

UD1. Adopción de pautas de seguridad informática. Legislación

Tarea 6 – Cifrado asimétrico y firma digital

Para la tarea de cifrado asimétrico usaremos la misma herramienta que en la tarea anterior, el comando `gpg`

A continuación vamos a explicar cómo se utiliza para generar un par de claves de firma asimétrica y, seguidamente, se planteará una actividad dividida en dos partes.

1. En primer lugar, para generar el par de claves se utiliza `gpg --gen-key`

Nos solicitará una serie de datos. Con esos datos se identificarán las claves, por lo que conviene poner datos reales.

```
administrador@administrador-VirtualBox:~$ gpg --gen-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Nota: Usa "`gpg --full-generate-key`" para el diálogo completo de generación de clave.

GnuPG debe construir un ID de usuario para identificar su clave.

Nombre y apellidos: █

También nos pedirá una contraseña con la que proteger el llavero donde se guardarán las claves generadas.

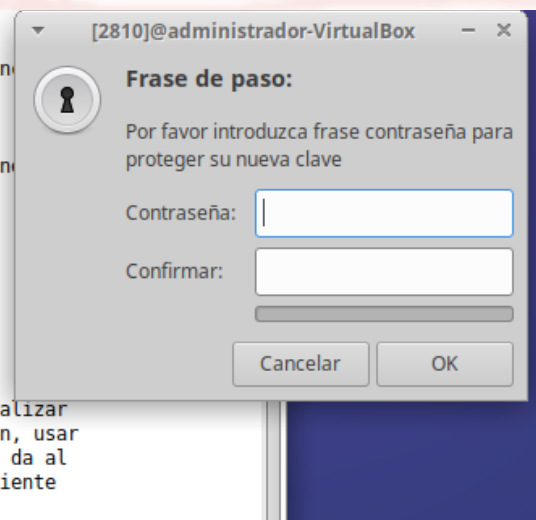
```
administrador@administrador-VirtualBox:~$ gpg --gen-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Nota: Usa "`gpg --full-generate-key`" para el diálogo completo de generación de clave.

GnuPG debe construir un ID de usuario para identificar su clave.

```
Nombre y apellidos: asdas
Dirección de correo electrónico: asd@asd.asd
Ha seleccionado este ID de usuario:
"asdas <asd@asd.asd>"
```

```
¿Cambia (N)ombre, (D)irección o (V)ale/(S)alir? V
Es necesario generar muchos bytes aleatorios. Es una buena idea realizar
alguna otra tarea (trabajar en otra ventana/consola, mover el ratón, usar
la red y los discos) durante la generación de números primos. Esto da al
generador de números aleatorios mayor oportunidad de recoger suficiente
entropía.
```



The image shows a terminal window titled "[2810]@administrador-VirtualBox" with a key icon. A password dialog box is overlaid on the terminal. The dialog box has a title bar with a key icon and the text "Frase de paso:". Below the title bar, it says "Por favor introduzca frase contraseña para proteger su nueva clave". There are two input fields: "Contraseña:" and "Confirmar:". At the bottom, there are two buttons: "Cancelar" and "OK".

Tardará un rato en generar las claves. Durante ese tiempo podemos ejecutar otros programas o mover el ratón para que el generador de bytes aleatorios tenga más datos con los que trabajar y generar así una clave más segura.

Cuando finalice nos mostrará las claves creadas, así como el uid, es decir, el nombre que le hemos dado.

