GnuPG básico

Introducción

GnuPG es una aplicación que sirve para encriptar mensajes, documentos ... GnuPG usa un sistema de claves públicas lo que quiere decir que cada usuario tiene una clave privada y una clave pública.

La clave privada es la que se usa para desencriptar aquello que nos envían encriptado con nuestra clave pública, La clave privada es una clave que solo ha de conocer el propietario ya que si alguien más la conociese podría desencriptar lo que nos mandan encriptado.

La clave pública es la que se da a la gente para que nos manden cosas encriptadas y usaran para encriptar aquello que nos quieran pasar.

El programa se puede obtener de la siguiente url http://www.gnupg.org

Creación de claves

Lo primero que hay que hacer una vez que se tiene GnuPG instalado es crear nuestra clave pública y privada. Para hacerlo hay usar el comando gpg --gen-key.

```
evolution:~# gpg --gen-key
gpg (GnuPG) 1.4.1; Copyright (C) 2005 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: directory `/root/.gnupg' created
gpg: creado un nuevo fichero de configuración `/root/.gnupg/gpg.conf'
gpg: AVISO: las opciones en `/root/.gnupg/gpg.conf' no están aún activas en esta ejecución
gpg: anillo `/root/.gnupg/secring.gpg' creado
gpg: anillo `/root/.gnupg/pubring.gpg' creado
Por favor seleccione tipo de clave deseado:
   (1) DSA and Elgamal (default)
   (2) DSA (sólo firmar)
   (5) RSA (sólo firmar)
Su elección: 1
```

Al ser la primera vez que se ejecuta nos crea un directorio en el que guardara el fichero de configuración así como los archivos secring.gpg y pubring.gpg. En el primero se almacenaran las claves privadas y en el segundo las claves públicas.

La primera pregunta que hace es que tipo de clave queremos. Lo normal suele ser seleccionar la primera opción (DSA and ElGamal) que nos permite encriptar y firmar.

```
DSA keypair will have 1024 bits.
ELG-E keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 2048
El tamaño requerido es de 2048 bits
```

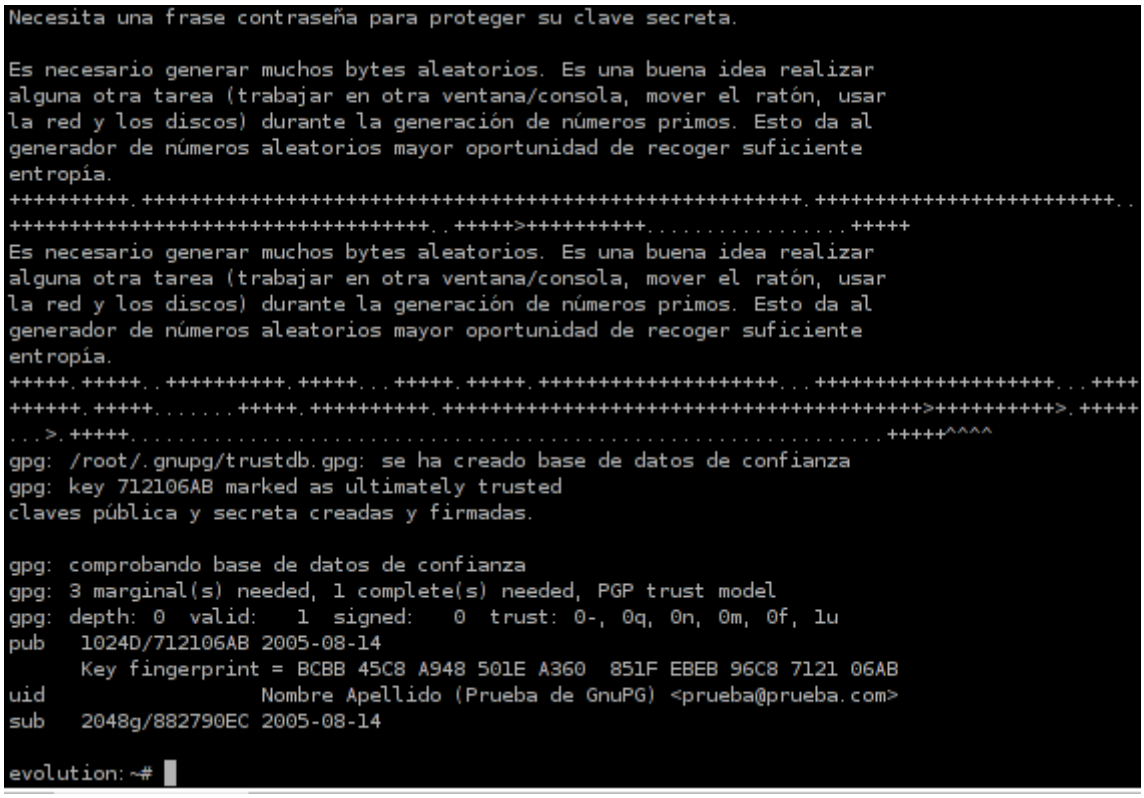
La siguiente pregunta es el tamaño de las claves que se puede elegir entre 1024 y 4096 bits. Por defecto se recomienda 2048, a mayor tamaño más segura es la clave. También a mayor tamaño más tiempo lleva encriptar y desencriptar.

```
Por favor, especifique el período 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
Key does not expire at all
Is this correct? (y/N) y
```

