Manual de GPG: cifra, firma y envía datos de forma segura

Tras 3 meses de uso, el MacBook Air M1 es un portátil brillante lastrado por su software: macOS sigue sin estar a la altura del hardware

Síguenos

23 Enero 2013 **PEDRO GUTIÉRREZ**

Hemos hablado ya sobre la criptografía simétrica y asimétrica y sobre la firma digital, solo toca ponerlo en práctica con **GnuPG** (la versión libre de **PGP** o mejor dicho *Pretty Good Privacy*), con el que cifraremos cualquier tipo de archivo que podremos mandar "libremente" con cierta seguridad de que nadie lo podrá leer. Y puede que a muchos os ayude a entender como funciona la criptografía y como funciona el polémico cifrado de **Mega**, que tanto está dando que hablar estos días.

¿Qué es GnuPG?

Antes de empezar con lo interesante tenemos que saber que es **GPG** (*GNU Privacy Guard*), que es un derivado libre de **PGP** y su utilidad es la de cifrar y firmar digitalmente, siendo además multiplataforma (podéis descargarlo desde la página oficial) aunque viene incorporado en algunos sistemas **Linux**, como en **Ubuntu** (será con el sistema que haré todos los ejemplos, en Windows se encuentra solo con gestor gráfico).

Anillo de claves

GPG tiene un repositorio de claves (*anillo de claves*) donde guarda todas las que tenemos almacenadas en nuestro sistema, ya sean privadas o públicas (como comenté, con la clave pública cifraremos un mensaje que solo podrá descifrar el que posee la clave privada).

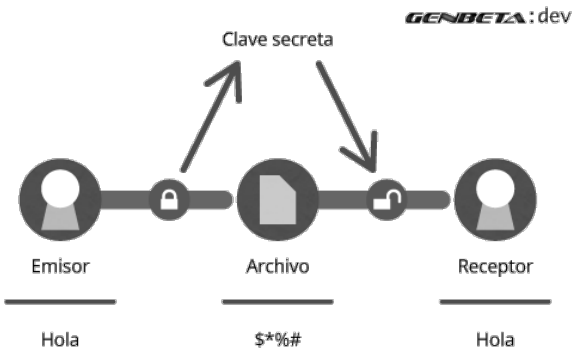
Más adelante cuando veamos un anillo de claves debemos de recordar que `pub` hace referencia a la clave pública y `sub` hace referencia a la privada (y que tenemos que tener a buen recaudo).

Servidores de claves

Para que nos cifren un mensaje tenemos que compartir la clave pública de nuestro par de claves para cifrar, y como es un poco engorroso difundir una clave a muchas personas existen los servidores de claves PGP (compatibles con GPG), donde subiré una clave pública para el que quiera probar los

ejemplos.
Unos ejemplos de servidores son estos: `pgp.rediris.es` (español, aunque falla algunas veces) o `pgp.mit.edu` (americano, del **MIT** y a mi no me ha dado problemas).

Cifrado simétrico



Como ya sabéis el cifrado simétrico es el tipo de cifrado más sencillo que hay, es más rápido de procesar y por desgracia menos seguro que el cifrado asimétrico.

Para empezar la prueba tenemos que tener un archivo de cualquier tipo e introducir en la terminal de Linux el comando `gpg` con el parámetro `-c` para cifrar y `-d` para descifrar.

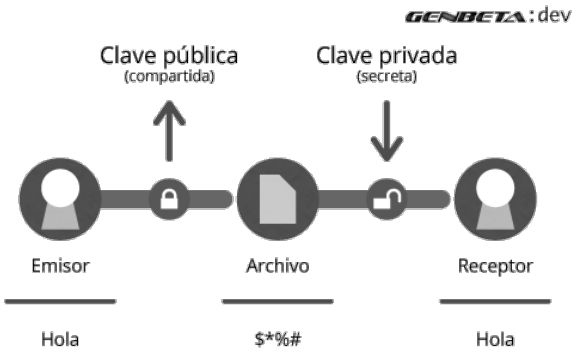
```
pedro@ubuntu:~/gpg$ echo "Genbeta Dev" > texto.txt
pedro@ubuntu:~/gpg$ gpg -c texto.txt
```

Tras crear un archivo de texto usamos el comando `gpg -c [archivo]`, nos aparecerá un cuadro que nos pide la contraseña y se generará un archivo `.gpg`. Y después lo descifraremos con el comando `gpg -d [archivo]` (e introduciendo la clave *de alta seguridad*, en este caso `qwerty`).

```
pedro@ubuntu:~/gpg$ gpg -d texto.txt.gpg
gpg: datos cifrados CAST5
gpg: cifrado con 1 frase contraseña
Genbeta Dev
gpg: AVISO: la integridad del mensaje no está protegida
```

Podéis probar a descifrar este archivo usando la clave `qwerty`.

