PGP y GnuPG
Teoría y práctica.

Francisco Lopera Morlas
<impulsor@nisu.org>

31 de diciembre de 2001

Resumen

Las necesidades de protección de la información se han incrementado en gran medida con la utilización de los ordenadores y las redes de comunicaciones. La criptografía proporciona herramientas que permiten: proteger un documento para que nadie pueda entender su contenido, comprobar que un programa obtenido por FTP es el original o garantizar que solo el usuario receptor de un mensaje de correo electrónico pueda leerlo o interpretarlo. PGP y GnuPG son herramientas disponibles para realizar estas acciones. Trataremos de explicar cuales fueron los orígenes de este programa, su evolución, sus curiosas licencias y su funcionamiento.

Índice General

1 PGP. 3
1.1 Introducción. . . . 3
1.2 Conceptos. . . . 3
1.2.1 Claves asimétricas y claves simétricas. . . . 3
1.2.2 Anillos de confianza. . . . 4
1.2.3 Certificación. . . . 4
1.3 Funcionalidad de PGP. . . . 5
1.3.1 Codificación. . . . 5
1.3.2 Firmas digitales. . . . 5
1.3.3 Armaduras ASCII. . . . 6
1.3.4 Huella digital. . . . 7
1.4 Vulnerabilidades de PGP. . . . 7
1.5 Versiones. . . . 8
1.6 Principales comandos de PGP. . . . 9

2 OpenPGP. 11

3 GnuPG. 11
3.1 Introducción. . . . 11
3.2 Funcionalidad de GnuPG. . . . 12
3.3 Consejos para la generación de claves. . . . 12
3.4 Principales comandos de GnuPG. . . . 13

4 Opinión personal. 14

1 PGP.

1.1 Introducción.

PGP significa *Pretty Good Privacy* (Privacidad bastante buena) y su desarrollo comenzó a principios de los 90 de la mano de Phill Zimmerman. Por aquél entonces no existían herramientas potentes y baratas de criptografía seria y su autor quiso completar ese hueco.

Phill Zimmerman trabajaba para el Departamento de Defensa de los EEUU. Aunque no era criptólogo ni matemático, su trabajo estaba muy relacionado con la seguridad informática y se interesó por la criptología. Al dejar de trabajar para el gobierno fue cuando decidió crear un sistema de seguridad para el correo electrónico.
Básicamente siguió unos principios fundamentes para su creación:

1. Usar los mejores algoritmos del momento libres de trabas del gobierno.
2. Integrar estos algoritmos en PGP de forma que funcione en cualquier máquina y sobre cualquier S.O.
3. Instalar en Internet la aplicación PGP, el código fuente y la documentación.
4. Fundar PGP Inc. y ofrecer servicios a precios muy asequibles.

Pretendía acercar la criptografía al usuario medio. La repuesta del gobierno no se hizo esperar y Phill fue denunciado por el FBI acusándole de proporcionar herramientas de comunicación seguras a delincuentes[1]. También fue denunciado por utilizar el algoritmo RSA. Phill ganó ambos juicios, si bien tuvo que ceder en algunas imposiciones legales en cuanto al RSA.

Con el paso del tiempo, PGP se ha convertido en un estándar[2] usado en el correo electrónico, usado en grandes empresas y sobre todo, por los pequeños usuarios. Los motivos son muy conocidos: hasta principios del 2001 la distribución de PGP permitió su uso gratuito para fines no comerciales y publicó su código fuente, ganándose la confianza de los usuarios. Cuando Pill abandonó la empresa en febrero del 2001, el código fuente se dejó de publicar.

PGP es un producto estándar internacional y no solo tiene aplicación en correo electrónico, se pueden codificar particiones enteras de disco duro (PGPDISK) y codificación automática de toda la red TCP/IP (PGPNET).

1.2 Conceptos.

1.2.1 Claves asimétricas y claves simétricas.

PGP trabaja con criptografía *asimétrica* y *simétrica*. Para poder entender mejor el sistema de codificación usado por los sistemas de claves *asimetrícas*, veamos cuales son sus diferencias respecto a los sistemas de claves *simétricas*.