La siguiente pregunta es cuanto tiempo de validez queremos que tenga la clave. La periocidad se puede poner que no caduque nunca, que dure ciertos semanas, meses o años. En el caso de poner que caduque al cabo de cierto tiempo habrá que volver a generar las claves y volver a mandar la nueva clave pública a aquellos que usaban la que ha caducado. Por defecto viene la opción 0 que es que no caduque nunca.

```
You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Nombre y apellidos: Nombre Apellido
Dirección de correo electrónico: prueba@prueba.com
Comentario: Prueba de GnuPG
Ha seleccionado este ID de usuario:
    "Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>"

¿Cambia (N)ombre, (C)omentario, (D)irección o (V)ale/(S)alir? V
```

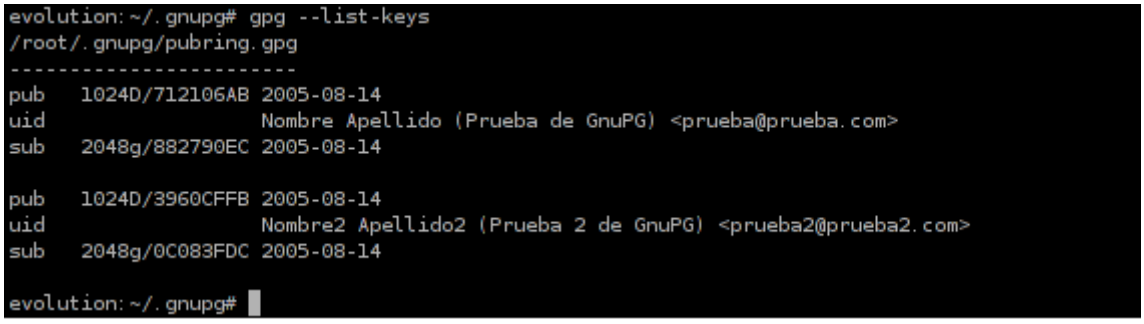
Ahora pregunta nuestro nombre y apellidos, dirección de correo y un comentario para la llave. Una vez introducidos todos los datos nos muestra cual es nuestro ID de usuario que lo crea a partir de los datos que le hemos introducido antes. Luego pregunta si queremos cambiar algún dato o si están bien los datos. Si estan correctos respondemos "V" y sigue adelante el proceso.



Por último se pregunta cual va a ser el password para nuestra clave privada. Al introducir el password no se ve nada de lo que se escribe ni se ve avanzar el cursror. Despu´s de introducirla nos vuelve a preguntar el password y si coincide con el primer password comienza la generación de las claves. Cuando se produce el proceso de generación de las claves es buena idea reproducir mp3, mover el ratón ... para que se generen números aleatorios y se creen antes las claves.

Ver claves públicas disponibles

Para ver las claves públicas que tenemos disponibles hay que hacerlo con el comando gpg --list-keys. Esto lo que haces listar las claves que hay disponibles dentro del fichero pubring.gpg.



El identificador de las claves es lo que hallamos metido en el nombre, en el apellido, en la dirección de correo o el número que aparece después del 1024D al hacer gpg --list-keys. Si en algún caso coincide el ID se mostrarán los que coinciden.

```
evolution:~# gpg --list-keys Nombre2
pub   1024D/3960CFFB 2005-08-14
uid                  Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>
sub   2048g/0C083FDC 2005-08-14

evolution:~# gpg --list-keys Apellido2
pub   1024D/3960CFFB 2005-08-14
uid                  Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>
sub   2048g/0C083FDC 2005-08-14

evolution:~# gpg --list-keys prueba2@prueba2.com
pub   1024D/3960CFFB 2005-08-14
uid                  Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>
sub   2048g/0C083FDC 2005-08-14

evolution:~# gpg --list-keys 0x3960CFFB
pub   1024D/3960CFFB 2005-08-14
uid                  Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>
sub   2048g/0C083FDC 2005-08-14

evolution:~# gpg --list-keys "Prueba 2 de GnuPG"
pub   1024D/3960CFFB 2005-08-14
uid                  Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>
sub   2048g/0C083FDC 2005-08-14

evolution:~# gpg --list-keys "Nombre"
pub   1024D/712106AB 2005-08-14
uid                  Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>
sub   2048g/882790EC 2005-08-14

pub   1024D/3960CFFB 2005-08-14
uid                  Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>
sub   2048g/0C083FDC 2005-08-14

evolution:~#
```