Es necesario generar muchos bytes aleatorios. Es una buena idea realizar alguna otra tarea (trabajar en otra ventana/consola, mover el ratón, usar la red y los discos) durante la generación de números primos. Esto da al generador de números aleatorios mayor oportunidad de recoger suficiente entropía.

```
gpg: clave 8DDFE776A4EE458F marcada como de confianza absoluta
gpg: creado el directorio '/home/administrador/.gnupg/openpgp-revocs.d'
gpg: certificado de revocación guardado como '/home/administrador/.gnupg/openpgp-revocs.d/4529B17A3CD960469744CCC08DDFE776A4EE458F.rev'
claves pública y secreta creadas y firmadas.
```

```
pub  rsa3072 2023-10-01 [SC] [caduca: 2025-09-30]
      4529B17A3CD960469744CCC08DDFE776A4EE458F
uid                               Javi Daroqui <fj.daroquimarti@edu.gva.es>
sub  rsa3072 2023-10-01 [E] [caduca: 2025-09-30]
```

En nuestro directorio HOME se habrá creado un directorio oculto .gnupg donde se guardan los ficheros internos que utiliza la herramienta.

-rw-r--r--	1	administrador	administrador	26	dic	31	2022	.dmrc
drwxr-xr-x	5	administrador	administrador	4096	sep	12	19:57	Documentos
drwxr-xr-x	2	administrador	administrador	4096	dic	31	2022	Escritorio
-rw-rw-r--	1	administrador	administrador	58	feb	26	2023	.gitconfig
drwxrwxr-x	7	administrador	administrador	4096	ene	1	2023	GNS3
drwx-----	4	administrador	administrador	4096	oct	1	13:05	.gnupg
-rw-----	1	administrador	administrador	0	dic	31	2022	.ICEauthority
drwxr-xr-x	2	administrador	administrador	4096	dic	31	2022	Imágenes
drwxrwxr-x	3	administrador	administrador	4096	feb	15	2023	.ipython
drwxrwxr-x	3	administrador	administrador	4096	sep	26	17:51	.java
drwxrwxr-x	2	administrador	administrador	4096	jul	4	18:40	.jupyter
-rw-----	1	administrador	administrador	20	sep	17	20:06	.lessht
drwxrwxr-x	6	administrador	administrador	4096	feb	15	2023	.local
drwxrwxr-x	3	administrador	administrador	4096	abr	28	19:08	'Local Sites'
drwx-----	3	administrador	administrador	4096	abr	2	10:04	.mozilla
drwxr-xr-x	2	administrador	administrador	4096	dic	31	2022	Música
drwx-----	3	administrador	administrador	4096	feb	26	2023	.pki
drwxr-xr-x	2	administrador	administrador	4096	dic	31	2022	Plantillas
-rw-r--r--	1	administrador	administrador	807	dic	31	2022	.profile
drwxr-xr-x	2	administrador	administrador	4096	dic	31	2022	Público
drwx-----	3	administrador	administrador	4096	ene	8	2023	snap
drwx-----	2	administrador	administrador	4096	abr	23	16:59	.ssh

2. Podemos listar las claves con el comando `gpg --list-keys`

```
administrador@administrador-VirtualBox:~$ gpg --list-keys  
/home/administrador/.gnupg/pubring.kbx  
-----  
pub   rsa3072 2023-10-01 [SC] [caduca: 2025-09-30]  
      4529B17A3CD960469744CCC08DDFE776A4EE458F  
uid    [ absoluta ] Javi Daroqui <fj.daroquimarti@edu.gva.es>  
sub    rsa3072 2023-10-01 [E] [caduca: 2025-09-30]
```

3. Para poder enviar nuestra clave privada a alguien con quien queramos comunicarnos, para que nos envíe mensajes cifrados con ella, primero tenemos que sacarla del llavero con el parámetro `export`:

`gpg -a --export -o ruta/nombre_fichero.pub usuario` donde hemos usado `-a` para que sea en formato texto y `ruta/nombre_fichero.pub` será el fichero en el que guardaremos la clave y `usuario` será el uid que introdujimos al crear la clave.