[1] La frase que empleó fué genial: ***Si la privacidad está fuera de la ley, sólo los delicuentes gozaran de la privacidad.***

[2] Apoyado en el RFC 1991 del IETF

1. Los sistemas de cifrado con clave *simétrica* son aquellos en los que la clave se usa para cifrar datos es la misma que se usa para descifrar. Es decir, en el caso del correo electrónico el remitente cifraría con una clave secreta y el receptor deberá descifrar con la misma clave secreta. Se ve con claridad que el mayor inconveniente es pasar esa clave de un modo seguro al receptor. No obstante, los algoritmos simétricos son mucho más rápidos que los asimétricos.

2. Los sistemas de cifrado con claves *asimétricas* usan claves diferentes para cifrar y posteriormente descifrar los datos. El remitente cifra con la clave pública del receptor y el receptor descifra usando su propia clave privada. Para que el sistema tenga algún tipo de validez, nadie debe conocer su clave privada mientras que la clave pública puede ser libremente distribuida. Si el sistema está bien implementado, la clave privada no debe derivar nunca de la pública.

PGP puede generar un par de claves pública y privada. La clave pública deberá distribuirse (existen diversos métodos, como más adelante se verá) mientras que la clave privada se guarda en un lugar seguro y accesible solo a su dueño.

1.2.2 Anillos de confianza.

Cada clave se almacena en unas estructuras llamadas anillos. Cada usuario tendrá dos anillos, cada una en un fichero, uno para claves públicas (PUBRING.PKR) y otra para claves privadas (SECRING.SKR).

Uno de los puntos flacos en los algoritmos de clave asimétrica es la transmisión del código público. Se puede poner en circulación código con un identificador de usuario (UID) falso. Si se codifican los mensajes con este código, el usuario falso puede descodificarlos y leerlos.

La solución que ofrece PGP es la siguiente: la clave pública de un usuario puede estar firmada con las claves de otros usuarios, con objetivo de reconocer que el UID de la clave pertenece al usuario a quien dice pertenecer. A partir de ahí, cada usuario tendrá que decidir hasta que punto se podrá fiar de la firma (dependiendo de si se sabe con seguridad de que dicha clave le pertenece). En definitiva, se basa un poco en el tópico *"To el mundo e bueno"*.

Para identificar que una clave es cierta se puede preguntar por un medio seguro la huella digital de esa clave (ver 1.3.4).

1.2.3 Certificación.

El problema anterior se puede solventar empleando a terceros (autoridades certificadoras) cuya función es extender certificados de claves firmados con su propia clave pública. PGP no delega responsabilidad de certificación a autoridades de certificación externa, sino que es el mismo usuario el que según su criterio, se encarga de la certificación. Existen servidores de claves públicas cuyo objetivo es distribuir las claves públicas, pero sin ningún tipo de responsabilidad de certificación. Un ejemplo se puede encontrar en *pgp.rediris.es* `http://www.rediris.es/cert/servicios/keyserver/`

1.3 Funcionalidad de PGP.

Como se ha dicho, el principal cometido de PGP es ser un método criptográfico fiable para su uso con el correo electrónico. Podemos decir que en general, ofrece tres tipos de servicios:

1. Privacidad: PGP permite cifrar ficheros utilizando cifrado asimétrico. También permite opcionalmente el cifrado simétrico.
2. Autentificación: al firmar con una clave privada podemos estar seguros de que es posible verificar quien es el autor.
3. Integridad: a través de la firma digital se puede comprobar que fichero o mensaje ha sido modificado.

1.3.1 Codificación.

PGP cifra primero el mensaje empleando un algoritmo simétrico (ver 1.2.1) con una clave de sesión (generada aleatoriamente) y después codifica la clave empleando la llave pública del receptor del mensaje. Esta clave se obtiene del anillo de claves públicas usando el identificador del usuario al que queremos mandar el mensaje.

Cuando se trata de decodificar el mensaje, PGP busca la cabecera de las claves públicas con las que está codificado y nos pide la contraseña de acceso a la clave privada. Si la contraseña es correcta, descifrará el mensaje. La contraseña es una medida de seguridad adicional. En caso de que nuestro anillo de claves privadas fuera accesible para un intruso, este aún tendría que averiguar la contraseña.
PGP depende de la calidad del generador de números aleatorios para calcular las claves, por que si alguien predice la secuencia de claves que estamos usando, podrá descifrar todos nuestros mensajes. PGP usa para evitar un generador de números pseudoaleatorios muy seguro.

