Criptografía asimétrica con GnuPG

Un poco de historia

Aunque la criptografía asimétrica comenzó su andadura mucho antes, no fue hasta 1991 cuando **Phil Zimmermann** construye una herramienta que usa tecnología en un ordenador personal y por tanto, algo que hasta ese momento estaba en manos de gobiernos o corporaciones que pudieran permitírselo, puede entrar en todos los hogares.

No fueron pocos los problemas a los que se enfrentó Zimmermann, ya que su software no estaba bien visto por el gobierno de los Estados Unidos que veía que la criptografía que se estaba usando su ejercito llegaba al ciudadano. Además se apoyaba en un algoritmo que estaba patentado: el **RSA** (en honor a sus inventores: Rivest, Shamir y Adleman).

Zimmerman funda la compañía **PGP** (Pretty Good Privacy) [1] para dar salida comercial a su software. En 1998, esta compañía propone al **IETF** el estándar **OpenPGP** [2]. Al ser un estándar, cualquiera puede realizar la implementación del mismo y esto mismo lo que supone la llegada en 1999 de **GnuPG** [3] (Gnu Privacy Guard). Así tenemos:

1. **PGP.** Se refiere a PGP Corporation y a sus productos que implementan el. En junio de 2010 esta compañía fue adquirida por Symantec Corp.
2. **OpenPGP.** Estándar del IETF para comunicaciones con criptografía asimétrica.
3. **GnuPG.** Implementación GNU del estándar anterior, que está presente en todas las distribuciones GNU/Linux.

Para qué sirve

La criptografía asimétrica nos ofrece a**utenticidad**, que consiste en:

a) **Confidencialidad.** Cifrando las comunicaciones.

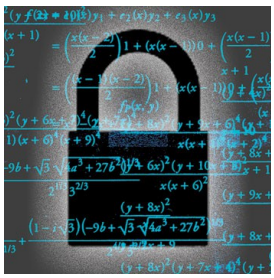
b) **No repudio.** Mediante firma electrónica.

c) **Integridad.** El mensaje que recibimos es de quien dice ser y contiene lo que el remitente escribió.

Algunos **usos** de este tipo de criptografía.

1. Cifrado y descifrado de mensajes
2. Firmado y verificación de mensajes.
3. Acceso seguro a servicios remotos.
4. Firmado de código.

Cómo funciona



Esta criptografía se basa en **dos claves** distintas (de ahí el nombre de criptografía asimétrica). Una de las claves se denomina **pública** y la otra **privada**.

La clave **pública** (como su nombre indica) puede hacerse pública, por contra la clave **privada** sólo es conocida por el propietario de la misma.

Cuando una persona quiere **firmar digitalmente** un mensaje usa su **clave privada**, de esta forma cualquier persona que posea la clave pública del remitente podrá comprobar que el mensaje ha sido firmado correctamente.

Para **cifrar** un mensaje se usa la **clave pública del destinatario**, así cuando éste reciba el mensaje podrá usar su clave privada para descifrarlo y por tanto sólo él puede ver el contenido del mensaje

Manos a la obra

Existen varias **interfaces gráficas de usuario** [4] (frontends) para trabajar con GnuPG que nos harán la vida más sencilla. En este artículo se muestran las ódenes que en muchas ocasiones ejecutan estas interfaces para los que deseen saber más o simplemente trabajar usando la línea de comandos [5].

Generación de las claves

El primer paso es **generar** nuestras propias claves (pública y privada) que se necesitan en cualquier sistema de criptografía asimétrica. Para ello usamos la orden:

```
$ gpg --gen-key
```

Esta orden es **interactiva** y nos permite generar las claves basándose en la información que le iremos suministrando:

