INSTRUCTIVO PARA MANEJAR EL PGP

(Versión 2.6.3i del PGP)

Luis G. del Valle

PGP (Pretty Good Privacy) es un programa desarrollado por Philip Zimmermann. Reelaborado para su uso internacional por Ståle Schumacher.

Los autores han difundido el programa gratuitamente. Lo cual no quiere decir que no tengan derechos de autor. Esta distribución gratuita es para usos no comerciales. Ver el capítulo "legal Issues" en pgpdoc1.txt y en pgpdoc2.txt que forman parte del paquete de distribución del PGP.

En el mismo espíritu de los autores la distribución de este instructivo es también gratuita. Si se cita, favor de señalar la fuente.

Quien tenga mi llave pública podrá comprobar que no se la ha hecho ningún cambio a mi texto si tiene este documento como archivo en word para windows 6.0 y tiene también el archivo de firma: pgpinstr.sig. Ambos archivos, compactados están en el archivo pgpinstr.zip.

Cualquier observación respecto al instructivo es bien recibida. Escribir a: lvalle@mixcoac.uia.mx

México, D.F. octubre de 1996
(Revisión: marzo 1998)

PGPINSTR.DOC. Grabado el 05/03/98 10:15 AM

INSTRUCTIVO PARA MANEJAR EL PGP

(Pretty Good Privacy)

Luis G. del Valle

INSTRUCTIVO PARA MANEJAR EL PGP

Si vas a usar el programa es porque estás convencido de su utilidad. Algunos de sus usos: guardar en la propia computadora archivos en clave de alta seguridad. Transmitir por correo electrónico o por otros medios (en disquetes) archivos cifrados, de manera que sólo los pueda descifrar aquél o aquellos para quienes los hayas cifrado: cartas confidenciales, información reservada,

I. Cargar debidamente el programa

A. Obtener el programa

El archivo se llama pgp263i.zip. Se puede obtener de alguien que lo tenga en disquete, o bien por internet. El programa, información y auxiliares:

Visitar: http://www.pgpi.com/ Es la página internacional del PGP. Allí se puede bajar el programa o encontrar lugares para ftp y para recibirlo por correo electrónico.

B. Crear el directorio: c:\pgp

C. Descompactar pgp263i.zip

Descompactar pgp263i.zip con pkunzip de modo que los archivos caigan en c:\pgp. Al descomprimirlo con pkunzip resultan algunos archivos, entre ellos otro zip (pgp263ii.zip) y pgp263ii.asc. Éste es la firma del anterior para checar con ayuda de la llave publica de Stale Schumacher que el archivo es auténtico. La llave de Stale resulta al descifrar el archivo keys.asc, que tambien es resultante de la descompresión de pgp263i.zip. El procedimiento para utilizar esto vendrá después.

Tener cuidado al descomprimir pgp263ii.zip de usar el parametro -d para que se haga el subdirectorio \doc y caigan allí los archivos correspondientes.

Sigue a continuacion la lista de los archivos como quedan. Se pueden luego borrar los pgp236ii.asc y pgp236ii.zip habiendo ya comprobado con el primero que el segundo archivo es auténtico, El segundo no se necesita por tener ya su contenido descompactado.

Directorio de C:\PGP	Directorio de C:\PGP\DOC
DOC.<DIR>	
CONFIG. TXT	. <DIR>
DE. HLP	. <DIR>
EN. HLP	
ES. HLP	BLURB.TXT
KEYS. ASC	CHANGES.DOC
LANGUAGE.TXT	KEYSERV. DOC
MD5SUM.EXE	MITLICEN TXT
PGP.EXE	PGFORMAT DOC
PGP.HLP	PGP262I.DIF
PGP263II ASC	PGP263I.DIF
PGP263II ZIP	PGPDOC1.TXT
PGPSORT.EXE	PGPDOC2.TXT
README. 1ST	POLITIC.DOC
README. USA	PRZON26I ASC
SETUP.DOC	

D. Modificar el autoexec.bat

Se añaden los renglones:

```
set pgppath=c:\pgp
set tz=cst6cdt (Así para la ciudad de México: 6 horas al oeste de Greenwich).
set path=c:\pgp;%path%, o bien se añade c:\pgp; en el path
```

E. Modificar el config.txt

Con un editor de textos, por ejemplo el edit de MSDOS de la versión 5 en adelante, modificar el archivo config.txt del PGP. Mientras no se tengan las preferencias propias sugiero las siguientes:

```
MyName = "Juan Hernández" (Cada uno pondrá aquí su nombre entre comillas).
Language = es
CharSet = cp850
Pager = "pgp"
ArmorLines = 400
Armor = on      # Use -a flag for ASCII armor whenever applicable
TextMode = on          # Attempt to use -t option where applicable
Verbose = 1
Interactive = on       # Interactively prompt the user when adding keys (-ka)
EncryptToSelf = on     # Encrypt all messages with your own public key
Legal_Kludge = off     # Use backwards-compatible format
Completes_Needed = 1
Marginals_Needed = 2
Cert_Depth = 4
TZFix=0
```

II. Generar el propio par de llaves, pública y privada.

Se presupone que el programa está debidamente cargado y configurado. Es importante que esté nuestro nombre en el config.txt y que lo usemos igual en el par de llaves.

El comando para generar el par de llaves es muy sencillo. Pero hay que estar preparados.