Para garantizar la compatibilidad se emplea un sistema para mapear los símbolos de 8 bits en palabras de 7 bits. Esto es necesario, ya que existen gestores de correo que solo entienden caracteres de 7 bits. El sistema utilizado por PGP es el algoritmo es el RADIX-64 que incrementa el tamaño del mensaje en un 33 %. Para reducirlo, PGP emplea el algoritmo de compresión ZIP.

Otra restricción propia del correo electrónico es la cantidad de caracteres por mensaje. PGP permite segmentación del mensaje al enviar y reemsamblado al llegar.

1.3.2 Firmas digitales.

La firma digital se basa en la verificación de la autoría de un mensaje, es decir, que debe poderse comprobar que el supuesto remitente es quien dice ser. Básicamente, el remitente firma con su clave privada y el receptor podrá comprobar su autenticidad verificándolo con la clave pública del remitente.

El método es útil para reducir riesgos de seguridad en un sistema (envío de programas troyanos), para asegurar la fiabilidad de un envío de datos. Es relativamente fácil falsificar el nombre y la dirección del remitente, pero imposible[3] falsificar una firma digital. Hay que destacar que

[3] imposible es la palabra empleada en la referencia.

una posible manipulación posterior del mensaje (después del cifrado) también "cantaría" en la verificación.

La firma digital es un bloque de caracteres de longitud fija que depende tanto del contenido del mensaje como del remitente del mismo. Dicho bloque se genera del siguiente modo:

1. Primero se aplica un algoritmo *hash* (como el SHA1) que transforma el mensaje en una especie de resumen de tamaño fijo independientemente del tamaño del fichero. El algoritmo utilizado para esto presenta la propiedad de que pequeñas variaciones en el mensaje original producen grandes variaciones en el mensaje, lo que es un añadido.

2. En segundo lugar, el resumen obtenido se cifra mediante un sistema de cifrado asimétrico (ver 1.2.1) utilizando la clave privada del remitente. En este proceso nos pide una clave de acceso, como siempre que se debe acceder a la clave privada para algo.

El proceso de verificación de la firma consta de tres partes:

1. El resumen enviado se descifra con la clave pública del emisor.

2. Se realiza un resumen nuevo del mensaje recibido.

3. Se comprueba si ambos resúmenes coinciden. Si es así la firma es válida y si no, inválida.

1.3.3 Armaduras ASCII.

Una de las características más atractivas de PGP es la capacidad de redirigir sus códigos de salida en formato ASCII. Todas las salidas de PGP son secuencias binarias (mensajes codificados, binarios codificados). Sin embargo, en determinados casos como el correo electrónico puede ser útil enviar la información en modo texto.

Por ejemplo:
—–BEGIN PGP PUBLIC KEY BLOCK—–
Version: GnuPG v1.0.6 (GNU/Linux)
Comment: For info see http://www.gnupg.org

mQGiBDpMnV0RBACxww3UW2/Hae6znn7XB1w1QFovJs4xpsVVOK3Yaq7zvV9IjIwW
8fj8XZTMfaGmmIVKFiPKlwnfcQFERveXrniSdEkneakuUBNlihsz+6RBXPIg5i3d
+fImp7uZ9Lurzp2+G0dg/qqqZNg/4+tXwUtRGoa7x9X1A1aDB9MmzEi5WwCgjH48
v49d6Z3rlfG5U5KtuOWz5y0D/R+IuH3kXe+KHbPY1Ld5zcZKyccPvd5tQn/cObJz
ikxcD/7OPEAkMbLQkS1jXWPOj3L6PHJoL2ZMJBqZkpawUukuYhivkR70ZT5TvtfS
bJBaDgscRRpHDMgdw7wcKZVFFo/tszoWa3OT7ZSzOzuEAe9SK45xs9OwmL5RZg8P
X4I8A/9vUP91U5/laFR0aO/3+R0mAFiN/zbfzjXXr4ljY5jC5vd+FJ9V4d8WPOtr
bew8xtmsU3s6YgYw36RbaD4+ObukQmpjNneU3n7W5vnU+aH6zuBs0iyLj9xAl6II
4uwYttNrrcgKwi3x43EU0C3SaWQ96yvhNKXhl/jxYbh7UongCLQrRnJhbmNpc2Nv
IExvcGVyYSBNb3JsYXMgPHBjX2xvcGVyYUBvbm8uY29tPohWBBMRAgAWBQI6TJ1d
BAsKBAMDFQMCAxYCAQIXgAAKCRAF6sETJpAjs6/DAJ0UDjJRlEZQbP90M77f7vn5