Cifrado asimétrico



Generar las claves

Para poder cifrar asimétricamente primero tenemos que crear la pareja de claves (pública y privada) con el comando `gpg --gen-key`.

```
pedro@ubuntu:~/gpg$ gpg --gen-key
gpg (GnuPG) 1.4.11; Copyright (C) 2010 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.
Por favor seleccione tipo de clave deseado:
   (1) RSA y RSA (predeterminado)
   (2) DSA y Elgamal
   (3) DSA (sólo firmar)
   (4) RSA (sólo firmar)
¿Su selección?:
```

GPG nos permite elegir el tipo de clave que queremos usar, hay opciones que solo permiten firmar y otras que permiten firmar y cifrar, en este caso usaremos `DSA y Elgamal`.

```
las claves DSA pueden tener entre 1024 y 3072 bits de longitud.
¿De qué tamaño quiere la clave? (2048)
```

Nos piden el tamaño de la clave que puede variar entre 1024 bits y 3072, esto es de libre elección, yo tomaré el término medio que es el que propone por defecto (2048).

A partir de aquí todo es más trivial, nos pide la fecha en la que expirará la clave, la información del emisor de la clave (nombre, mail y algunos datos extra que queramos dar) y por último nos pedirá la contraseña que salvaguarda la clave privada.

Tras generar las claves podemos verlas con el comando `gpg -k` que nos muestra nuestro anillo de claves, lo importante de este paso es que veremos la identificación de cada una, que es necesaria para poderlas exportar y enviar.

```
pedro@ubuntu:~/gpg$ gpg -k
/home/pedro/.gnupg/pubring.gpg
------------------------------
pub   2048D/18384645 2013-01-23
uid                  Pedro Gutiérrez (Manual GPG - Genbeta Dev) <info@xitrus.es>
sub   2048g/C4A9EA7A 2013-01-23
```

Exportar y enviar la clave privada

El objetivo de esta pareja de claves es **que cualquiera nos pueda mandar un archivo cifrado** que solo veremos nosotros y esto se hace difundiendo la clave pública que acabamos de crear (la pública, **nunca** la privada), para exportarla en un archivo usaremos el comando `gpg -output [archivo destino] --export [ID de a clave pública]` (la clave pública generada antes tiene la ID `18384645`).

```
pedro@ubuntu:~/gpg$ gpg --output CPub.gpg --export 18384645
pedro@ubuntu:~/gpg$ ls
CPub.gpg
```

Este archivo ahora se puede difundir por el medio que queramos, tenemos que tener en cuenta que el único problema de seguridad que habría en difundir la clave es que alguien se hiciese pasar por otro al mandarnos un mensaje, algo que pasaría igual si no estuviese cifrado, por eso el que nos envíe algo lo debería de firmar (si fuese pertinente).

Podéis descargar esta clave pública, que ahora veremos como importar y sirve para mandarme un archivo cifrado o para comprobar que un archivo lo he firmado yo.

Subir una clave pública a un servidor de claves

Los servidores de claves suelen ser de acceso público (al no haber mucho problema por difundir una clave pública) y en este caso subiremos una clave a los servidores del **MIT** (`pgp.mit.edu`) usando el comando `gpg --send-keys --keyserver [Dirección del servidor] [ID de la clave pública]` (al igual que antes la ID es `18384645`).

```
pedro@ubuntu:~/gpg$ gpg --send-keys --keyserver pgp.mit.edu 18384645
gpg: enviando clave 18384645 a hkp servidor pgp.mit.edu
```

A partir de este momento la clave estará accesible desde este servidor específico.

Importar la clave desde el archivo o servidor de claves

Para poder usar la clave pública para cifrar o comprobar la identidad del remitente tenemos que importar previamente la clave, desde un archivo debemos de usar el comando `gpg --import [Archivo de la clave pública]` (el que hemos descargado anteriormente).

```
pedro@ubuntu:~/gpg$ gpg --import CPub.gpg
gpg: clave 18384645: «Pedro Gutiérrez (Manual GPG - Genbeta Dev) <info@xitrus.es>»
gpg: Cantidad total procesada: 1
gpg:               sin cambios: 1
```

Al tener la clave ya en mi anillo de claves me contesta que no hay cambios.
Para realizar la importación desde el servidor tenemos que usar el comando `gpg --keyserver [Dirección del servidor] --recv-keys [ID de la clave]`.

```
pedro@ubuntu:~/gpg$ gpg --keyserver pgp.mit.edu --recv-keys 18384645
gpg: solicitando clave 18384645 de hkp servidor pgp.mit.edu
gpg: clave 18384645: «Pedro Gutiérrez (Manual GPG - Genbeta Dev) <info@xitrus.es>»
gpg: Cantidad total procesada: 1
gpg:               sin cambios: 1
```

Como podemos ver al tener ya la clave nos devuelve el mismo mensaje.