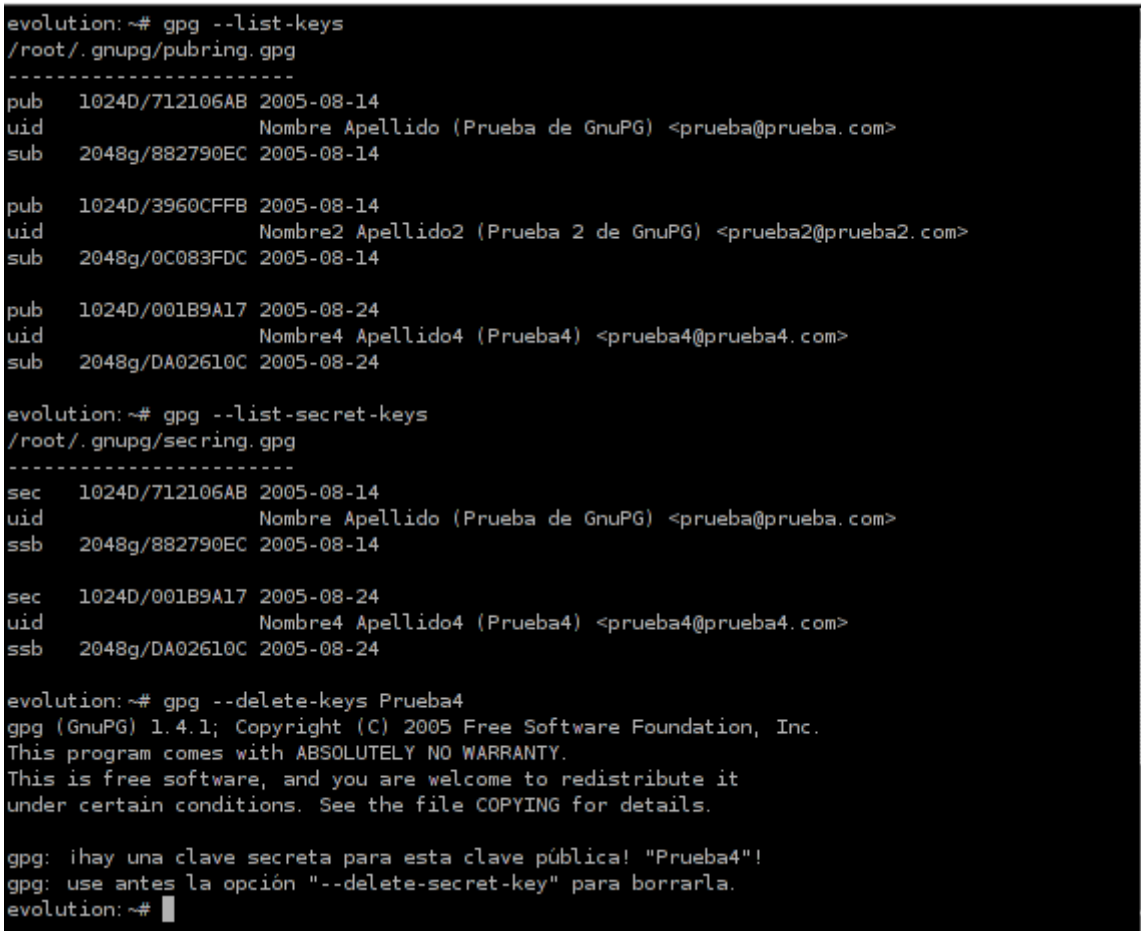
Ver claves privadas disponibles

Para ver las claves privadas que tenemos disponibles hay que hacerlo con el comando gpg --list-secret-keys. Esto lo que haces listar las claves que hay disponibles dentro del fichero secring.gpg.

```
evolution:~# gpg --list-secret-keys
/root/.gnupg/secring.gpg
------------------------
sec   1024D/712106AB 2005-08-14
uid                  Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>
ssb   2048g/882790EC 2005-08-14

evolution:~#
```

Borrar claves de los anillos

Se llama anillos a los archivos en los que se guardan las claves públicas y las privadas. Generalmente donde se guardan las claves públicas es el archivo pubring.gpg y en el que se guardan las claves secretas secring.gpg. Si se quiere borrar alguna clave primero hay que borrar la clave privada y después la pública. Si se intenta borrar primero la clave pública y esta tiene asociada una clave privada da un mensaje de error.



Para borrar claves privadas se hace con el comando gpg --delete-secret-key ClaveID
Para borrar claves públicas se hace con el comando gpg --delete-key ClaveID

```
evolution:~# gpg --delete-secret-keys Prueba4
gpg (GnuPG) 1.4.1; Copyright (C) 2005 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.


sec  1024D/93A63FAC 2005-08-24 Nombre4 Apellido4 (Prueba4) <prueba4@prueba4.com>

Delete this key from the keyring? (y/N) y
This is a secret key! - really delete? (y/N) y
evolution:~#
evolution:~# gpg --delete-keys Prueba4
gpg (GnuPG) 1.4.1; Copyright (C) 2005 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.


pub  1024D/93A63FAC 2005-08-24 Nombre4 Apellido4 (Prueba4) <prueba4@prueba4.com>

Delete this key from the keyring? (y/N) y
evolution:~# gpg --list-keys
gpg: comprobando base de datos de confianza
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
/root/.gnupg/pubring.gpg
------------------------
pub   1024D/712106AB 2005-08-14
uid                  Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>
sub   2048g/882790EC 2005-08-14

pub   1024D/3960CFFB 2005-08-14
uid                  Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>
sub   2048g/0C083FDC 2005-08-14

evolution:~#
evolution:~# gpg --list-secret-keys
/root/.gnupg/secring.gpg
------------------------
sec   1024D/712106AB 2005-08-14
uid                  Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>
ssb   2048g/882790EC 2005-08-14

evolution:~#
```

Ver huella de la clave

Las claves están identificadas por lo que se llama huella. La huella es una serie de números que se usa para verificar si una clave pertenece realmente al propietario. Si se recive una clave podemos ver cual es su huella y luego pedirle a su propietario que nos diga su huella. Si ambas coinciden la clave es correcta y no ha sido manipulada. Si no fuese igual es que ha sido modificada. La huella es como el md5 que verifica que un archivo no ha sido manipulado.