1. **Tipo de clave.** Lo normal es seleccionar la opción 1 (RSA)
2. **Longitud de la clave.** Los 2048 bits que se ofrecen por defecto nos servirán perfectamente.
3. **Caducidad de la clave.** Es importante darle caducidad a nuestra clave. Una validez de un año será suficiente. Por lo tanto teclearemos 1 y pulsamos la tecla Intro.
4. **Nombre y Apellidos.** Conviene no usar tildes ni eñes si nuestro nombre las contiene, ya que vamos a publicar esta clave en un servidor y podría haber problemas a la hora de leer nuestro nombre completo.
5. **Dirección de correo electrónico.** La dirección de correo electrónico asociada a nuestra clave.
6. **Comentario.** Si lo deseamos podemos añadir información extra, por ejemplo a qué nos dedicamos o para qué usamos la clave.
7. **Contraseña.** Para poder usar nuestra clave, ésta debe ser desbloquedada, por lo que se pide

una contraseña que sólo debemos conocer nosotros y que tendremos que teclear cuando queramos usar la clave. Conviene elegir una buena contraseña para que nuestra clave privada esté bien protegida.

Una vez tecleada (y confirmada) nuestra contraseña, el sistema comenzará a generar las claves. Tal y como muestra la orden, conviene que mientras se están generando las claves, movamos el ratón, usemos el teclado y el disco para generar eventos aleatorios que el sistema pueda utilizar para generar las claves y por tanto tardar menos en generarlas. Cuando termine la generación, nuestras claves estarán en el directorio `.gnupg` de nuestro directorio personal.

Conviene proteger adecuadamente este directorio para que únicamente nosotros podamos acceder a él, así como realizar una copia de seguridad del mismo por si ocurriera un error grave en nuestro sistema. Es importante recordar que si perdemos la clave privada no podremos firmar ni descifrar mensajes y deberemos crear un nuevo par de claves.

Identificador y huella digital de la clave

Cuando se genera una clave, se le asocia un **identificador**. Este identificador permite hacer referencia a nuestra clave en sucesivas operaciones.

Cada clave tiene también una **"huella digital"**. Esta huella es un número de 20 bytes que se utilizará a la hora de validar y firmar las claves.

El identificador y la huella digital están muy relacionados ya que el identificador de la clave son los **cuatro últimos** bytes de la huella digital. Sería muy difícil que dos claves tuvieran el mismo identificador y estuviéramos trabajando con ambas.

Para averiguar el identificador de nuestra clave podemos usar la orden:

```
$ gpg --list-keys
```

El identificador es la secuencia de ocho dígitos hexadecimales que aparece en la línea que comienza con la palabra pub, por ejemplo:

```
/home/user/.gnupg/pubring.gpg
------------------------------
pub   2048R/E99DCB45 2010-08-30 [caduca: 2011-08-30]
uid                  Juan Perez <jperez@nodomain.org>
sub   2048R/8CB35894 2010-08-30 [caduca: 2011-08-30]
```

El identificador de esta clave es: E99DCB45.

Para averiguar su huella digital, tenemos:

```
$ gpg –fingerprint
```

Esta orden nos da más información que la anterior y con ella se puede comprobar la relación entre identificador y huella digital mencionada arriba.

Podemos utilizar el identificador de la clave para sucesivas operaciones como:

Generación del certificado de revocación

Es importante disponer de este **certificado** que usaremos cuando nuestra clave sea comprometida, la hayamos perdido o no la queramos usar más. Para ello ejecutaremos la siguiente orden:

```
$ gpg --gen-revoke identificador_de_la_clave > miclave-revoke.asc
```

Se nos preguntará la **razón** por la cual generamos el certificado, podemos elegir la 0 y escribir algo como "Certificado generado para mi seguridad".

Necesitaremos teclear la contraseña de nuestra clave (recordar que cualquier operación con nuestra clave requiere esta contraseña).

El certificado será almacenado en el fichero `miclave-revoke.asc`. Es muy recomendable **conservar este fichero** en un lugar alejado de las claves (directorio `.gnupg`). Normalmente se graba en un CD y se guarda en lugar seguro.

Identificadores de usuario

Un identificador de usuario consiste en un **nombre de usuario, una dirección de correo y un comentario**. En el momento de generar nuestras claves, suministramos esta información, por lo que nuestras claves en este momento ya tienen al menos un identificador de usuario.

