

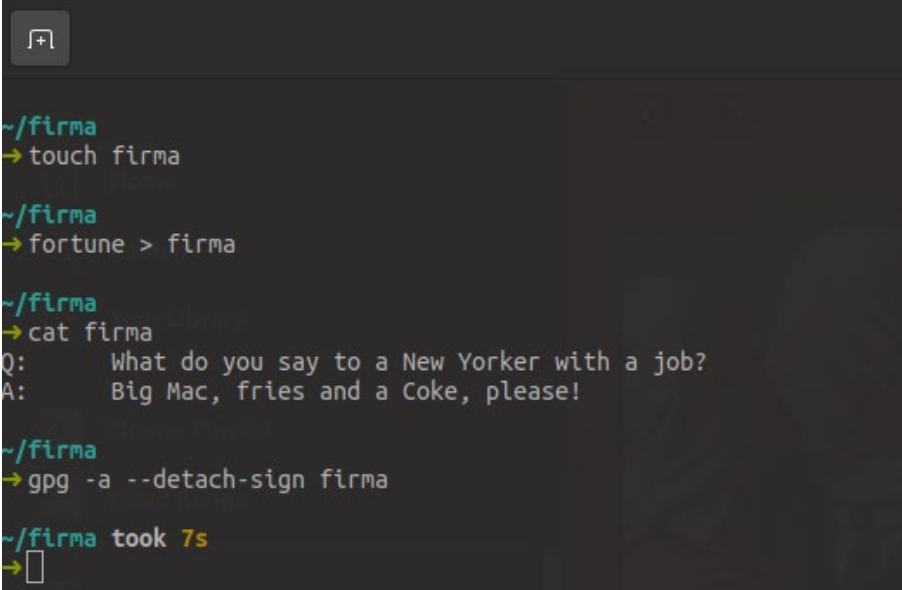
# FIRMA DIGITAL LINUX



Juan Carlos Navidad García  
Seguridad Informática

## 1. Firma digital en Linux:

- Recordemos que habíamos generado dos parejas de claves, una pareja **DSA** y una pareja **Elgamal**. En el cifrado del fichero habíamos utilizado la clave **Elgamal**.
- En la sesión de alumno creamos un fichero mensaje y lo firmamos con nuestra clave privada para que cualquiera pueda confirmar que es nuestro. Usaremos el parámetro **detach-sign**, que crea un fichero nuevo solo con la firma (el cifrado del resultado de aplicar el hash al fichero original). Utilizaremos también el parámetro **a** para observar ese fichero. Los comandos son:
  - **alumno\$ fortune > firma**
  - **alumno\$ gpg -a --detach-sign firma**



```
~/firma
→ touch firma

~/firma
→ fortune > firma

~/firma
→ cat firma
Q:    What do you say to a New Yorker with a job?
A:    Big Mac, fries and a Coke, please!

~/firma
→ gpg -a --detach-sign firma

~/firma took 7s
→
```

- Vemos que la herramienta nos pide la contraseña de nuestra clave secreta:



- Ahora realizamos el envío al usuario profesor (por ejemplo, podríamos estar entregando un trabajo). En este caso simplemente lo copiaremos en **/tmp**. Hay que copiar los dos ficheros: **firma**, que no lleva la firma, y **firma.asc**, que solo es una firma. Los comandos serían:
  - **alumno\$ cp firma /tmp**
  - **alumno\$ cp firma.asc /tmp**

```
~/firma
→ cp firma /tmp

~/firma
→ cp firma.asc /tmp

~/firma
→
```

- Iniciamos una sesión con el usuario **profesor**, copiamos los ficheros a nuestro directorio y comprobamos la firma. El parámetro es **verify** junto con el nombre del fichero que lleva la firma (en mi caso, **firma.asc**). Los comandos son:
  - **profesor\$ cp /tmp/firma\* .**
  - **profesor\$ gpg --verify firma.asc**

```
profesor@Juanca-PC:~$ cp /tmp/firma* .
profesor@Juanca-PC:~$ ls
firma  firma.asc
profesor@Juanca-PC:~$ gpg --verify firma.asc
gpg: asumiendo que los datos firmados están en 'firma'
gpg: Firmado el jue 28 oct 2021 20:32:37 CEST
gpg:          usando DSA clave DFF8A10C8CEAB3AD6533575B5CCAE263D2B597FA
gpg: Firma correcta de "alumno <alumno@gmail.com>" [desconocido]
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de confianza!
gpg:          No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: DFF8 A10C 8CEA B3AD 6533  575B 5CCA E263 D2B5 97FA
profesor@Juanca-PC:~$
```