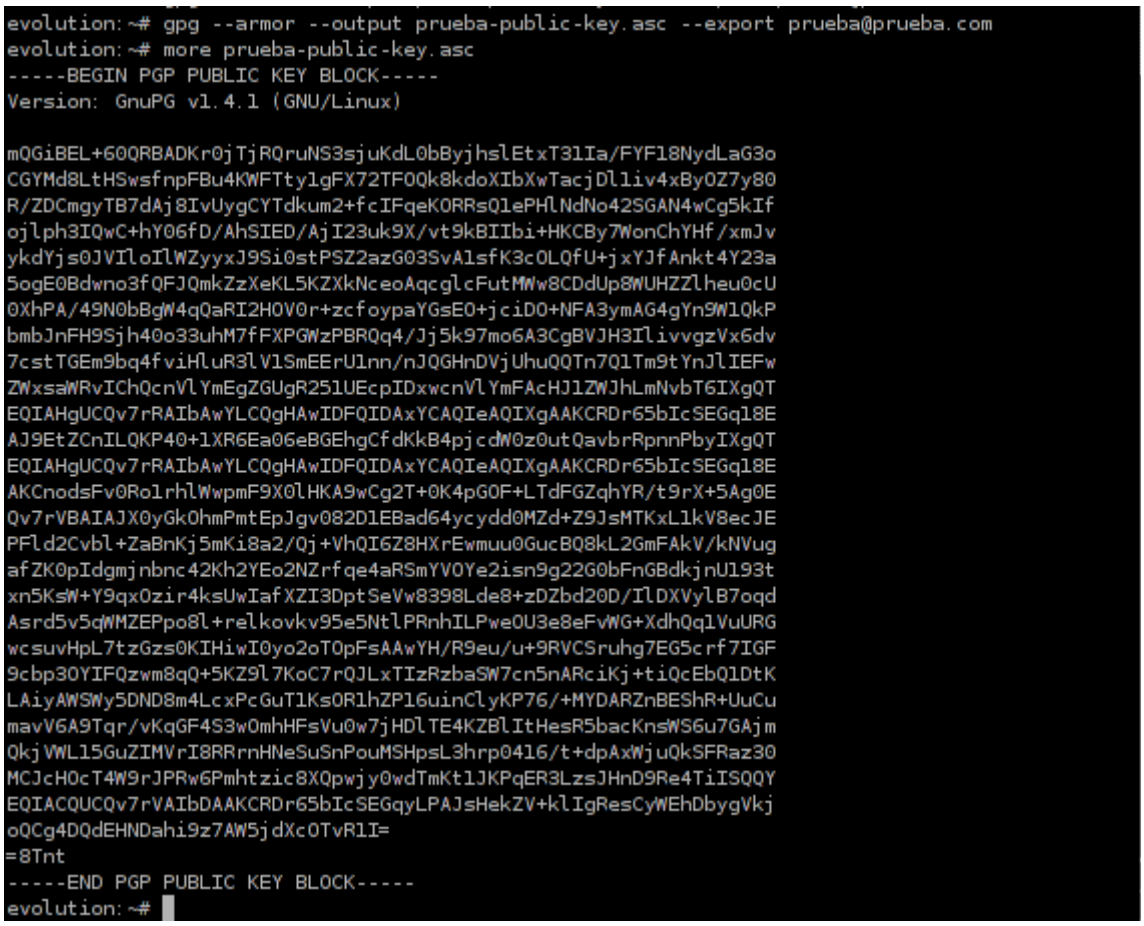
```
evolution:~/.gnupg# gpg --fingerprint prueba@prueba.com
pub   1024D/712106AB 2005-08-14
      Key fingerprint = BCBB 45C8 A948 501E A360  851F EBEB 96C8 7121 06AB
uid                  Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>
sub   2048g/882790EC 2005-08-14

evolution:~/.gnupg#
```

Exportar claves

Las claves se pueden exportar a ficheros para que las podamos distribuir entre la gente que queremos que nos encripte o firme cosas o bien porque vamos a formatear el equipo y necesitamos

salvarlas.

Para exportar la clave publica se hace poniendo gpg --armor --output fichoeDeSalida --export ClaveID



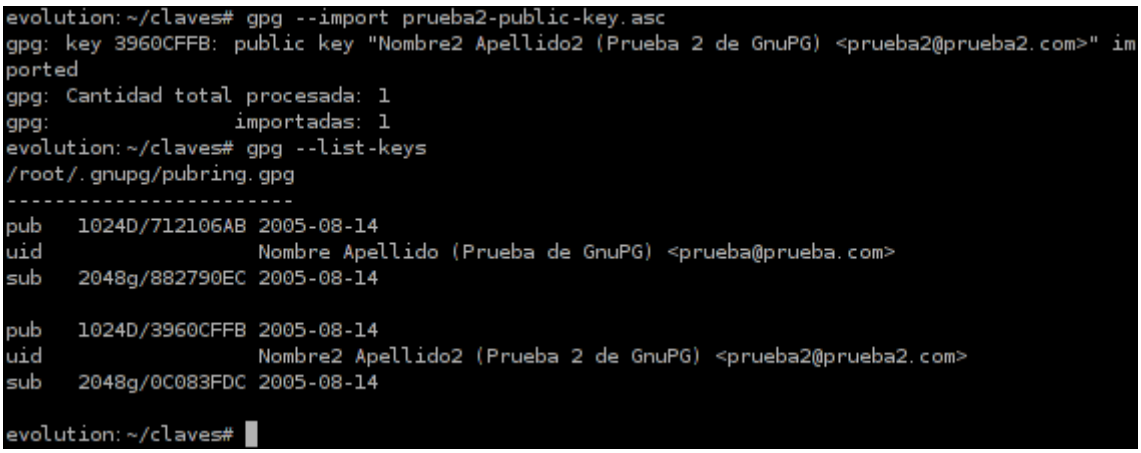
Para exportar la clave privada se haría poniendo gpg --armor --output fichoeDeSalida --export-secret-key ClaveID

Si quisieramos salvar todas las claves que tenemos valdría con copiar los archivos pubring.gpg y secring.gpg y luego cuando vayamos al nuevo equipo poneralas en el directorio de GnuPG.

