

Criptografía de clave pública para Linux: GPG

GPG es libre y nos permite criptografía fuerte de clave pública, sin hacer uso ni del algoritmo IDEA ni del RSA que tienen unas licencias problemáticas para ser incluidas en un producto realmente libre.

Clave privada

La idea es que existe una palabra clave que es conocida por el emisor y por el receptor del mensaje, sin embargo, tiene algunas vulnerabilidades especialmente delicadas. La más importante es la del modelo del *asedio al castillo*.

Supongamos que estamos en un castillo que por sorpresa ha sido asediado. Las tropas de apoyo llega y las fuerzas están equilibradas entre las tropas de asedio y las de apoyo. De que seamos capaces de organizar un plan depende, por tanto, el éxito de la batalla.

Para coordinar el plan necesitamos hacer llegar información secreta y leer de las tropas de apoyo. Sin embargo, por el hecho de estar sitiados, no existe forma de asegurarnos que el enemigo no a leer nuestros mensajes. El ejemplo más común es el correo electrónico.

La primera idea que se nos puede ocurrir es usar un método de clave privada. Sin embargo, la clave debe ser conocida por el emisor y el receptor, si no, sería imposible poder descifrar lo que tratan de decirnos del otro lado, por otro lado, no podemos transmitir la clave secreta porque el enemigo puede escuchar.

Podemos encontrar el caso de que un individuo puede introducir mensajes falsos en el canal, aunque no pueda leer. Es el caso particular del correo electrónico, donde cualquiera puede hacer un telnet a la puerta de correo y mandar un mensaje haciéndose pasar por otra persona. No siempre es posible autenticar un mensaje a partir de la dirección IP que envió el correo.

Para que el GPG sea seguro, es conveniente seguir algunas prácticas:

- Comprobar la firma de las claves públicas que nos llegan, por eso es conveniente que al mandarnos una clave pública, se verifique la firma para asegurarnos que pertenece a quien realmente dice.
- Cuidado con nuestra cuenta de usuario. Si entran, pueden capturar nuestra clave privada y/o sustituir las claves públicas verdaderas que tengamos almacenadas.
- Cuidado con la seña del GPG. Si nos pillan la cuenta de usuario o el disco donde tenemos los archivos de GPG, todavía no pueden suplantar el correo, pero si cae la clave de GPG, pueden hacerse pasar por nosotros.

Instalación de GPG

Una vez descargado, podemos instalar el GPG directamente de un RPM o si nos encontramos con un tarball, podemos utilizar las siguientes instrucciones, después de haberlo descomprimido:

```
./configure  
make  
make install
```

Estas instrucciones se deben ejecutar dentro del directorio donde se descomprimió el tarball.

Creación de las claves

El primer paso es crear las claves.

```
gpg -gen-key
```

Preguntará si queremos trabajar con DSA, El Gammal o los dos, escogemos ambos y después

pregunta el tamaño de la clave, generalmente entre 768 y 2048, pero mientras mayor sea, mejor aunque mas lenta. Después preguntará si queremos realmente una clave tan larga y confirmamos. Al preguntar por la caducidad de la clave, escogemos que no expira nunca y confirmamos.

Después preguntará por nuestro nombre y una vez introducidos nuestros datos, se lista el resultado, si nos agrada, aceptamos, de lo contrario repetimos el proceso. Después pedirá la clave que el programa utilizará cada vez que empleemos nuestra llave privada, para ello se recomienda una razonablemente larga y no la misma de nuestra cuenta, puesto que si cae, todo el sistema se vera comprometido.

Después intentará buscar un número primo aleatorio grande, lo recomendable es realizar otra cosa en la máquina puesto que esto permite al motor de números aleatorios funcionar mejor. El proceso de generación de entropía suele ser bastante lento, pero una vez terminado, habrán quedado listas la llave privada y la pública.

Intercambiando claves

Lo lógico es tener nuestra clave pública en un formato utilizable por otros, esto lo podemos hacer como abajo se describe. La primera opción es por si lo queremos en formato binario, mientras que la segunda opción es para modo texto. De esta manera tendremos un archivo llamado miLlaveBinaria o miLlaveTexto en binario o en texto respectivamente.

```
gpg -export > miLlaveBinaria
gpg -export -armour > miLlaveTexto
```

Por otro lado, si lo que queremos hacer es importarla, debemos hacer:

```
gpg -import archivo
```

donde archivo es la clave pública que deseamos importar

Cifrando y descifrando

Para cifrar un archivo debemos tener la clave pública del destinatario y debemos haberla importado en nuestro anillo e señas de la siguiente forma:

```
gpg -er destinatario archivoMensaje
gpg -armour -er destinatario archivoMensaje
```

donde destinatario es el nombre completo del destinatario (entre comillas) y archivoMensaje es el nombre del archivo donde está el mensaje, lo cual generará un archivo llamado archivoMensaje.gpg en formato binario para la opción 1 o archivoMensaje.asc para la opción 2 en formato texto que podremos mandar al destinatario.

Ejemplo:

```
gpg -armour -er "Leonardo Romero Gutierrez" mimgsgparaLeo
ó
gpg -armour -er leorogu@servidor.org mimgsgparaLeo
```

para mandar el archivo mimgsgparaLeo cifrado con salida de texto

También es recomendable firmar el mensaje cifrado para evitar cualquier suplantación. Si el problema es saber como se llama el destinatario, basta con ejecutar lo siguiente para listar las firmas que tenemos en el anillo con el dueño.

```
gpg -fingerprint
```

Para descifrar es mucho mas sencillo, solo se tiene que ejecutar la siguiente instrucción donde archivo.asc es el archivo cifrado que se ha recibido.

```
gpg archivo.asc
```

Firmando archivos

Para firmar un archivo, con la firma incluida en el fichero, podemos hacer:

```
gpg -sign archivo  
gpg -clearsign archivo
```

lo cual genera un archivo de nombre archivo.gpg firmado y codificado en binario para la opción 1 y en texto para la opción 2, pero no cifrado. Esta opción es convenient, por ejemplo, para mandar correo certificado, así podemos firmar un archivo y que la firma este en un archivo aparte, parecido a la lógica de CRC o checksum, esto se hace:

```
gpg -b archivo
```

La firma estará en un archivo aparte, llamado archivo.sig. Es especialmente recomendable para firmar binarios que vamos a distribuir, pues así no modificamos el binario.

Si nos queremos ahorrar la firma y cifrado en un solo paso, hacemos:

```
gpg -u remitente -er destinatario -sign mensaje
```

donde remitente es el nombre que tenemos en el anillo de claves, destinatario al que tiene que llegar y mensaje lo que se debe cifrar, lo cual genera un archivo mensaje.gpg en binario, para tenerlo en modo texto hay que agregar la opción -armor. Solo que de esta manera el destinatario será el único que podrá verificar la firma, mientras que las formas anteriores lo permiten a cualquier usuario.

Para verificar una firma, ejecutamos lo siguiente:

```
gpg -verify archivo
```

La firma se verifica automáticamente al descifrar el mensaje. Cabe destacar que las firmas de GPG cumplen la Open PGP V4, mientras que PGP apenas cumple la OpenPGP v3. Para permitir que el PGP pueda verificar la firma generada por GPG, es necesario anexar la opción --force-v3-sigs.