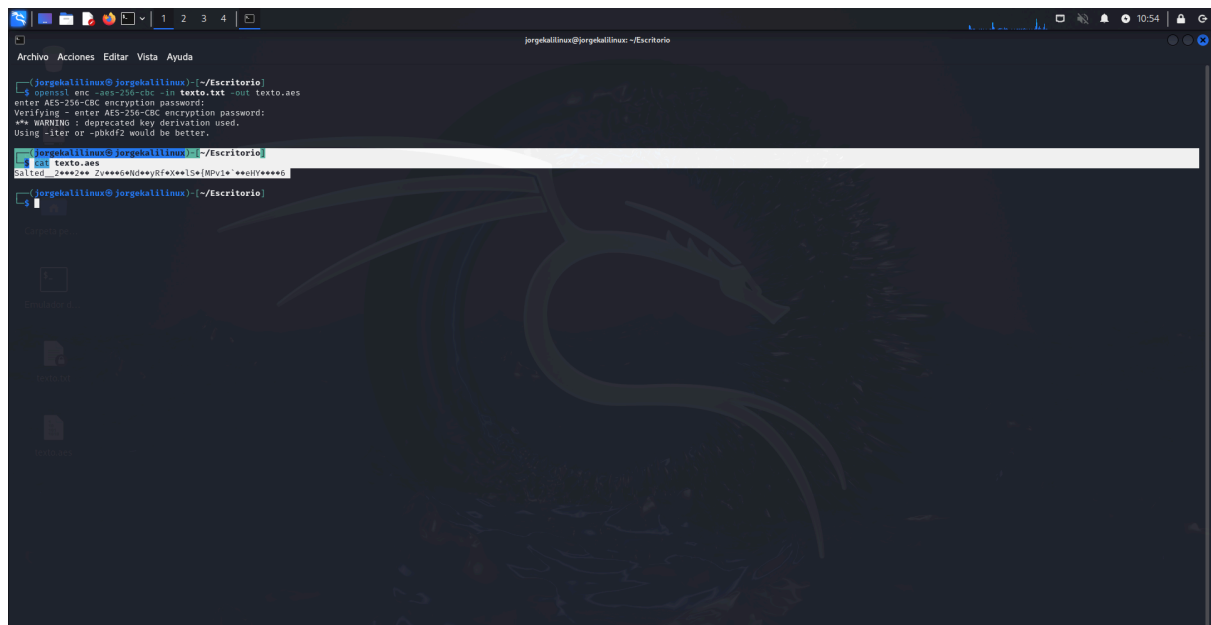
A screenshot of a Linux terminal window with a dark background and a large, faint dragon logo. The terminal shows the user 'jorgekalilinux' at the prompt 'jorgekalilinux@jorgekalilinux: ~/Escritorio'. The user enters the command 'cat > prueba.txt' and the terminal displays the file's contents: 'Hola,' and 'Esto es un texto en TXT.' followed by a blank line and 'Un saludo.'.

```
jorgekalilinux@jorgekalilinux: ~/Escritorio
jorgekalilinux@jorgekalilinux:~/Escritorio$ cat > prueba.txt
Hola,
Esto es un texto en TXT.
Un saludo.
jorgekalilinux@jorgekalilinux:~/Escritorio$
```

Vamos a cifrar el documento prueba.txt utilizando el cifrado aes-256-cbc con el siguiente comando:



Va a solicitar una contraseña, que hay que verificar:



A continuación, visualizamos el fichero encriptado (texto.aes). Vamos a descifrar el documento texto.aes con el siguiente comando. A continuación, visualizamos el fichero encriptado:

```
jorgekalilinux@jorgekalilinux: ~/Escritorio
[jorgekalilinux@jorgekalilinux]~/Escritorio
$ openssl enc -aes-256-cbc -in texto.txt -out texto.aes
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[jorgekalilinux@jorgekalilinux]~/Escritorio
$ cat texto.aes
Salted__***2***GEND**Rfex**ls*[MPv]a **0t*****6
[jorgekalilinux@jorgekalilinux]~/Escritorio
$ openssl enc -aes-256-cbc -d -in texto.aes -out texto.descifrado
enter AES-256-CBC decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[jorgekalilinux@jorgekalilinux]~/Escritorio
$ cat texto.descifrado
HOLA,
Esto es un texto en TXT.
Un saludo.
[jorgekalilinux@jorgekalilinux]~/Escritorio
$
```