Importar claves

Si se quiere importar claves nuevas porque por ejemplo hemos formateado el equipo y queremos volver a tener nuestras claves las importamos con el comando gpg --import ClaveID. En el apartado anterior se han salvado las claves publica y privada pues ahora vamos a importarlas. Primero importamos la publica y luego la privada.

Ahora si queremos importar la clave de una amigo pues se haría igual.



Encriptar mensajes

Si se quiere encriptar mensajes se puede hacer poniendo gpg --armor --recipient ClaveID --encrypt mensaje. Si por ejemplo queremos encriptar el archivo a.txt habría que poner gpg --armor --recipient prueba@prueba.com --encrypt a.txt

```
evolution:~/.gnupg# gpg --armor --recipient prueba@prueba.com --encrypt a.txt
evolution:~/.gnupg# more a.txt.asc
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.1 (GNU/Linux)

hQIOA2/sL+2IJ5DsEAf/QmXOGGFaxQ7vKGuoGUqltnxDOgIZ8cOdbrXpZh2HN+NU
IGPF/fWOhe/JKEWJDwXUg3VinXhkJc2fZsE007iEiXPM1XeUfFxAEALVrLlb80fU
vJyshR7K7OnCKk4mKTZgMtp9NxwFbzRbCI3N0dWlY9V9pBPsPiqhS9PFQ00L4DFd
/G/Yq6f3HMa+jW0o5aMexUuhHRCCyuMG9eN9La0LK8UsHSDsxEXgBHw7yM27Yrt0
WzUowL54tpceR/I66fyu+U4i5OF+2ybXfwiPs70iWokJf5hecRTfUt01cvGxH5CJ
/P/u8g9hKLYUjyc4hGku6RXF1YvNVZ3QnYoNPrQFEQf+Pfqe4zk3wSjmRi2scFhJ
kdseCbGeWAin9H8J5EsXMmudLUjBXsUEQQklHITmWpHHodkCHcmTs+2I6SpdofK4
kJJzluGyW1d4eB7V7da36EcsLrBNmtU4yAj2pIeFujZ4QdxZ4wL8cbClKkKwSnCG
GJuxRXKLksaLbEA5KSHFmg1jJWt31d0xZ80dCtfpuok8wZNLcxzXlaTOsDTLEQGB
gpIcFT8Pmb40loU/nPIDJOUbzEnY1P6AufLr0gkrt3kUexPSjGTu6qVmaBlJTcia
Gf3umrdzBaPtUYilPefAMDI4SrW3zbCbv0afJ9o6ILBVh6CnHNnVOBLKX8kMqwBP
LNJFASWLAr45E5iYmLfsreUpxbp8TWxPsGMGpg5q3N6cQ00+WcKZiVFRfcTkZDcy
rBGiEPcVVrdRR+4V78Rd9KakfPiaw5rm
=WB73
-----END PGP MESSAGE-----
evolution:~/.gnupg#
```

También se puede encriptar a un fichero en concreto con la opción --output nombreFichero

```
evolution:~/.gnupg# gpg --armor --output salida.txt --recipient prueba@prueba.com --encrypt a.
txt
evolution:~/.gnupg# more salida.txt
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.1 (GNU/Linux)

hQIOA2/sL+2IJ5DsEAf/Ytc78XgTlIbmVsomHcihRxMKF57i8lRP28gvWKY8Goet
06lma5die0dL55rUGT2z7aRshAK44eC6/kjHA00pvwDRhzPJ2jHuqyf13gI6CRUp
gSQ5qXZR5yMRP8r+hEgoSinogG8eHfa+3roXF6owneEARLrlc7MPHbgQvg/AIC8t
gBxt33CEBYlZh7hAaVgJT1/oYhWRFzgPHlOeAiKjflWzxj5tTsukV18AFuXo+1ms
P+z3RYnFAY0HFvUrlOkXbIY0E6bmzSTNU6e9cZc7ksPLyfwGxFZNhB5tTdXVk1Ud
yWdTRD4cZAddQ+IY4/RNeVGKYEkwkdRp0Jy+wtAJbgf/Xe2ipy0FljSka4k653Rn
dWOzmx98jaBRl0dTq/0vhSL5QlJDwyzJcYvL4tM1mB2kkLqOIzz/VbdmGkoRzMXp
xmvFD6JRUHCJv+6bQA+nOsnlM1qSxdhGFKyvTQhe/IwIR067WinmWS2xtOnl9WjJ
cHD9yxOKAC9k51FX1ofo4kRfFDQvg8KYcjDhmCu9PIAO19CD75foi/hvk9xUE2ON
i5y/BZNruTceqXpE6HKe/C8y4uE/KzKcnQ6GX8TcrSKvpp84/qkKL3rXWW1VVXVO
yvmoG7RzbkRTm4elLuMQ8B9cWGGal3/hAHf3UnaJR1pGyMiJyU7b4M/mISWoX3JZ
AdJFATzJqSSx/UnIJayJIxEfJJcP9w2wN3Peok/H+vyWbCkZAz88sxlePXoYPGCJ
8CLHPRIKnw6FgKtb6X6JF6F/V56ABzEw
=Eib/
-----END PGP MESSAGE-----
evolution:~/.gnupg#
```

Si en las opciones no se le pasa el parámetro --armor lo que se encripta lo deja en un archivo de tipo binario. Al poner la opción --armor transforma lo que se encripta en texto ASCII con el mensaje encriptado.