Ese fichero ya lo podremos enviar a la persona con quien queramos comunicarnos.

```
administrador@administrador-VirtualBox:~$ gpg -a --export -o ClavePublicaJD.pub "Javi Daroqui"  
administrador@administrador-VirtualBox:~$ ls -l  
total 884  
-rw-rw-r-- 1 administrador administrador 59 feb 28 2023 c7200_i0 log.txt  
-rw-rw-r-- 1 administrador administrador 2464 oct 1 13:08 ClavePublicaJD.pub  
drwxr-xr-x 4 administrador administrador 4096 sep 26 18:20 Descargas  
drwxr-xr-x 5 administrador administrador 4096 sep 12 19:57 Documentos  
drwxr-xr-x 2 administrador administrador 4096 dic 31 2022 Escritorio  
drwxrwxr-x 7 administrador administrador 4096 ene 1 2023 GNS3  
drwxr-xr-x 2 administrador administrador 4096 dic 31 2022 Imágenes  
drwxrwxr-x 3 administrador administrador 4096 abr 28 19:08 'Local Sites'  
drwxr-xr-x 2 administrador administrador 4096 dic 31 2022 Música
```

4. Esa persona deberá importar nuestra clave pública para poder utilizarla. Se utiliza el comando:

`gpg --import nombre_fichero`

```
administrador@administrador-VirtualBox:~$ gpg --import ClaveJuanPerez.pub  
gpg: clave ECF8E1001F0C2022: clave pública "Juan Pérez <jperez@notengocorreos.es>" importada  
gpg: Cantidad total procesada: 1  
gpg:          importadas: 1
```

Si ese usuario vuelve a ejecutar `gpg --list-keys` verá que, junto con sus propias claves, ha aparecido una nueva del usuario alumno.


```
administrador@administrador-VirtualBox:~$ gpg --list-keys  
/home/administrador/.gnupg/pubring.kbx  
-----  
pub   rsa3072 2023-10-01 [SC] [caduca: 2025-09-30]  
      4529B17A3CD960469744CCC08DDFE776A4EE458F  
uid           [ absoluta ] Javi Daroqui <fj.daroquimarti@edu.gva.es>  
sub   rsa3072 2023-10-01 [E] [caduca: 2025-09-30]  
  
pub   rsa3072 2023-10-01 [SC] [caduca: 2025-09-30]  
      6A4832CF7A398BC0E3D956A7ECF8E1001F0C2022  
uid           [desconocida] Juan Pérez <jperez@notengocorreo.es>  
sub   rsa3072 2023-10-01 [E] [caduca: 2025-09-30]
```

5. Creamos un fichero con un mensaje.

Mensaje de prueba

~
~

Para encriptar el fichero con la clave pública recién importada utilizaremos el siguiente comando:

`gpg -v -a -o ruta/mensaje.cifrado --encrypt --recipient usuario fichero` donde:

- `-v` es modo verbose, o sea, más información visual
- `-a` y `-o` ya los conocemos
- `--encrypt` indica que estamos cifrando
- `--recipient` indica la clave del usuario con la que vamos a cifrar
- Por último, mensaje es el fichero a cifrar

Al intentar cifrar el sistema nos indicará que no hay seguridad de que esa clave pertenezca realmente al usuario que especificamos. Cualquiera podría haber copiado el fichero `alumno.pub` antes de que hiciésemos el `import`. El comando nos ofrece la huella como ayuda y nos pide que confirmemos si queremos continuar.

```
administrador@administrador-VirtualBox:~$ gpg -v -a -o mensaje_cifrado --encrypt --recipient "Juan Pérez" mensaje.txt  
gpg: usando gpg como modelo de confianza  
gpg: usando subclave 973B296F66CF78E6 en vez de clave primaria ECF8E1001F0C2022  
gpg: 973B296F66CF78E6: No hay seguridad de que esta clave pertenezca realmente  
al usuario que se nombra  
  
sub   rsa3072/973B296F66CF78E6 2023-10-01 Juan Pérez <jperez@notengocorreo.es>  
Huella clave primaria: 6A48 32CF 7A39 8BC0 E3D9 56A7 ECF8 E100 1F0C 2022  
Huella de subclave: EFB4 9736 7CC2 203A F0DF AA57 973B 296F 66CF 78E6  
  
No es seguro que la clave pertenezca a la persona que se nombra en el  
identificador de usuario. Si *realmente* sabe lo que está haciendo,  
puede contestar sí a la siguiente pregunta.  
  
¿Usar esta clave de todas formas? (s/N)
```

Para ver la huella de nuestra clave pública podemos usar el comando `gpg --fingerprint` con lo que podríamos pedirle al destinatario que nos envíe la clave pública que nos enviase también su huella y comprobar que coincide con la que nos da el sistema.