Podemos comprobar que el fichero texto.descifrado es igual a texto.txt con el comando diff:

```
jorgekalilinux@jorgekalilinux: ~/Escritorio
[jorgekalilinux@jorgekalilinux]~/Escritorio
$ openssl enc -aes-256-cbc -in texto.txt -out texto.aes
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[jorgekalilinux@jorgekalilinux]~/Escritorio
$ cat texto.aes
Salted__***2***GEND**Rfex**ls*[MPv]a **0t*****6
[jorgekalilinux@jorgekalilinux]~/Escritorio
$ openssl enc -aes-256-cbc -d -in texto.aes -out texto.descifrado
enter AES-256-CBC decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[jorgekalilinux@jorgekalilinux]~/Escritorio
$ cat texto.descifrado
HOLA,
Esto es un texto en TXT.
Un saludo.
[jorgekalilinux@jorgekalilinux]~/Escritorio
$ diff texto.txt texto.descifrado
[jorgekalilinux@jorgekalilinux]~/Escritorio
$
```

2. Cifrado y Descifrado asimétrico

Vamos a crear un documento de texto plano:

```
jorgehallinux@jorgehallinux: ~/Escritorio
$ nano texto2.txt
jorgehallinux@jorgehallinux: ~/Escritorio
$ cat texto2.txt
Hola,
Este es el texto 2 del TXT.
Un saludo.
jorgehallinux@jorgehallinux: ~/Escritorio
$
```

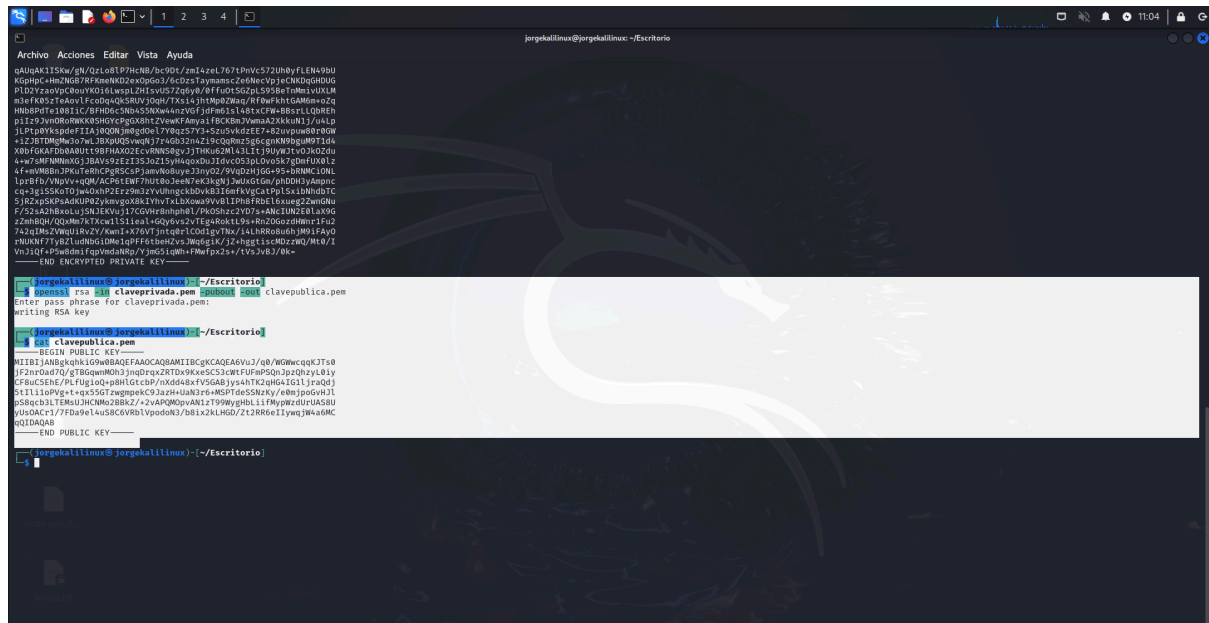
Vamos a generar la clave privada utilizando el algoritmo aes128. De esta forma, solicitará una contraseña para generar la clave, cada vez que se utilice. Para ello utilizamos el siguiente comando:

Va a solicitar una clave. Y ya tenemos la clave privada:

```
jorgehallinux@jorgehallinux: ~/Escritorio
$ cat texto2.txt
Hola,
Este es el texto 2 del TXT.
Un saludo.
jorgehallinux@jorgehallinux: ~/Escritorio
$ openssl genrsa -aes128 -out claveprivada.pem
Enter PEM pass phrase:
40C71ED3A37F0800:error:04000000:PEM routines:PEM_def_callback:problems getting password:../crypto/pem/lib.c:62:
40C71ED3A37F0800:error:07800009:common libcrypto routines:do_ui_passphrase:interrupted or cancelled:../crypto/passphrase.c:178:
40C71ED3A37F0800:error:1C80000F:Provider routines:pinfo_to_encpb:unable to get passphrase:../providers/implementations/encode_decode/encode_key2any.c:116:
jorgehallinux@jorgehallinux: ~/Escritorio
$ openssl genrsa -aes128 -out claveprivada.pem
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
jorgehallinux@jorgehallinux: ~/Escritorio
$ cat claveprivada.pem
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFHTBfBgkqhkiG9w0BBQwJAA8gqhkIG9wBBQwJAAQwBggL2LQ5631J51
a0Z2/ALCCAAwDAYIKoZInvcHAgAFAAABgLnghgQZQEAQIENRPTBf1K10x53
P2buTtEgTQ2Phd9pQ2zmeTDCouGhL/25c9vKZSA3YzqWmduZwY1J2Q
Kq+hCQu3v5ghagPDVPM0z0CT1B13W0wcv9M0A2Wj9N0N1Ct6gEXm5dH2
Wb/DeaL1N2BzFjHhGh/1+3WtTjWmSc6dWhRtECP2Jic2blqV8Ww5Rt
anYn/v99H8KfKdDtpQdMcE5fKtT6SWTgTnWpWVjoqrcvKvOf2k3UC1No3Wxc
WgWw-Z6W0D3PMA2A3Cc52Q8aR1J3J370D18KLqML3F0NMP4M576w7Ym1ocp
E5KTUE3Ja+T1uKx2yF5q/ZnaGf9FmYp1K1u8G6mGdGgF9vUu0K0Two
BgKf5m0Hd2Bplo/gYnSEJ2205Fk/LPw4vP1g1V5H5M151jhpJwJzUa+1H189K
6AQAk13Kw/gLQ2L8lP7HcN6/bc90f/zm14zL7677Hvvc57UhyfLE490u
6G6pC+hN26B78Fm6X0Zv0dG6J/6c3rTaymanc26dMcVpJcNDQ6H0UG
P1D2YzavP8ouYQ16LwspLZHisv572qy8/8Ffu05GzPl5958etNmavUXLM
4chF85T1AduV1CdqQKSRUJ0gk/7X151jHhG2Ww/PfWfKh1GdMw+Zg
WNB8Pdt10B1C/BF0Hc5Nb45N0W44nzVGFJ9Fm61548tCFW-B8srLLQREH
a1I29Jvnd0RWKX8SHOYcPg6XhTzVwKfAmya1f8C8BmJvmaA2KxkU1j/u4Lp
5LPt8YKsp8F1IA3R0Q6Jmg6d6L7Ybq57V3+Scu5vKtEE7+82vupw6B7R0W
+1Z8TDM0w3o7wL2BxPuQ5vWqN37r4G532n4Z19CqQmz56cgnK9Bgw9T1d4
X0bFGAFD0AABU1E9BFA02EcVNN58gVj7Thku2H143L1:39UW7v0J02du
4+7r8F8MNA0G1J8aV5tE13530Z15yHqouDuJ16v0S3pL0v87TgmUvU1Z
4F+mW6Bn3PKuterRCP85Csp3amK0u8ye23y0Z/79VQZ1H0G+95bRNMCI0NL
1p1r5b/7h0v+qM/ACPELEF7JUI8o+eN7KckgK1W0X0G0u/pH0D3Yamnc
4+5g15S6K0TJW40hP2Er2m22VUthgckdDvK316WfKvGcatPl5x10NhdTC
5J2z5p5KPAduP82ymvgoX8IYhVtLbX0wa9Vb1Pn8fRDE16ueng2Zw0Nu
F52a2NbaLuj5N1EKVU1J7CG0Hr8mHh8/PK0S1x22YD1+4Mc1N2L0LA9P6
2ZmhBQh/QQAme7K7xw151leal+qQy6v52Vt8qARokLT9sRn20G6ZdHm1Fu2
742Zm2Zw0u1RvZV/KanI+76VTJntq8rLC0e1gVhZ/14LhR08u8hJm1Fay0
hUNK777p2Lun8b0Lm61pFFr8t8mZ+3W6dK/1Z7Hgt1s+M0ZcWU/MW/1
Vh3IQfP5W8dM1f9vmdaR0P/7Jm6S1qHh+FMfpx2s+/V5Jv63/0k=
-----END ENCRYPTED PRIVATE KEY-----
jorgehallinux@jorgehallinux: ~/Escritorio
$
```