Desencriptar mensajes

Para desencriptar el mensaje que hemos encriptado antes hay que poner gpg --decrypt archivo. Para el caso anterior sería gpg --decrypt a.txt.asc.

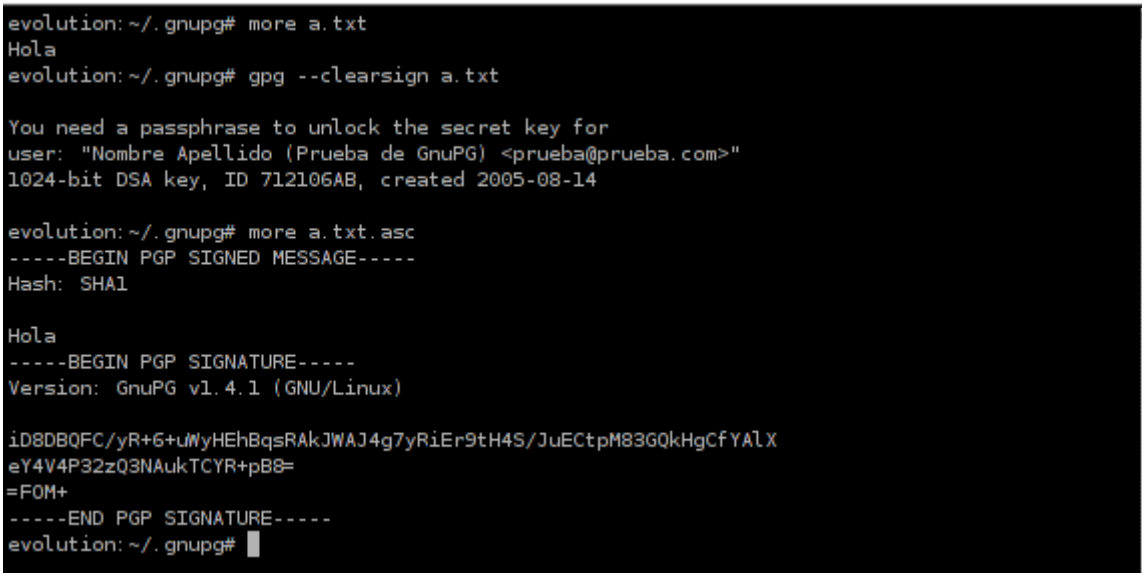
```
evolution:~/.gnupg# gpg --decrypt a.txt.asc

You need a passphrase to unlock the secret key for
user: "Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>"
2048-bit ELG-E key, ID 882790EC, created 2005-08-14 (main key ID 712106AB)

gpg: encrypted with 2048-bit ELG-E key, ID 882790EC, created 2005-08-14
      "Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>"
Hola
evolution:~/.gnupg# more a.txt
Hola
evolution:~/.gnupg#
```

Cuando desencriptamos algo se pide la password de nuestra clave para poder desencriptarlo. Para nuestro caso tenemos el archivo a.txt.asc encriptado al desencriptarlo nos deja el archivo a.txt y nos

muestra su contenido.

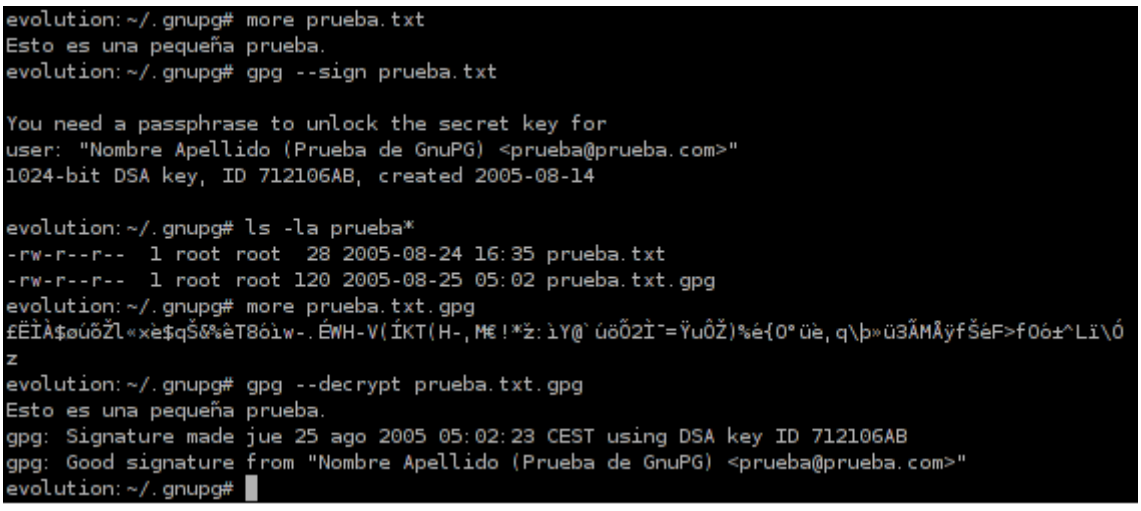
Firmar mensajes

Firmar mensajes sirve para que cuando a alguien le llegue un mensaje que hemos firmado la persona que lo ha recibido verifique con GnuPG que la firma es buena y que entonces hemos sido nosotros quien le ha enviado el mensaje. Por ejemplo vamos a firmar el archivo a.txt para ello se pondría gpg --clearsign a.txt. Esto nos creara el arhivo a.txt.asc con el contenido que se ve en la imagen.