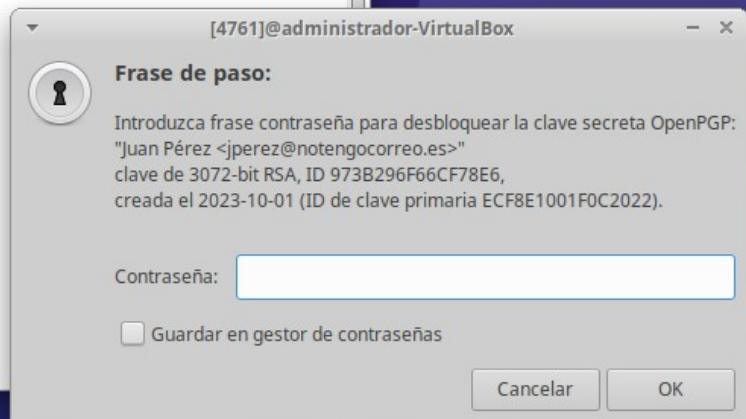
```
administrador@administrador-VirtualBox:~$ gpg --fingerprint
/home/administrador/.gnupg/pubring.kbx
-----
pub  rsa3072 2023-10-01 [SC] [caduca: 2025-09-30]
    4529 B17A 3CD9 6046 9744 CCC0 8DDF E776 A4EE 458F
uid  [ absoluta ] Javi Daroqui <fj.daroquimarti@edu.gva.es>
sub  rsa3072 2023-10-01 [E] [caduca: 2025-09-30]

pub  rsa3072 2023-10-01 [SC] [caduca: 2025-09-30]
    6A48 32CF 7A39 8BC0 E3D9 56A7 ECF8 E100 1F0C 2022
uid  [desconocida] Juan Pérez <jperez@notengocorreo.es>
sub  rsa3072 2023-10-01 [E] [caduca: 2025-09-30]
```

6. Finalmente, cuando recibimos el mensaje cifrado con nuestra clave pública, podemos descifrarlo con el comando `gpg --decrypt mensaje_cifrado`

El sistema nos solicitará la contraseña de nuestro llavero y mostrará el mensaje por pantalla. También podemos usar `-o` para llevarlo a un fichero.

```
user1@administrador-VirtualBox:~$ gpg --decrypt mensaje_cifrado
```

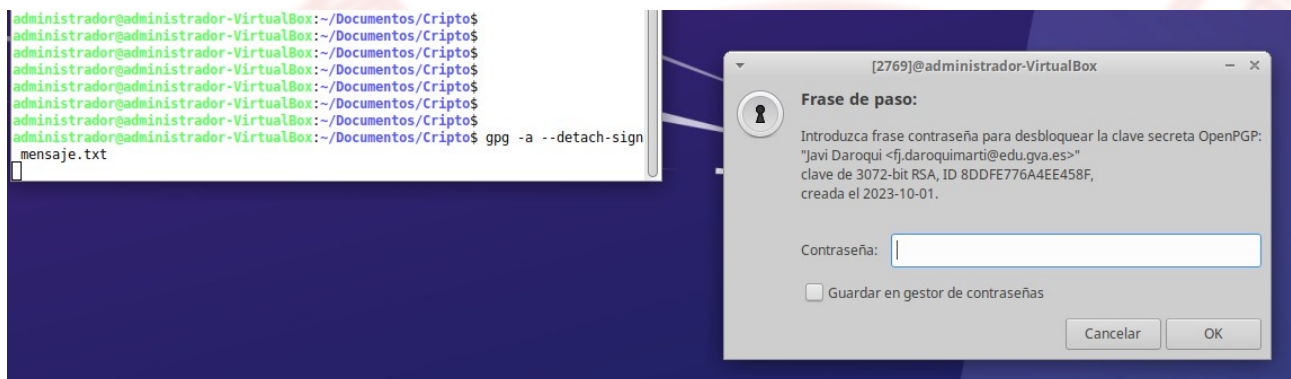


```
user1@administrador-VirtualBox:~$ gpg --decrypt mensaje_cifrado
gpg: cifrado con clave de 3072 bits RSA, ID 973B296F66CF78E6, creada el 2023-10-01
    "Juan Pérez <jperez@notengocorreo.es>"
Mensaje de prueba
user1@administrador-VirtualBox:~$ █
```