Es más, podemos **usar este identificador** para hacer referencia a nuestras claves en las operaciones que realicemos. La orden para la generación del certificado de seguridad que hemos usado anteriormente podría ser también:

```
$ gpg --gen-revoke email_de_la_clave > miclave-revoke.asc
```

Ya que tenemos un identificador de usuario asociado a nuestra clave, podemos acceder a ella con nuestra dirección de correo electrónico, e incluso nuestro nombre, lo que nos resulte más cómodo.

Lo normal hoy en día es **tener más de una dirección** de correo electrónico personal, y es probable que queramos usar nuestra clave para enviar mensajes firmados/cifrados desde otras direcciones de correo diferentes a la que hemos usado para generar nuestra clave.

Si queremos **asociar** más direcciones de correo (o mejor dicho identificadores de usuario) a nuestra clave, podemos usar la orden:

```
$ gpg --edit-key identificador_de_la_clave
```

En este momento nos aparecerá una **línea de comandos** que comienza con el indicador `gpg>` esperando a que introduzcamos una orden para editar la clave. Usaremos `adduid` para **añadir** un identificador de usuario. Para hacernos una idea de las ódenes que se pueden usar desde esta línea, podemos teclear `help`.

Se nos pedirá un nombre, una dirección de correo electrónico y un comentario, al igual que ocurrió cuando **generamos nuestras claves** (pasos 4, 5 y 6).

Podemos añadir todos los identificadores de usuario que queramos usando el **comentario** para informar de la **actividad** que realizamos cuando usamos esa cuenta de correo electrónico.

Exportar nuestra clave

Exportar la clave privada no es buena idea salvo que nos la vayamos a llevar a otro sitio o sepamos muy bien lo que estamos haciendo. En cambio, exportar la clave pública es una forma para **hacer llegar** esta clave a las personas con las que nos vayamos a comunicar usando criptografía asimétrica.

Para exportar nuestra clave pública a un fichero:

```
$ gpg -a --export identificador_de_la_clave > clave-publica.asc
```

Publicar nuestra clave pública

Otra forma para distribuir nuestra clave que se usa con bastante frecuencia es la **publicación** en un **servidor de claves**, de esta forma únicamente tendremos que facilitar el identificador de nuestra clave y el servidor en la que la hemos publicado y de esta forma cualquier persona que quiera usar PGP puede acceder a nuestra clave (pública).

Hay muchos servidores de claves repartidos por Internet. Algunos de ellos son:

```
pgp.rediris.es
pgp.mit.edu
subkeys.pgp.net
pgp.webtru.st
```

Para **publicar** nuestra clave ejecutaremos:

```
$ gpg –keyserver servidor-de-claves –send-key identificador_de_la_clave
```

Normalmente los servidores de claves están sincronizados, por lo que al publicar nuestra clave en uno de ellos, ésta será publicada en el resto.

Ahora es el momento de publicitar su identificador, huella digital y servidor donde la hemos publicado para que nuestros interlocutores puedan importarla y así poderse comunicar con nosotros.

Para publicitarla podemos usar nuestra página web personal, correo electrónico u otro medio que consideremos oportuno.

Importar la clave pública de otra persona

En este momento estamos preparados para **importar** las claves de otras personas. Cuando importamos otras claves, éstas se almacenan en lo que se llama nuestro **anillo de claves**.

Podemos recibir estas claves por correo electrónico, descargarlas de una página web o importarlas de un servidor de claves (lógicamente para poder importarlas de un servidor de claves, el propietario de la clave tendrá que haberla publicado en el servidor como hemos hecho nosotros).

Para importarla desde el **fichero** `clave-publica-juan.asc` usaremos:

```
$ gpg –import clave-publica-juan.asc
```

Para importarla de un **servidor de claves**:

```
$ gpg –keyserver servidor-de-claves –recv-key identificador_de_la_clave
```

El identificador de la clave nos lo tendrá que facilitar el propietario de la misma.

Importante: Cuando exportemos o importemos una clave de un servidor de claves, debemos de

tener en cuenta que se usa el **protocolo hkp** (puerto 11371) y por lo tanto debemos de asegurarnos de que nuestro **cortafuegos** permite el tráfico por este puerto.