1JuOGwCfTQtoJOvflmsbPyKwfNiupZooILS5AQ0EOkydhBAEALA66wqH2GKzMg/u
joTuN2qvI7CdkPzXmuMbHoHf53RsnKXRe1GOLh+ShOc7eQevShzB+swzEwazNT5m
/60PjKpHN4oXMTn8y9nK1lpECXNq+1fJIj7p72TOizyuwvG0x9aqYk3avcHwbfUY
kjz5O/ZqfOAgPDYDy6EE0K/M0EZ/AAQNBACENrUV0s56npSd1nG2lzZI0pmQBZUb
d9o9VgV9FHlELp5V2srANN0BS/KaZUSwFELuHkj6ibRK285y+nKsiHeSl/iUt4i2
vbAf0ZDS9IkjelXJs7J/WOCwB8v5jJCzp6deO+TyC794guOlgVzPMqfuPq1iW9uD
+tJ+deb2hZPnbIhGBBgRAgAGBQI6TJ2EAAoJEAXqwRMmkCOzBY4AnRc/0/DOMxVM
X5U1DtxqEn1JN1MeAKCIeLvBA8dblL54qMsU1LThjrz4dg==
=k4ER
——END PGP PUBLIC KEY BLOCK——

Esto permite una fácil distribución de la clave privada o incluso que un mensaje pudiera copiarse a mano y después cifrar por medios informáticos. Para generar una armadura ASCII no se emplean los 128 caracteres disponibles del código ASCII, algo peligroso si tenemos en cuenta que algunos de estos caracteres son retornos de carro o tabuladores, de modo que se emplean solo 64 caracteres realmente imprimibles.

1.3.4 Huella digital.

La forma más común de averiguar que una clave pertenece a un usuario es preguntarle su huella digital. Cada clave, aparte de la secuencia binaria correspondiente al algoritmo que emplee, tiene unos datos: el UID (ver 1.2.2), fecha de expiración, versión de PGP con que se creó y por último una secuencia hexadecimal lo bastante larga para ser única y lo bastante corta como para ser escrito en un papel con facilidad. Para confirmar la autenticidad de una clave, solo hay que preguntarle a su autor su huella digital y comprobar si coincide. Un ejemplo de huella digital y otros datos pueden ser:

pub 1024D/269023B3 2000-12-29 Francisco Lopera Morlas <pc_lopera@ono.com>
Key fingerprint = 5B43 7385 90E9 E446 9522 3140 05EA C113 2690 23B3

1.4 Vulnerabilidades de PGP.

Como cualquier herramienta, PGP proporciona una gran rendimiento si se emplea correctamente, pero un uso inadecuado lo puede convertir en una protección totalmente inútil. Es necesario, pues, adquirir ciertas costumbres que harán de PGP una herramienta útil.

- *Escoger unas contraseñas adecuadas*: “Paco” y “Pepe” no suelen ser una buena opción.
- *Proteger adecuadamente los archivos “sensibles”*: es decir, los ficheros que contienen los anillos de claves y el fichero que alberga la semilla aleatoria[4]. Esta protección debe llevarse a cabo de cara a posibles curiosos o a pérdidas de datos. Hay que tener en cuenta que si

[4]El método de generación de números aleatorios viene dado por un fichero (semilla) llamado RANDSEED.BIN. Este fichero es sensible y debemos evitar que quede expuesto.

formateamos el "Windows" nos cargamos los anillos y no podremos descifrar nunca jamás un mensaje cifrado para nuestras claves.

- *Emitir revocaciones de nuestras claves al generarlas y guardarlas en un lugar seguro*: es el único mecanismo válido para revocar una clave en caso de pérdida del anillo privado. PGP 6 permite nombrar revocadores para nuestras claves, de forma que estos pueden invalidarla sin necesidad de usar la clave privada.

- *Firmar sólo las claves de cuya autenticidad estemos seguros*: es de cajón. Es la única manera de que los anillos de confianza puedan funcionar. Si todos firmamos las claves al estilo "To er mundo e bueno" podríamos certificar claves falsas.