7. Una vez visto esto podemos entender cómo funciona la firma digital. Vamos a seguir trabajando con el mensaje original. En primer lugar tenemos que crear la huella del mensaje, para lo que utilizaremos el comando:

```
gpg -a --detach-sign nombre_fichero
```

Este comando nos generará, a partir de nuestra clave privada, un nuevo fichero llamado `nombre_fichero.asc` que contendrá el cifrado del hash del fichero original, es decir, la firma. Si enviamos el fichero junto con la firma a un destinatario que tenga nuestra clave pública, podrá comprobar que el fichero es nuestro y no se ha modificado.



El contenido del fichero será algo similar a esto:

```
administrador@administrador-VirtualBox:~/Documentos/Cripto$ cat mensaje.txt.asc
-----BEGIN PGP SIGNATURE-----

iQGzBAABCGAdFiEERSmxejzZYEAxRMzAjd/ndqTuRY8FAMUZhZUACgkQjd/ndqTu
RY+wMAwAiW7ZvgfEHSF5KgOPWSntS9lgczTCMPCBT/v3qFrjt3qlj6FPVxlu67V7
CXWxEuUXrWaf8ToFSTN8TT//KRFHYAiGjbnV2lUrpAQb71cDSHfpvnAuA4QoIRUe
7sGkZ0Yte6IXkPJz6Uto4JnbIRJoa0f/MW75ACxEWHGnLHfAAj7Q+qCJC40/TWRT
Be6LqDjuQ6Cl3UWxyTmYbEaVbToj6hbEVLznZTdYdFgNqObAz04E09V0rp++i8hz
aJIJCo4Lt2p9kl8Lvlc/se0VcU6mzxMHZlMkwed0Ndgt0kqKG7eIZPmzD7NvPNNw
JxGBT1c3+zYTs37D/mCW0lgPrWRorxzVvtQIop2IrT7n40/NuZvporT+CoFRZCN+
dAK532WJstAeG0sZqZNl+w5nx33g8fvl0UrpnpNtY6j9DUH9sd8d1MhAZi3fwhQ6
UtTYTFbps89PYgcvHxS0yRPnxhKYx7s69RzTYDfUkLU/Px12wZ2YAunUVqk1nWB
b3w6wy14
=Mdn+
-----END PGP SIGNATURE-----
```

- Cuando recibimos un fichero con su firma (como en el caso anterior) debemos usar el siguiente comando para comprobar esa firma:

```
gpg --verify nombre_fichero.asc
```

teniendo en cuenta que tanto el fichero original como el fichero con la firma han de estar en el mismo directorio.

Ese comando nos dirá si realmente la firma corresponde al fichero original, con lo que podremos garantizar que el fichero es de quien dice ser y no se ha modificado. De hecho, podemos probar a modificar el fichero original y veremos como al ejecutar el comando anterior la verificación fallará.