- Para evitar enviar dos ficheros, podemos incluirlo todo en el mismo. Volvemos a la sesión del usuario **alumno** y ahora firmamos con el parámetro **clearsign**. El comando sería:
  - **alumno\$ gpg -a --clearsign firma**
- De nuevo hay un fichero **firma2.asc** pero ahora contiene tanto la firma como el texto del mensaje

```
~/firma
→ touch firma2

~/firma
→ fortune > firma2

~/firma
→ cat firma2
Q:    What lies on the bottom of the ocean and twitches?
A:    A nervous wreck.

~/firma
→ gpg -a --clearsign firma2

~/firma
→ ls
firma  firma2  firma2.asc  firma.asc

~/firma
→
```

- Podemos copiarlo de nuevo a **/tmp** y tomarlo desde la sesión del **profesor**. El comando de verificación es el mismo que en el caso del fichero separado.

```
~/firma
→ cp firma2.asc /tmp
```

```
profesor@Juanca-PC:~$ cp /tmp/firma2.asc* .
profesor@Juanca-PC:~$ ls
firma firma2.asc firma.asc
profesor@Juanca-PC:~$ gpg --verify firma2.asc
gpg: Firmado el jue 28 oct 2021 20:38:35 CEST
gpg: usando DSA clave DFF8A10C8CEAB3AD6533575B5CCAE263D2B597FA
gpg: Firma correcta de "alumno <alumno@gmail.com>" [desconocido]
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de confianza!
gpg: No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: DFF8 A10C 8CEA B3AD 6533 575B 5CCA E263 D2B5 97FA
profesor@Juanca-PC:~$
```

- De nuevo podemos llevarlo al directorio **/tmp** para que el usuario **profesor** haga una copia y lo verifique. Si además queremos extraer el fichero utilizaremos **decrypt**:
  - **profesor\$ cp /tmp/firma3.asc .**
  - **profesor\$ gpg --verify firma3.asc**
  - **profesor\$ gpg --decrypt -o firma3 firma3.asc**

```
profesor@Juanca-PC:~$ cp /tmp/firma3* .
profesor@Juanca-PC:~$ ls
firma firma2.asc firma3.asc firma.asc
profesor@Juanca-PC:~$ gpg --verify firma3.asc
gpg: Firmado el jue 28 oct 2021 20:43:42 CEST
gpg: usando DSA clave DFF8A10C8CEAB3AD6533575B5CCAE263D2B597FA
gpg: Firma correcta de "alumno <alumno@gmail.com>" [desconocido]
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de confianza!
gpg: No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: DFF8 A10C 8CEA B3AD 6533 575B 5CCA E263 D2B5 97FA
profesor@Juanca-PC:~$ gpg --decrypt -o firma3 firma3.asc
gpg: Firmado el jue 28 oct 2021 20:43:42 CEST
gpg: usando DSA clave DFF8A10C8CEAB3AD6533575B5CCAE263D2B597FA
gpg: Firma correcta de "alumno <alumno@gmail.com>" [desconocido]
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de confianza!
gpg: No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: DFF8 A10C 8CEA B3AD 6533 575B 5CCA E263 D2B5 97FA
profesor@Juanca-PC:~$ cat firma3
Q: What's a WASP's idea of open-mindedness?
A: Dating a Canadian.
profesor@Juanca-PC:~$
```