Como todo en informática, PGP también tiene fallos detectados. Clasificaremos según dos criterios:

1. *Debido a la implementación*: agujeros de seguridad provocados por una implementación defectuosa de PGP correspondientes a versiones concretas del programa. Por ejemplo, la versión 5.0 de UNIX hacía que las claves no fuesen completamente aleatorias, o en las versiones windows (5.0-7.0.4), en las que un procesamiento inadecuado de las armaduras ASCII permitía a un atacante introducir ficheros en la computadora de la víctima.

2. *Debido al protocolo*: En este apartado habría que reseñar aquellos agujeros de seguridad que son inherentes a la definición del estándar Open PGP. Existe una técnica que permite a un atacante falsificar firmas digitales, si bien el atacante debe tener acceso físico a la computadora de la víctima, ya es un fallo de protocolo.

1.5 Versiones.

Al ser PGP un producto desarrollado en EEUU esta sujeto a ciertas leyes sobre exportación de programas que incluyen código criptográfico; por esta razón existe una versión internacional para casi todas las versiones de PGP que se distinguen de las demás por acabar en la letra "i". EEUU entiende la criptografía como un arma, de manera que "capan" la versión internacional de cara a su versión oficial; un ejemplo de esto es la versión 2.6.3 de su PGP, cuya versión internacional no tiene licencia para utilizar el criptosistema RSA, mientras que la versión USA si.

Para entendernos (y ser legales) comentaremos las versiones internacionales (casi todas disponibles en `http://www.pgpi.com`):

- *PGP 1.0*: apareció en 1991 y estaba solo disponible para la plataforma msdos. El sistema de cifrado estaba basado en el criptosistema RSA y es incompatible con las versiones posteriores.

- *PGP 2.6.3i*: es probablemente la versión más popular de PGP hasta la actualidad. Se ofreció de manera gratuita para uso no comercial. Muchas aplicaciones actuales aún están basadas en esa versión por lo que probablemente sea necesario mantener (todavía) una copia por si se necesita. PGP2 genera claves con el algoritmo RSA y como algoritmo de cifrado usa IDEA.

- *PGP 5.0i*: es la versión gratuita internacional del *PGP 5.0 for personal privacy* adaptada para su uso personal. Fue la primera que introdujo claves Diffie-Hellman y ofrece una gran mejora en cuanto a la velocidad. Ésta es la primera versión que incorpora una interfaz gráfica para las plataformas Windows 95/NT y Mac. La versión original americana no permite generar claves RSA (aunque si utilizarlas) ni permite el cifrado convencional utilizando *claves simétricas* (ver 1.2.1), sin embargo, la versión internacional ha sido modificada para permitir estas dos opciones. Las claves generadas por PGP5 son del tipo DSS/DH(*Digital Signature Standard / Diffie-Helman*). Como algoritmos de cifrado emplea CAST, Triple-DES e IDEA.

- *PGP 5.5.3i*: funcionalmente equivalente a la versión de pago *PGP 5.5 for personal privacy*, pero sigue siendo completamente gratis. Incluye todas las funcionalidades desaparecidas en la versión de 5.0 como por ejemplo la generación de claves RSA y el cifrado convencional simétrico.

- *PGP 6.0i*: versión que tampoco permite crear claves RSA. Incorpora una serie de novedades como PGPDISK, SECURE VIEWER, partición de claves privadas, etc. Esta versión no está disponible para entornos UNIX.

1.6 Principales comandos de PGP.

A continuación se muestran como realizar las operaciones más frecuentes de manejo de claves, cifrado y firmado empleando PGP.

1. Generar un par de claves pública/privada:

   - PGP 2.6.3i: pgp -kg
   - PGP 5.0i: pgpk -g

2. Cifrar un mensaje de texto con una clave pública (generando un fichero cifrado binario de extensión .pgp):

   - PGP 2.6.3i: pgp -e ejemplo.txt
   - PGP 5.0i: pgpe -r nombre_usuario ejemplo.txt

3. Cifrar un mensaje de texto con una clave pública (generando un fichero cifrado ASCII, para enviar por correo de extensión asc):

   - PGP 2.6.3i: pgp -ea ejemplo.txt
   - PGP 5.0i: pgpe -ar nombre_usuario ejemplo.txt