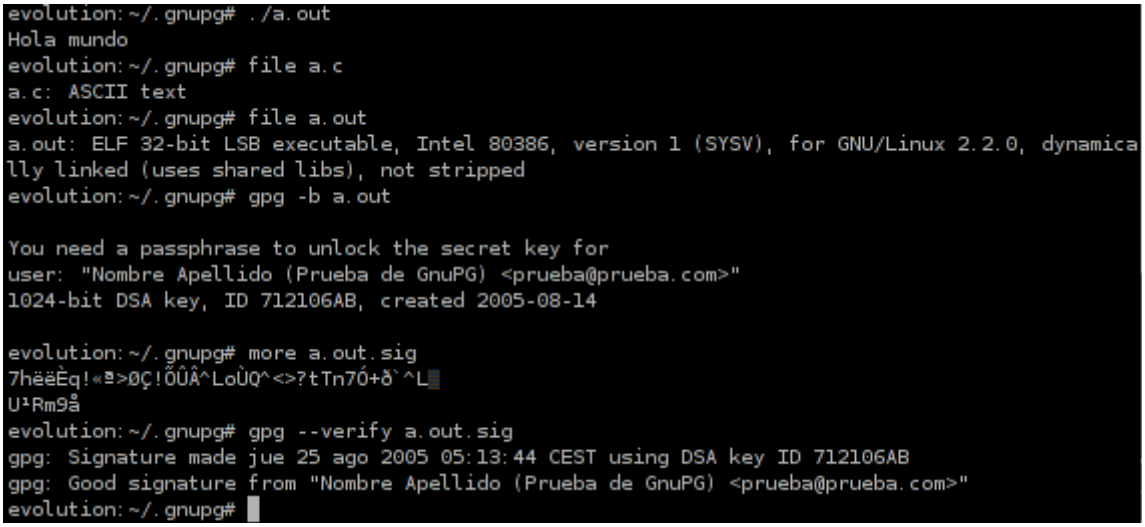


Para firmar algo se pide la contraseña para poder firmarlo. Como se ve en la imagen lo que se ha hecho en el fichero firmado es añadir unas líneas que contienen la firma.

A la hora de fimar si se firma con el parámetro --sign en lugar de --clearsign nos generara un fichero de salida en binario con extensión .gpg. Para validar la firma y ver el contenido hay desencriptarlo con la opción --decrypt.

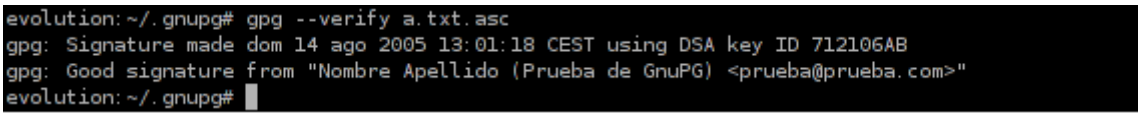


La firma también se puede hacer que se muestre en un fichero aparte con la opción -b. Esta opción se suele usar para firmar archivos binarios.



Verificar mensajes firmados

Para verificar mensajes firmados se hace poniendo gpg --verify mensaje. Para el caso anterior seria poner gpg --verify a.txt.asc



Si la firma no fuese correcta podríamos ver un mensaje como el siguiente:

Trabajar con claves en servidores

Podemos buscar claves públicas de gente a la que qeremos enviar mensajes cifrados en servidores de claves. Para hacer una busqueda se hace poniendo los parámetros --keyserver NombreDelServidor --search-keys ClaveID. Si encuentra claves que coinciden con esa ID nos las muestras, luego tenemos tres opciones mostrar los siguientes si es que se han encontrado muchos, poner el número del registro que nos interesa (en ese caso nos importa la clave al anilllo de claves públicas) o salir.

Si queremos importar una clave en concreto se hace con los parámetros --keyserver NombreDelServidor --recv-keys ClaveID.

Si queremos subir una clave a un servidor para que esté disponbile para la gente se hace con los parámetros --keyserver NombreDelServidor --send-keys ClaveID

GnuPG subshell

GnuPG viene con una especie de shell que nos da multitud de opciones para trabajar con la clave, Nos permite firmar la clave, cambiar la contraseña, cambiar la fecha de expiración de la llave ...

Para acceder a esta shell hay que poner el parametro --edit-key ClaveID.

Clave de revocación

La clave de revocación es una clave que lo que hace es que cuando la importemos a nuestro anillo de claves inavalide esa clave. Para generarla se hace con la opción --gen-revoke. Esta clave se puede crear nada más generar las llaves o bien cuando se halla comprometido. Hay gente que lo crea nada más crear las claves porque si por ejemplo ha olvidad la contraseña no podra generar la clave de recovación ya que al final del proceso de generación se pide la contraseña. Esta clave ha de guardarse en un lugar seguro ya que si alguien la obtuviese podría revocar nuestras claves y dejarnos las claves inutilizadas.