Firmado de claves

La clave pública de otra persona ya está en nuestro anillo de claves, pero ¿es esta persona quien dice ser?. Si una **Autoridad Certificadora (CA)** de confianza, u otra persona en la que confiáramos hubiera **firmado** esa clave podríamos estar seguros de que la persona se ha tenido que **identificar** para validar sus datos.

Existen Autoridades Certificadoras conocidas y reconocidas (como Thawte, Verisign, etc), que son comerciales y seguramente deberemos pagar para que firmen nuestra clave. Existen por el contrario otras Autoridades Certificadoras que se basan en Anillos de Confianza en las que los usuarios registrados son los avales de que las claves pertenecen a quién dicen pertenecer, por ejemplo Cacert [6].

Para **garantizar** que el propietario de la clave es quién dice ser, sin tener que acudir a una CA comercial, podemos:

1. Recurrir a Autoridades Certificadoras sin ánimo de lucro como CAcert.
2. Reunirnos con el propietario de la clave y pedirle que se identifique.
3. Acudir a una fiesta de firmado de claves.
4. Usar círculos de confianza (Web of Trust).

Una vez comprobada la identidad de la persona, podemos proceder a firmar su clave:

```
$ gpg –sign-key identificador_de_la_clave_a_firmar
```

Debemos estar seguros de la identidad de las personas antes de firmar sus claves.

Fiestas de firmado de claves

Estas reuniones son muy interesantes ya que podemos comprobar la identidad de las personas que acuden y así firmar sus claves y que ellos firmen la nuestra.

Es conveniente informarse de lo que hay que hacer o llevar a una fiesta de claves antes de acudir a ella, puesto que hay muchas forma de organizarlas.

Dependiendo cómo se organice la fiesta, en general tendremos que llevar con nosotros:

a) Muchas **copias en papel** indicando en cada una el ID de nuestra clave con su huella digital impresa y el servidor de claves en la que está publicado. Llevaremos tantas copias como personas queramos que firmen nuestra clave.

b) Llevar la **documentación** necesaria para acreditar nuestra identidad.

c) Llevar un **bolígrafo o lápiz** para tomar notas.

Normalmente no se permite el uso de ordenadores y otros dispositivos en las fiestas de firmado de

claves. Es probable que el anfitrión de la fiesta nos pida que **publiquemos** nuestra clave en algún anillo de claves para facilitar su tarea en sitios como Biglumber [7].

En estas reuniones **acreditaremos** nuestra identidad a las personas que nos lo soliciten y aprovecharemos para comprobar la suya y de esta forma poder firmar el mayor número de claves posible.

El **grupo nibbler** [8] de organiza todos los años desde 2010, una fiesta de firmado de claves abierta a cualquier persona que desee participar.

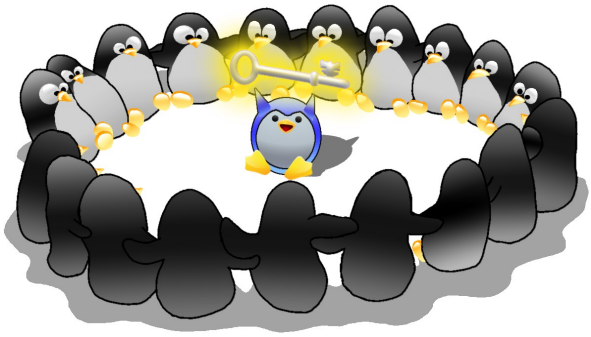
Círculos de confianza

Cuando **firmamos** una clave **garantizamos** que esa clave pertenece a la persona correcta. De esta forma nos convertimos en una pequeñísima entidad de confianza. Cuando una persona u organización firma muchas claves y hay otras muchas personas que han firmado la suya, se establece un **círculo de confianza** entre las personas que la forman, así la credibilidad de la círculo está garantizada por el número de personas implicadas.

Si una persona que pertenece a uno de estos círculos firma nuestra clave, nuestra identidad estará respaldada por el círculo de confianza a la que pertenece esa persona.

Es común establecer círculos de confianza en proyectos de conocimiento libre, en la que los desarrolladores y usuarios forman una Autoridad Certificadora.