4. Descifrar un mensaje con la clave privada (generando un fichero de texto llamado desencriptado.txt):

- PGP 2.6.3i: pgp ejemplo.txt.pgp -o desencriptado.txt
- PGP 5.0i: pgpv -o desencriptado.txt ejemplo.txt.pgp

5. Firmar un mensaje con la clave privada (generando un fichero de firma binario comprimido de extensión .pgp):

   - PGP 2.6.3i: pgp -s ejemplo.txt
   - PGP 5.0i: pgps ejemplo.txt

6. Firmar un mensaje con la clave privada (generando un de firma ASCII comprimido de extensión .asc):

   - PGP 2.6.3i: pgp -sa ejemplo.txt
   - PGP 5.0i: pgps -a ejemplo.txt

7. Firmar un mensaje con la clave privada (generando un de firma ASCII legible de extensión .asc):

   - PGP 2.6.3i: pgp -sat ejemplo.txt
   - PGP 5.0i: pgpk -at ejemplo.txt

8. Verificar la firma de un mensaje con una clave pública (generando un fichero de texto llamado desencriptado.txt):

   - PGP 2.6.3i: pgp ejemplo.txt.pgp -o desencriptado.txt
   - PGP 5.0i: pgpv ejemplo.txt.pgp -o desencriptado.txt

9. Cifrar utilizando cifrado simétrico (convencional) (generando un fichero cifrado binario de extensión .pgp):

   - PGP 2.6.3i: pgp -c ejemplo.txt
   - PGP 5.0i: pgpe -c ejemplo.txt

10. Descifrar utilizando cifrado simétrico (convencional):

    - PGP 2.6.3i: pgp ejemplo.txt.pgp
    - PGP 5.0i: pgpv ejemplo.txt.pgp

11. Exportar una clave pública, para enviarla a otro usuario:

    - PGP 2.6.3i: pgp -kxa nombre_usuario fichero_clave_usuario
    - PGP 5.0i: pgpk -xa nombre_usuario -o fichero_clave_usuario

12. Importar una clave pública:

    - PGP 2.6.3i: pgp -ka fichero_clave_usuario
    - PGP 5.0i: pgpk -a fichero_clave_usuario

13. Listar las claves públicas conocidas:

    - PGP 2.6.3i: pgp -kv
    - PGP 5.0i: pgpk -l

2 OpenPGP.

OPENPGP es una actualización del estandar PGP orientado a de PGP/MIME[5], es decir, a la estructura de correo MIME.
Ofrece mejoras como:

- Nuevo formato de mensaje.
- Introducción de MIME.
- Adaptación a autoridades certificadoras.
- Otras mejoras generales.

3 GnuPG.

3.1 Introducción.

La rápida popularización de PGP entre ciertos sectores de la comunidad de Internet y el desarrollo de un estándar público llamado *Open PGP* ha permitido la creacción de variantes del programa original. Una mención especial recae sobre el proyecto *GnuPG (GNU Privacy Guard)* que funciona en una gran cantidad de plataformas.

GnuPG tienen las mismas funciones que PGP, pero evita usar algoritmos con patentes restrictivas. PGP puede ser usado libremente con fines personales, pero no comerciales y se desarrollo se volvió cerrado en sus últimas versiones. En cambio, *GnuPG* es de libre uso y de desarrollo abierto. *GnuPG* genera claves del tipo DSA/ElGamal *(Digital Signature Algorithm*, también conocido como *DSS)*, que es compatible con PGP excepto en los algoritmos con patentes restrictivas RSA e IDEA[6] .

[5]Definido en el RFC 2015

[6]He encontrado referencias de que la licencia de estos algoritmos caducó y ya esta implementado en los ultimas versiones de GnuPG.

3.2 Funcionalidad de GnuPG.

GnuPG es una alternativa a PGP que implementa toda su funcionalidad pero sin las restricciones que arrastraba puesto que no utiliza algoritmos registrados ni limitados para su exportación internacional.

Sus características más importantes son:

- Sigue el RFC 2440 (También conocido como *Open PGP*).
- Ha sido escrito partiendo de cero, bajo licencia *GNU.*
- Permite descifrar y verificar firmas PGP 5.x
- Añade nuevas funcionalidades y mejoras de seguridad con respecto a PGP.
- Soporta los siguientes algoritmos: El Gamal, DSA, triple DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 y TIGER.
- Permite integrar fácilmente nuevos algoritmos empleando módulos de extensión.