Cifrar con la clave pública

Ahora tenemos que pensar que hemos importado una clave pública, por ejemplo de nuestro jefe y tenemos que mandarle un documento, para cifrar el documento usaremos el comando gpg —encrypt —recipient [ID de la clave] [Archivo]

```
pedro@ubuntu:~/gpg$ echo "Genbeta Dev" > documento.txt
pedro@ubuntu:~/gpg$ gpg --encrypt --recipient 18384645 documento.txt
pedro@ubuntu:~/gpg$ ls
documento.txt documento.txt.gpg
```

Y ya tenemos el archivo listo para mandarlo de forma segura.

Descifrar un archivo con la clave privada

Y ahora es el momento de descifrar con nuestra clave privada el documento tras recibirlo, con el comando `gpg -d [Archivo]` e introduciendo la contraseña que creamos para salvaguardar la clave privada.

```
pedro@ubuntu:~/gpg$ gpg -d documento.txt.gpg
Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Pedro Gutiérrez (Manual GPG - Genbeta Dev) <info@xitrus.es>"
clave ELG-E de 2048 bits, ID C4A9EA7A, creada el 2013-01-23 (ID de clave primaria 1
gpg: cifrado con clave ELG-E de 2048 bits, ID C4A9EA7A, creada el 2013-01-23
      «Pedro Gutiérrez (Manual GPG - Genbeta Dev) <info@xitrus.es>»
Genbeta Dev
```

Y el resultado nos lo muestra a continuación (`Genbeta Dev`), aunque si queremos especificar la salida debemos de usar el parámetro `-o [Archivo de salida]`.

Firmar archivos

Una de las medidas de seguridad básicas al pasar un mensaje es asegurarnos que el emisor es quien dice ser, para asegurarnos de esto digitalmente existe la firma digital, en el artículo anterior expliqué como **GPG** usaba los **hash** para crear una firma simple, pero también podemos cifrarlo y a su vez firmarlo, que es lo que haremos con el comando `gpg -u [ID de la clave privada] --output [Archivo resultante] --sign [Archivo para firmar]` e introduciendo la contraseña de la clave privada.

```
pedro@ubuntu:~/gpg$ echo "Genbeta Dev" > firmar.txt
pedro@ubuntu:~/gpg$ gpg -u C4A9EA7A --output firmar.txt.gpg --sign firmar.txt
Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Pedro Gutiérrez (Manual GPG - Genbeta Dev) <info@xitrus.es>"
clave DSA de 2048 bits, ID 18384645, creada el 2013-01-23
```

Y ahora para asegurarse la confidencialidad del documento (ahora que esta firmado por nosotros) deberíamos de cifrarlo con la clave pública del destinatario.

Verificar y descifrar un archivo firmado

Cualquiera con la clave pública asociada a la que ha firmado el documento puede leerlo, de la misma forma que desciframos un archivo (`gpg -d [Archivo]`) o verificándolo únicamente con el comando `gpg --verify [Archivo]`.

```
pedro@ubuntu:~/gpg$ gpg --verify firmar.txt.gpg
gpg: Firmado el mié 23 ene 2013 05:25:18 CET usando clave DSA ID 18384645
gpg: Firma correcta de «Pedro Gutiérrez (Manual GPG - Genbeta Dev) <info@xitrus.es>
```

Y el resultado es la información del remitente, que podéis comprobar vosotros con este archivo y con

la clave pública de los pasos anteriores.

Resumen

GPG es una herramienta de cifrado muy potente y fácil de usar, que en principio, a la mayoría no nos hace falta, pero puede que se nos presente la necesidad de enviar algo por medio inseguros (porque no haya más remedio), haciéndolo de esta forma podremos hacerlo sin miedo ha que lean el contenido del archivo o nos den el cambiazo.
Este es un tema bastante extenso y si no lo acabas de entender puede ser un lío, pero para preguntar están los comentarios.

En Genbeta Dev | Criptografía