Si queremos formar parte de un círculo de confianza deberemos volver a importar nuestra clave al servidor de claves cada vez que firmemos alguna clave, de esta forma cuando una persona importe nuestra clave tendrá actualizadas las firmas que hemos realizado.



Firmar y verificar mensajes

Llegó el momento de establecer la **primera comunicación**. Prácticamente todas las herramientas de correo electrónico tienen una extensión para criptografía PGP, por lo que la labor de firmar un mensaje se realiza desde la propia herramienta de correo.

Si no podemos firmar desde nuestra herramienta de correo, podemos usar el método **manual**, que consiste en obtener la firma del mensaje mediante la orden.

```
$ gpg -a -o fichero_con_firma –detach-sign fichero_con_mensaje
```

Y enviar ambos ficheros **adjuntos** al destinatario. Si el destinatario tiene nuestra clave, podrá **comprobar** la firma usando la siguiente orden.

```
$ gpg –verify fichero_con_firma fichero_con_mensaje
```

Cifrar y descifrar mensajes

De igual modo que para el firmado de mensajes, nuestra herramienta de correo tendrá opciones para cifrar y descifrar mensajes.

Si queremos hacerlo **paso por paso**, usaremos

```
$ gpg -a -o fichero_cifrado -e -r id_del_destinatario fichero_a_cifrar
```

Para **cifrar** el mensaje con la clave pública del destinatario. Esta clave deberá estar importada en nuestro anillo de claves.

Nuestro interlocutor podrá **descifrar** el mensaje usando su clave privada y la orden:

```
$ gpg -o fichero_en_claro –d fichero_cifrado
```

Criptografía simétrica

Es posible utilizar GnuPG para cifrar de forma **simétrica**. Aunque no es recomendable usar este tipo de criptografía, se puede usar en situaciones en los que no sea necesario **transmitir** la clave de cifrado, por ejemplo para cifrar contenido que sólo nosotros vamos a leer.

Para cifrar un fichero con una contraseña usaremos:

```
$ gpg -o fichero_cifrado -c fichero
```

Conclusiones

La criptografía asimétrica nos da una solución real a los problemas de cifrado de la información y certificación de identidad. Su uso hoy en día está muy extendido, desde comunicaciones entre empresas y particulares hasta su presencia en el nuevo DNI electrónico.

Es conveniente tener una idea aunque no sea detallada de cómo funcionan estos sistemas en los cuales ponemos nuestra información privada.

Para aquéllos que quieran utilizar GnuPG en sus comunicaciones, este artículo les ha abierto la puerta a esta herramienta que en muchas ocasiones ya está integrada en los sistemas operativos o en las herramientas de correo electrónico.

Para los más curiosos se han indicado en las órdenes más comunes de GnuPG.

Es muy recomendable usar herramientas criptográficas en nuestras comunicaciones para evitar la gran multitud de ataques se producen contra la integridad de nuestras comunicaciones.

Ningún proyecto de software libre tendría futuro sin este tipo de criptografía que garantiza la identidad de los desarrolladores para tranquilidad de los usuarios.

Quedan muchas opciones de GnuPG por mostrar, para más información, consultar [9] y [10].

Enlaces

[1] http://www.pgp.com/ Página oficial de PGP Corporation.
[2] http://www.openpgp.org/ Página oficial de la alianza OpenPGP.
[3] http://www.gnupg.org/ Página oficial de GnuPG.
[4] http://gnupg.org/related_software/frontends.es.html Interfaces de usuario de GnuPG.
[5] http://biblioweb.sindominio.net/telematica/command_es/ En el principio... fue la línea de comandos.
[6] http://cacert.org CAcert.
[7] http://biglumber.com Anillos de claves en Biglumber.
[8] http://nibbler.es Grupo nibbler.
[9] http://www.gnupg.org/gph/es/manual.html Guía de GNU Privacy Guard.
[10] http://www.dewinter.com/gnupg_howto/spanish/index.html Manual Cómo de GnuPG.

José María Alonso Josa
Profesor, autor y desarrollador
chema@nibbler.es
Agosto 2010
Actualizado en Marzo 2012