3.3 Consejos para la generación de claves.

1. Cuando generamos una clave empleando *GnuPG* la primera pregunta que se nos formulará es que algoritmo a usar. Lo más habitual es emplear DSA/ElGamal por no estar patentado.
2. La siguiente pregunta es la longitud de la clave. Aquí dependerá de la relarelaciónción calidad/tiempo que quiera el usuario. Cuanto mayor sea la clave, menor el riesgo de decodificación, pero más tiempo para los cálculos. El tamaño mínimo de GnuPG es de 768 bits y el máximo 2048. Para DSA 1024 es un tamaño fijo.
3. A continuación, se nos pedirá teclear el nombre, dirección de correo electrónico. El código se calculará de acuerdo a estas entradas. También se puede introducir un comentario.
4. Ahora hay que introducir una contraseña. Para que sea efectiva debe cumplir los siguientes requisitos:
   - Debe ser larga.
   - Debe combinar mayúsculas, minúsculas y números.
   - Debe contener caracteres alfanuméricos.
   - Debe ser difícil de adivinar: “Paco” y “Pepe” no.

En general, la contraseña debe ser lo bastante difícil para que no pueda ser traspasado por un ataque de fuerza bruta, pero lo bastante fácil para que la recordemos.

3.4 Principales comandos de GnuPG.

A continuación se muestran como realizar las operaciones más frecuentes de manejo de claves, cifrado y firmado empleando PGP.

1. Generar un par de claves pública/privada:
   - gpg –gen-key
2. Cifrar un mensaje de texto con una clave pública (generando un fichero cifrado binario de extensión .pgp):
   - gpg -e ejemplo.txt
3. Cifrar un mensaje de texto con una clave pública (generando un fichero cifrado ASCII, para enviar por correo de extensión asc):
   - gpg -ea ejemplo.txt
4. Descifrar un mensaje con la clave privada (generando un fichero de texto llamado desencriptado.txt):
   - gpg -o desencriptado.txt ejemplo.txt.gpg
5. Firmar un mensaje con la clave privada (generando un fichero de firma binario comprimido de extensión .pgp):
   - gpg -s ejemplo.txt
6. Firmar un mensaje con la clave privada (generando un de firma ASCII comprimido de extensión .asc):
   - gpg -sa ejemplo.txt
7. Firmar un mensaje con la clave privada (generando un de firma ASCII legible de extensión .asc):
   - gpg -sat ejemplo.txt
8. Verificar la firma de un mensaje con una clave pública (generando un fichero de texto llamado desencriptado.txt):
   - gpg –verify -o desencriptado.txt ejemplo.txt.gpg
9. Cifrar utilizando cifrado simétrico (convencional) (generando un fichero cifrado binario de extensión .gpg):

- gpg –symmetric ejemplo.txt

10. Descifrar utilizando cifrado simétrico (convencional):
    - gpg ejemplo.txt.gpg
11. Exportar una clave pública, para enviarla a otro usuario:
    - gpg –export -r nombre_usuario -o fichero_clave_usuario
12. Importar una clave pública:
    - gpg –import fichero_clave_usuario
13. Listar las claves públicas conocidas:
    - gpg –list-keys

4 Opinión personal.

PGP está muerto. PGP/MIME con GNUPG siguen siendo empleados por el pequeño usuario para cifrar sus mensajes. Si bien a la mayoría de los usuarios “caseros” se las trae al fresco que se conozcan los contenidos de sus mensajes, los pocos que quedan no tienen demasiado donde elegir a la hora de buscar un programa de cifrado bien soportado y estandarizado, optando por la opción GNUPG con el incentivo de emplear un algoritmo europeo y permitir MIME al seguir las especificaciones OPENPGP. Hasta que no se haga nada mejor y se extienda, PGP/GNUPG parece una buena elección.

Referencias

[1] *Criptografía y Seguridad en Computadores* - Manual J. Lucena López.

[2] *Seguridad en el Correo electrónico* - Manuel Pons Martorell.

[3] *GnuPG Mini Como* - Horacio (version castellana).

[4] *Mutt-i, GnuPG y PGP Como* - Andrés Seco y J. Horacio.

[5] `http://www.pgpi.com`

[6] `http://www.gnupg.org`