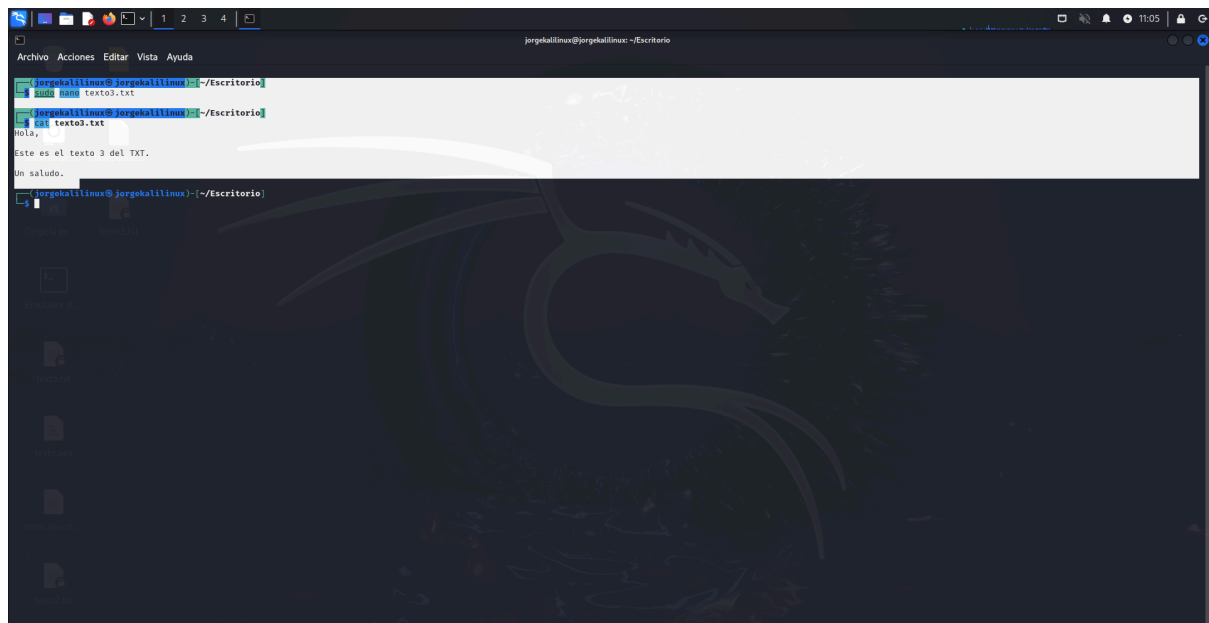
A continuación, generamos la clave pública. A la clave pública, no le vamos a solicitar contraseña. Para ello, utilizamos el comando:

Y ya tenemos la clave pública:



```
jorge@kali:~$ openssl genrsa -out claveprivada.pem 2048
jorge@kali:~$ openssl rsa -in claveprivada.pem -pubout -out clavepublica.pem
Enter pass phrase for claveprivada.pem:
writing RSA key
jorge@kali:~$ cat clavepublica.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6VUj/0b/W0wccqK7T40
F2n0u070/gT0w0m0h3j0p0p0x20T0v0eS03c0w0F0u0p0d0j0p0h0z0y0z0y0
CF8UCSEH/PcF0u0q0p0B0H0G0c0p0/xx0d0x0FV5GABjys4HTC2q0G4G01jra0d0
St0I0p0Vg+tcq05S0Tzw0p0K0C9Jaz0h0u0k0r0+NS0PT0e0S0R0z0y0/0e0j0p0d0h0J0
S0R0c0b0J0T0E0U0H0C0M0b0B0Z0/0v0A0Q0G0V0A0T0I0T090p0d0L0F0p0e0d0U0A0S0U
V0U0D0A0C0r0T0F0D0e0L0u0S0C0V0R0V0p0d0N0/0b0x0Z0K0H0D0/020R0E0I0y0w0J0W0A0M0C
0Q0I0A0Q0A0
-----END PUBLIC KEY-----
jorge@kali:~$
```

El texto debe ser menor que en el cifrado simétrico, ya que el cifrado con clave pública y privada no permite ficheros muy grandes:



```
jorge@kali:~$ cat texto3.txt
Hola,

Este es el texto 3 del TXT.

Un saludo.
jorge@kali:~$ openssl rsauts -in texto3.txt -pubin clavepublica.pem -out texto3.enc
jorge@kali:~$ openssl rsauts -in texto3.enc -privin claveprivada.pem -out texto3.dec
jorge@kali:~$ cat texto3.dec
Hola,

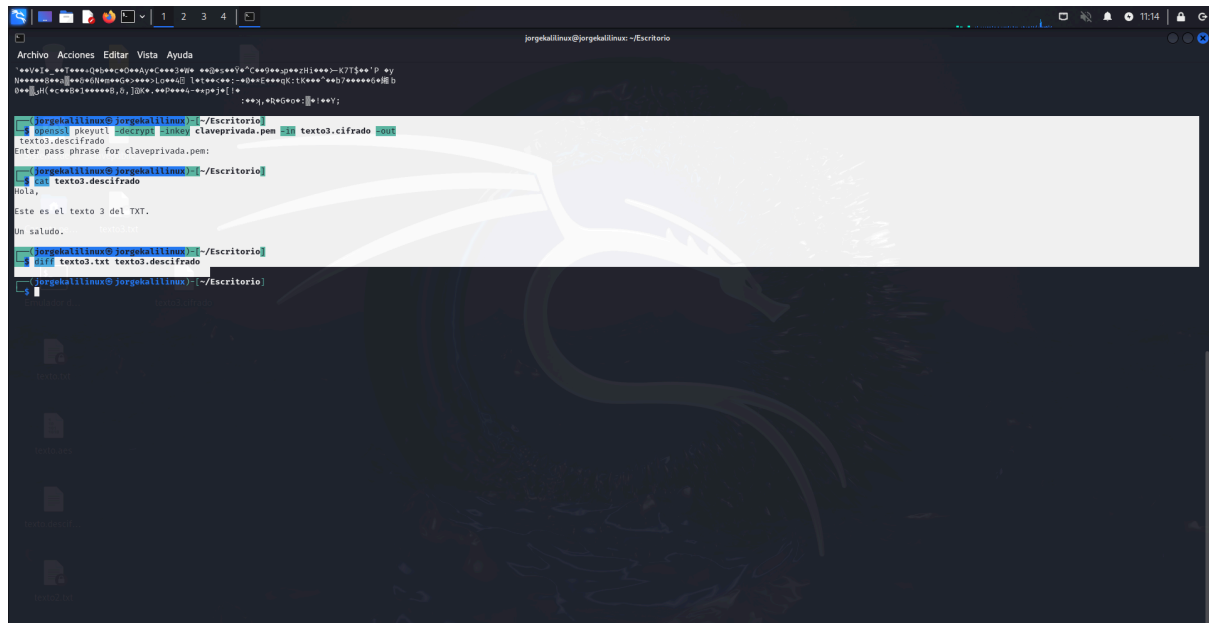
Este es el texto 3 del TXT.

Un saludo.
jorge@kali:~$
```