8. Si en lugar de enviar dos ficheros, queremos enviar tanto el mensaje como la firma en el mismo fichero, podemos utilizar el comando siguiente:


```
gpg -a --clearsign nombre_fichero
```

En este caso tendremos tanto la firma como el contenido del fichero original dentro del fichero `nombre_fichero.asc`. El contenido del fichero será similar a esto:

```
administrador@administrador-VirtualBox:~/Documentos/Cripto$ cat mensaje.txt.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA12
```

Mensaje de prueba

mensaje

```
-----BEGIN PGP SIGNATURE-----
```

```
iQGzBAEBCgAdFiEERSmxejjZYEAxRMzAjd/ndqTuRY8FamUZiBgACgkQjd/ndqTu
RY8+uuv+NxRkg22VFpdr5qMpGXq94LfgytEqTK2TBQo0LpCRJGqAxSj3tBAC/0jS
ozaV6VNIbD0PxQkpygqii35alcpmXD+kVPbLAAsvALz+bGuDWONX0u90khqeYwe2
mVaXnd2WsSwoy4s+OnPLl4VGXWpDzv3ahZ2WYDXD0ZCeD7k0TCVzoqQK+cdMLVB
AGLTNGaFiOugkQ9VK+dQCv48nD0JMZ1GXuyCcrwTDNhqh504SEy3GD0IuQpyWdMJ
PhpCKxsqzIxpL7XHS6Ay9AvQDuUQVXWipwb0K2b5hAxhgtYaniurhekD02xzwFIQ
g7GY9dmCrMl0ayATKo4BqNTCLrVVC0Fokxj08gh8jexWgbzvfK0dIwMYaQdAuVl9
ACnoZMnI2vSNlTyFWRx9h+ygTvr3iNjQqigqxl5/DLRmB3qbZPHVdOHRF/EDkRll
S+73a/DGZTGLN40g7yk0vfUsU/nwKpd1XL5SxwtwxgmFVvS2gwQJdB0KJvghF06g
RhAgNceP
=a1SW
-----END PGP SIGNATURE-----
```

firma

También podemos comprobar que el fichero es de quién dice ser utilizando el comando del punto anterior.

9. El problema de esto es que el fichero lo hemos enviado sin cifrar, es decir, cualquiera podría haberlo interceptado y haber visto el contenido. Para enviar un único fichero con el contenido y la firma, pero en formato encriptado, usaremos el siguiente comando:

```
gpg -a --sign nombre_fichero
```

Esta vez tendremos un fichero llamado `nombre_fichero.asc` que contendrá tanto el contenido del fichero original como la firma y, además, estará cifrado con nuestra clave privada. Ya podemos enviarlo de forma segura al destinatario que queramos.

```
administrador@administrador-VirtualBox:~/Documentos/Cripto$ gpg -a --sign mensaje.txt
administrador@administrador-VirtualBox:~/Documentos/Cripto$ cat mensaje.txt.asc
-----BEGIN PGP MESSAGE-----
```

```
owEB6gEV/pANAwAKAY3f53ak7kWPAawjYgttZW5zYWplLnR4dGUZieBNZW5zYWpl
IGRlIHBydWVlYQqJABMEAAEKAB0WIQRfKbF6PNlgRpdEzMCN3+d2p05FjwUCZRMj
4AAKCRcN3+d2p05Fj8bFDAC/WHIhBsWzXnweIEb5UbyS2LlfGfknQ5/5psro3cbN
/lfhIqCISLsVdprM9U65jLZ/kmbU438FbJfXXYXEEAHWHSpbnlm6UND36Kbkfwc+
xWkDdEkwcB2bCxlubq9GnHqMqrGbmFXytCfpYUFvGZvydzf7HV2/swbsXB0Rss29
0Fi1Zq0rcViC5uIVlcvqU2hSZZtKUTg7XlU1ARBDxSv1b0Uo0I5RDKtfxFub+H5
gmf9kTXr6UpmwUKE2TqEVIqHkUnbLkKQ4K6ypBnKVRbY5KQWF2IjSKsj1J/VWY7M
HeKgzA15XookewSsuCHRhh3XPtRQA2bUQeR4XqVl0eslhIHM41IwzUMtQAOKPye+
A4XrDRNmI6G+Ptn3BcLR11FowThw3f+L361SIZbFieQH/cXA1AXCkhME0d5I58B
2QaLQqFi+KUqtPDOXQdy0B8P3U7poraz9/eivhN9PQ9rqZgMXQJfjvcjmt0EGjw
t87u3ZiJ0n5PQys8xLPbFUK=
=Feo6
-----END PGP MESSAGE-----
```