- Aunque esta protección es bastante débil, porque cualquiera que tenga la clave pública de alumno tendrá acceso al contenido del fichero. Debemos utilizar el cifrado normal, lo que nos lleva a que el usuario profesor genere su par de claves (hasta ahora no ha sido necesario).
  - **profesor\$ gpg --full-generate-key**

```
profesor@Juanca-PC:~$ sudo gpg --full-generate-key
[sudo] contraseña para profesor:
gpg (GnuPG) 2.2.19; Copyright (C) 2019 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.

gpg: creado el directorio '/root/.gnupg'
gpg: caja de claves '/root/.gnupg/pubring.kbx' creada
Por favor seleccione tipo de clave deseado:
  (1) RSA y RSA (por defecto)
  (2) DSA y ElGamal
  (3) DSA (sólo firmar)
  (4) RSA (sólo firmar)
  (14) Existing key from card
Su elección: 2
Las claves DSA pueden tener entre 1024 y 3072 bits de longitud.
¿De qué tamaño quiere la clave? (2048) 1024
El tamaño requerido es de 1024 bits
Por favor, especifique el periodo de validez de la clave.
  0 = la clave nunca caduca
  <n> = la clave caduca en n días
  <n>w = la clave caduca en n semanas
  <n>m = la clave caduca en n meses
  <n>y = la clave caduca en n años
¿Validez de la clave (0)? 0
La clave nunca caduca
¿Es correcto? (s/n) s
```

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

Nombre y apellidos: profesor
Dirección de correo electrónico: profesor@gmail.com
Comentario:
Ha seleccionado este ID de usuario:
  "profesor <profesor@gmail.com>"

¿Cambia (N)ombre, (C)omentario, (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/console, 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.
Es necesario generar muchos bytes aleatorios. Es una buena idea realizar
alguna otra tarea (trabajar en otra ventana/console, 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: /root/.gnupg/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave CE20C68DBFAFF8F marcada como de confianza absoluta
gpg: creado el directorio '/root/.gnupg/openpgp-revocs.d'
gpg: certificado de revocación guardado como '/root/.gnupg/openpgp-revocs.d'
claves pública y secreta creadas y firmadas.

pub  dsa1024 2021-10-28 [SC]
uid  BAA4912EB6F1284E5E0EA828CE20C68DBFAFF8F
uid  profesor <profesor@gmail.com>
sub  elg1024 2021-10-28 [E]

profesor@Juanca-PC:~$
```

- Con **list-keys** podemos comprobar que tenemos dos claves: las nuestras y las de alumno.

- **profesor\$ gpg --list-keys**

```
profesor@Juanca-PC:~$ gpg --list-keys
gpg: comprobando base de datos de confianza
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: nivel: 0 validez: 1 firmada: 0 confianza: 0-, 0q, 0n, 0m, 0f, 1u
/home/profesor/.gnupg/pubring.kbx
-----
pub   dsa1024 2021-10-28 [SC]
       DFF8A10C8CEAB3AD6533575B5CCAE263D2B597FA
uid     [desconocida] alumno <alumno@gmail.com>
sub   elg1024 2021-10-28 [E]

pub   dsa1024 2021-10-28 [SC]
       949C22E8693E83D1F4E97537D4A2EDBDA4731FAE
uid     [ absoluta ] profesor <profesor@gmail.com>
sub   elg1024 2021-10-28 [E]
```

- Para que **alumno** nos pueda cifrar el fichero firmado, debemos seguir el procedimiento conocido: exportar la clave pública de **profesor** e importarla en **alumno**. Los comandos a introducir en cada sesión serían:

- **profesor\$ gpg -a --export -o /tmp/profesor.pub profesor**
- **alumno\$ gpg --import /tmp/profesor.pub**

```
profesor@Juanca-PC:~$ gpg -a --export -o /tmp/profesor.pub profesor
profesor@Juanca-PC:~$
```

```
~
-> gpg --import /tmp/profesor.pub
gpg: clave D4A2EDBDA4731FAE: "profesor <profesor@gmail.com>" sin cambios
gpg: Cantidad total procesada: 1
gpg: sin cambios: 1
```



- En la sesión de **alumno** vamos a generar un nuevo fichero, lo cifraremos para que solo **profesor** pueda recuperarlo y lo firmaremos para que sepa que es nuestro.

- **alumno\$ fortune > mensaje**

```
~/firma1
→ touch mensaje

~/firma1
→ fortune > mensaje

~/firma1
→ cat mensaje
You are fighting for survival in your own sweet and gentle way.
```

- **alumno\$ gpg -a --sign --encrypt --recipient profesor mensaje**

```
~/firma1 took 15s
→ gpg -a --sign --encrypt --recipient profesor mensaje
gpg: 86E6CEE743AB0C46: No hay seguridad de que esta clave pertenezca realmente
al usuario que se nombra

sub  elg1024/86E6CEE743AB0C46 2021-10-28 profesor <profesor@gmail.com>
  Huella clave primaria: 949C 22E8 693E 83D1 F4E9 7537 D4A2 EDBD A473 1FAE
  Huella de subclave: 4415 8E28 6D17 6D68 FA49 0EAA 86E6 CEE7 43AB 0C46