En primer lugar, vamos a cifrar el documento de texto plano con la clave pública:

Ahora, vamos a descifrar el documento de texto plano con la clave privada:

Nos va a solicitar la contraseña para utilizar la clave privada:



```
jorgehallinux@jorgehallinux: ~/Escritorio
Archivo Acciones Editar Vista Ayuda
jorgehallinux@jorgehallinux:~/Escritorio
$ cat texto3.cifrado
texto3.cifrado
$ cat claveprivada.pem
texto3.cifrado
$ openssl decrypt -in texto3.cifrado -inkey claveprivada.pem -out texto3.descifrado
Enter pass phrase for claveprivada.pem:
$ cat texto3.descifrado
Este es el texto 3 del TXT.
Un saludo.
$ diff texto3.txt texto3.descifrado
$
```

3. Función hash

Vamos a trabajar con los documentos de texto plano:

El primero es un documento pequeño, de varias líneas, y el segundo es un fichero grande que contiene miles de líneas:

Como se puede observar, el tamaño del hash es el mismo:

También, se puede generar una función HMAC (con contraseña):

El algoritmo de Hash que utiliza es sha-256. De esta manera, con una contraseña previamente convenida con el receptor se puede calcular el Hash. Esto añade la autenticación al algoritmo.

```
jorgekalilinux@jorgekalilinux: ~/Escritorio
└─$ openssl dgst -sha256 texto.txt
SHA2-256(texto.txt)= a2fb426cab3095c06cb7dcf2ece118058128541a3780877b645cd
c7bc9bc01

jorgekalilinux@jorgekalilinux: ~/Escritorio
└─$ openssl dgst -sha256 texto2.txt
SHA2-256(texto2.txt)= 0f3660bae039edee3173c61e4dcc2ad6992251827e0ef575c1806d
405f1c0fd2

jorgekalilinux@jorgekalilinux: ~/Escritorio
└─$ openssl dgst -sha256 texto3.txt
SHA2-256(texto3.txt)= 9273eb406fe160780e20e0ef9b0b78b2b757acc870eabf978ab2fe
cd759a6a92

jorgekalilinux@jorgekalilinux: ~/Escritorio
```

```
jorgekalilinux@jorgekalilinux: ~/Escritorio
└─$ openssl dgst -sha256 texto3.txt
SHA2-256(texto3.txt)= 9273eb406fe160780e20e0ef9b0b78b2b757acc870eabf978ab2fe
cd759a6a92

jorgekalilinux@jorgekalilinux: ~/Escritorio
└─$ openssl dgst -hmac "1eius-231Y50" texto.txt
HMAC-SHA256(texto.txt)= e1df0b631da0e3420450ef3fe17e6fcb702186b6a4d16a0007
0075482dda92

jorgekalilinux@jorgekalilinux: ~/Escritorio
└─$ openssl dgst -hmac "1eius-231Y50" texto2.txt
HMAC-SHA256(texto2.txt)= 209a4bab97b6680b9547531b089592970c0f89d613197d781
658b9fe66750e

jorgekalilinux@jorgekalilinux: ~/Escritorio
└─$ openssl dgst -hmac "1eius-231Y50" texto3.txt
HMAC-SHA256(texto3.txt)= faa54aa6e6b4201539bdabd5d66334320a948611b53316a300
a880080f6e025

jorgekalilinux@jorgekalilinux: ~/Escritorio
```