10. Ahora vemos otro punto débil. Cualquiera que intercepte el mensaje y tenga nuestra clave pública, podrá descifrarlo. ¿Cómo lo solucionamos? La solución es encriptar con la clave pública del destinatario y después, firmar con nuestra clave privada. Así, cualquiera que intercepte el mensaje y tenga nuestra clave pública podrá comprobar que el mensaje que ha interceptado es nuestro, pero no podrá acceder al contenido ya que sólo el destinatario podrá descifrarlo con su clave privada.

Sin embargo, el destinatario podrá tanto comprobar que el mensaje es nuestro como acceder al contenido de ese mensaje. Para poder utilizar correctamente todos estos pasos, debemos firmar las claves públicas que tengamos de los destinatarios, para que así, siempre que las utilicemos, el sistema sepa con seguridad que son correctas.

Para firmar la clave pública de un destinatario, una vez importada como vimos en la práctica anterior, utilizaremos el comando:

```
gpg --sign-key destinatario
```

Una vez visto esto, podemos comenzar con la tarea.

La tarea se realizará por parejas y consistirá en dos partes:

Primera parte:

- Debes generar tu par de claves pública y privada
- Sube a Aules tu clave pública para que el profesor pueda encriptar con ella un fichero con un mensaje
- Cuando el profesor suba el fichero, descárgalo y descifrálo con tu clave privada
- Escribe en Aules el mensaje que te ha mandado el profesor

Segunda parte:

- Descarga de Aules la clave pública del profesor y el fichero de texto con su huella
- Encripta con esa clave un fichero de texto con algún mensaje
- Comprueba antes de proceder al cifrado que la huella coincide
- Sube a Aules el fichero encriptado. El profesor debería poder descifrarlo con su clave privada.

Tercera parte:

En primer lugar, crea un par de claves para tu usuario y sube la clave pública a Aules.

- Descarga el fichero `mensaje.txt.asc` de Aules. Descarga además los ficheros `mensaje1.txt`, `mensaje2.txt` y `mensaje3.txt`. Deberás comprobar cuál de los mensajes no ha sido modificado tras firmarlo. Escribe en un fichero llamado `Ejercicio1.txt` la frase que hay en el fichero correcto. Crea otro fichero con la firma del mensaje y sube los dos ficheros a Aules

- Descarga los ficheros `mensaje1.txt.asc`, `mensaje2.txt.asc` y `mensaje3.txt.asc`. Estos ficheros contienen tanto el mensaje como la firma. Debes hacer lo mismo que en el primer ejercicio, es decir, averiguar cuál de ellos no se ha modificado tras la firma. Escribe en un fichero llamado `Ejercicio2.txt` la frase que hay en el fichero correcto. Firma el fichero de manera que se genere otro fichero con el mensaje y la firma, ambos en el mismo fichero. Sube ese fichero a Aules.
- Descarga de Aules el fichero con tus iniciales y extensión `.txt.asc`. Se trata de un fichero firmado por el profesor y cifrado con tu clave pública. Debes realizar los siguientes pasos:
 - Firma la clave pública del profesor
 - Descifra el fichero
 - Comprueba que el mensaje no se ha modificado por el camino, es decir, que la firma del profesor es válida
 - Escribe lo siguiente en un fichero de texto con el nombre `Ejercicio3.txt`:
 1. Los pasos utilizados en los puntos 1, 2 y 3
 2. Un mensaje que quieras enviar al profesor
 - Firma el fichero que has creado de manera que se cree un único fichero con el mensaje y la firma y que, además, ese fichero esté cifrado con tu clave privada. Cifra ese fichero con la clave pública del profesor y súbelo a Aules