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) s
```

- **alumno\$ cp mensaje.asc /tmp**

```
~/firma1 took 3s
→ ls
mensaje mensaje.asc

~/firma1
→ cp mensaje.asc /tmp
```



- El fichero **mensaje.asc** es ilegible y lleva dentro el contenido del fichero **mensaje** y la **firma**. Lo copiamos en **/tmp** para recuperarlo desde la sesión de **profesor**, descifrarlo, verificar la firma y recuperar el fichero **mensaje**. El parámetro será **decrypt**, porque también nos ofrece la confirmación de la firma. Nos pedirá la contraseña que protege la clave privada de **profesor**, necesaria para descifrar el fichero. Los comandos serían:

- **profesor\$ cp /tmp/mensaje.asc .**

```
profesor@Juanca-PC:~/firma1$ cp /tmp/mensaje.asc .
```

- **profesor\$ gpg --decrypt -o mensaje mensaje.asc**

- **profesor\$ cat mensaje**

```
profesor@Juanca-PC:~/firma1$ gpg --decrypt -o mensaje mensaje.asc
gpg: cifrado con clave de 1024 bits ELG, ID 86E6CEE743AB0C46, creada el 2021-10-28
      "profesor <profesor@gmail.com>"
gpg: Firmado el mar 02 nov 2021 18:50:25 CET
gpg:      usando DSA clave DFF8A10C8CEAB3AD6533575B5CCAE263D2B597FA
gpg: Firma correcta de "alumno <alumno@gmail.com>" [desconocido]
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de confianza!
gpg:      No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: DFF8 A10C 8CEA B3AD 6533 575B 5CCA E26
3 D2B5 97FA
profesor@Juanca-PC:~/firma1$ cat mensaje
You are fighting for survival in your own sweet and gentle way.
profesor@Juanca-PC:~/firma1$
```

- Si no tenemos la clave privada de **profesor**, no podemos descifrar ni, por tanto, recuperar el fichero ni comprobar la firma. Por ejemplo, podemos borrar nuestras propias claves con los parámetros **delete-secretkey** y **delete-key**. Después ya no funcionará ni descifrar ni verificar. El mensaje de error indicará que el fichero fue cifrado con una clave que ya no tenemos. Los comandos son:

- **profesor\$ gpg --delete-secret-key profesor**

```
profesor@Juanca-PC:~/firma1$ gpg --delete-secret-key profesor
gpg (GnuPG) 2.2.19; Copyright (C) 2019 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.

sec  dsa1024/D4A2EDBDA4731FAE 2021-10-28 profesor <profesor@gmail.com>

¿Eliminar esta clave del anillo? (s/N) s
¿Es una clave secreta! ¿Eliminar realmente? (s/N) s
profesor@Juanca-PC:~/firma1$
```

- **profesor\$ gpg --delete-key profesor**

```
profesor$ gpg --delete-key profesor
gpg (GnuPG) 2.2.20; Copyright (C) 2020 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.

pub  dsa1024/54F7B153A19DD73C 2021-10-29 profesor <profesor@gmail.com>

¿Eliminar esta clave del anillo? (s/N) s
```

- **profesor\$ gpg --list-key**

```
profesor@Juanca-PC:~/firma1$ gpg --list-key
/home/profesor/.gnupg/pubring.kbx
-----
pub  dsa1024 2021-10-28 [SC]
    DFF8A10C8CEAB3AD6533575B5CCAE263D2B597FA
uid  [desconocida] alumno <alumno@gmail.com>
sub  elg1024 2021-10-28 [E]
```

- ***profesor\$ gpg --decrypt mensaje.asc***
- ***profesor\$ gpg --verify mensaje.asc***

```
profesor@Juanca-PC:~/firma1$ gpg --decrypt mensaje.asc
gpg: cifrado con clave de 1024 bits ELG, ID 86E6CEE743AB0C46, creada el 2021-10-28
"profesor <profesor@gmail.com>"
gpg: descifrado fallido: No tenemos la clave secreta
profesor@Juanca-PC:~/firma1$ gpg --verify mensaje.asc
gpg: verify signatures failed: Error inesperado
profesor@Juanca-PC:~/firma1$
```