Si queremos revocar la clave hay que importar el fichero que tiene la clave de revocación y ya está. Una vez revocada la clave ya no podemos cifrar mensajes aunque si se pueden desencriptar, aunque al desencriptar se avisa de que la clave ha sido revocada..

```
evolution:~/.gnupg# gpg --import Nombre5Revoke.asc
gpg: key CE99AF10: "Nombre5 Apellido5 (Prueba5) <prueba5@prueba5.com>" revocation certificate
imported
gpg: Cantidad total procesada: 1
gpg:         nuevas revocaciones de claves: 1
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
evolution:~/.gnupg# gpg --decrypt nombre5.txt.asc

You need a passphrase to unlock the secret key for
user: "Nombre5 Apellido5 (Prueba5) <prueba5@prueba5.com>"
2048-bit ELG-E key, ID 1B182FD4, created 2005-08-25 (main key ID CE99AF10)

gpg: NOTA: la clave ha sido revocada
gpg: razón para la revocación: La clave ha sido comprometida
gpg: comentario a la revocación: La clave ya no es segura
gpg: encrypted with 2048-bit ELG-E key, ID 1B182FD4, created 2005-08-25
      "Nombre5 Apellido5 (Prueba5) <prueba5@prueba5.com>"
Archivo a cifrar por Nombre5.
evolution:~/.gnupg# gpg --armor -r Nombre5 --encrypt nombre5.txt
gpg: Nombre5: omitido: clave pública inutilizable
gpg: nombre5.txt: encryption failed: clave pública inutilizable
evolution:~/.gnupg#
```

Al invalidar la clave tampoco se pueden firmar mensajes.

```
evolution:~/.gnupg# gpg --default-key Nombre5 --sign nombre5.txt
gpg: no default secret key: clave secreta inutilizable
gpg: signing failed: clave secreta inutilizable
evolution:~/.gnupg#
```

Anillo de confianza

Crear un anillo de confianza consiste en tener claves de gente firmada por otra gente que la han firmado y que con su firma aseguran que esa clave es realmente de quien dice ser y no ha sido alterada.

Si por ejemplo tenemos las persona A y B. Las personas A y B son amigas y se intercambian entre ellas las claves públicas, verifican sus fingerprints para ver que las claves son las correctas y quedan para ver las claves que se han pasado son correctas. Entonces una vez verificado que todo es correcto cada uno firma la clave de su amigo. Ahora si por ejemplo yo obtengo la clave de B y veo que está firmada por A (que es una persona que conozco y en la que confió) entonces me fio de que esa clave es la clave correcta de B y la puedo usar. Si por un casual quedría firmar la clave de B con mi firma para abalar que su clave es buena sería bueno que me pusiese en contacto con él y verficásemos la clave.

```
evolution:~/.gnupg# gpg --sign-key Nombre2

pub  1024D/3960CFFB  created: 2005-08-14  expires: nunca       usage: CS
                     trust: desconocido   validity: desconocido
sub  2048g/0C083FDC  created: 2005-08-14  expires: nunca       usage: E
[ unknown] (1). Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>


pub  1024D/3960CFFB  created: 2005-08-14  expires: nunca       usage: CS
                     trust: desconocido   validity: desconocido
 Huella de clave primaria: 03ED F15D 8552 614A FB74  FE80 3616 E327 3960 CFFB

     Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>

Are you sure that you want to sign this key with your
key "Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>" (712106AB)

Really sign? (y/N) y

You need a passphrase to unlock the secret key for
user: "Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>"
1024-bit DSA key, ID 712106AB, created 2005-08-14


evolution:~/.gnupg# gpg --list-sigs
gpg: comprobando base de datos de confianza
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   1  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: depth: 1  valid:   1  signed:   0  trust: 1-, 0q, 0n, 0m, 0f, 0u
/root/.gnupg/pubring.gpg
------------------------
pub   1024D/712106AB 2005-08-14
uid                  Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>
sig 3        712106AB 2005-08-14  Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>
sig 3        712106AB 2005-08-14  Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>
sub   2048g/882790EC 2005-08-14
sig          712106AB 2005-08-14  Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>

pub   1024D/3960CFFB 2005-08-14
uid                  Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>
sig 3        3960CFFB 2005-08-14  Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>
sig          712106AB 2005-08-25  Nombre Apellido (Prueba de GnuPG) <prueba@prueba.com>
sub   2048g/0C083FDC 2005-08-14
sig          3960CFFB 2005-08-14  Nombre2 Apellido2 (Prueba 2 de GnuPG) <prueba2@prueba2.com>

pub   1024D/CE99AF10 2005-08-25 [revoked: 2005-08-25]
rev          CE99AF10 2005-08-25  Nombre5 Apellido5 (Prueba5) <prueba5@prueba5.com>
uid                  Nombre5 Apellido5 (Prueba5) <prueba5@prueba5.com>
sig 3        CE99AF10 2005-08-25  Nombre5 Apellido5 (Prueba5) <prueba5@prueba5.com>

evolution:~/.gnupg#
```

Programas que soportan GnuPG

Para utilizar GnuPG de forma gráfica y no tener que estar escribiendo los comandos podemos encontrar para KDE el programa KGPG y para Gnome el programa SeaHorse.

Hay algunos clientes de correo que también soportan GnuPG como Evolution o Mozilla Thunderbird al que hay que ponerle el plugin Enigmail para que tenga soporte de GnuPG.

Referencia del Manual:

http://www.lostscene.com/manuales/gnupg.php