A. El nombre que va a estar en las llaves

Por ejemplo: Juan Hernánez <jhdez@servidor.unam.mx>

- Es conveniente que vaya una dirección electrónica. En caso de no tenerla puede ir un número de teléfono, o algo que identifique a la persona sin ambigüedades. No utilizar más que letras, signos y caracteres de ASCIIs simples. Por tanto no acentos, no eñes, no signos de abrir interrogación o admiración (¿ ¡).
- También es conveniente que el nombre no sea institucional, sino individual. Si es institucional, varios conocerán la clave (frase contraseña), con mayor riesgo de que se filtre a otros. Y si alguno deja la institución se lleva el conocimiento de la clave y posiblemente el archivo-llave secreto. Si se quiere, quien cifra para la institución puede hacerlo para los diversos individuos de ella usando las claves individuales de cada uno. Se verá más adelante cómo se hace esto.

B. La frase que irá embebida en la llave

O sea la frase clave o contraseña que servirá para abrir la llave privada (secreta). Una frase no muy corta. Que no se adivine fácilmente. Que no sea una frase hecha. Que tenga algunos signos de puntuación, números o caracteres de la tabla de ASCIIs simples. Pueden ser palabras sueltas, combinaciones de letras y signos que no hagan sentido ¡Pero no la vayan a olvidar!

Se entiende de lo que se trata: que se conserve secreta, que no la adivinen cuando los vean teclearla, que tampoco sea demasiado larga porque habrá ocasiones en que tendrán que escribirla varias veces. Que si se les olvida la puedan reconstruir. No conviene tenerla apuntada, o apuntarla de tal manera disfrazado el apunte que sólo el autor pueda entenderlo.

Una vez decidida la frase será bueno practicarla muchas veces hasta que el movimento de escribirla se automatice.

C. Nivel de seguridad

El nivel de seguridad está en relación con el tamaño de las claves que se generarán al cifrar cada vez. Ese tamaño medido en bits. Cuanto más grande el número de bits, también los procesos serán más lentos. Con la velocidad de las máquinas de ahora 1024 bits es razonablemente rápido y con mucha seguridad; la mayoría de los pegeperos (Zimmermann, Schumacher...) las hicieron de 1024

D. Generar el par de llaves

Teclear: pgp -kg

Y luego seguir las instrucciones que irá dando el PGP.
Tendrás al fin de este proceso tus dos llaves.

- La pública en pubring.pgp
- La secreta en secring.pgp

Son además ya los llaveros público y secreto a los que posteriormente podrás añadir nuevas llaves. Las públicas de otros, y otras secretas, si te quieres complicar la vida generando varios pares de llaves. Si te llamas Pedro Pérez, pero un círculo de gente te conoce como "El Gallo" y ni saben que eres Pedro podrás tener dos pares de llaves: dos llaves secretas en secring.pgp y muchas en pubring.pgp Pero por ahora no nos metamos en eso. Vendrá en el capítulo VIII al tratar de manejo de llaveros.

NOTA: Se habrá notado que he estado usando "llave" "llavero" para los archivos, y clave para la frase. El instructivo en castellano habla de clave para los archivos, de contraseña para la frase y de anillo para los llaveros. El "identificador" es tu nombre o parte de él.

E. Qué hacer con los 2 archivos: pubring.pgp y secring.pgp

Lo primero es hacer una copia de ambos en disquete. Un disquete que tengas guardado bajo llave, o bien escondido. Esto es importante por cualquier eventualidad de perder la información de tu compu. Si falla tu disco duro, si un virus te borra esta información, si te distraes y formateas tu disco sin querer, si se daña físicamente alguna parte del disco en donde estén estos archivos, si te roban tu laptop, si...

Si a tu máquina no tienen acceso otros más que tú, pueden quedarse los dos archivos en c:\pgp.

Si otros tienen acceso a tu máquina, bórralos y cada vez que los vayas a necesitar cópialos a c:\pgp y al terminar vuélvelos a borrar. En este caso es bueno que hagas dos copias de tus llaveros en dos disquetes, una en cada uno. Una de respaldo y otra de uso.

No es ésta la mejor solución. Mejor sería tener todo el programa PGP en un disquete bien guardado fuera de la compu, y al ir a utilizar el PGP ponerlo en el tocadiscos correspondiente (a: b:) y dar los comandos desde allí. Esto supone cierta destreza en el manejo del DOS, y puede ser un tema al tratar de cifrar y descifrar archivos que están en un directorio distinto al directorio a:\pgp o al c:\pgp.

Otra solución es tener todo el programa en c:\pgp sin los archivos pubring.pgp secring.pgp y randseed.bin. Estos se tienen en un disquete aparte y en el config.txt se instruye a pgp que busque esos archivos en el tocadiscos a: o en el b: según el caso.

No es aquella primera la mejor solución porque borrar un archivo no es quitar la información de la computadora. Sólo se borra la letra inicial del nombre y se declara en la tabla FAT que ese espacio está libre para grabar encima de lo que haya allí. Habría que borrar con wipefile de Norton, o algo similar.

III. Practicar el cifrado y descifrado a la manera tradicional

A. Por qué esta práctica

- ♦ Porque es útil para guardar cifrados textos y archivos en la propia computadora. No hay necesidad de utilizar llaves, pues sólo se utilizará una clave consistente en una frase. No debe ser la frase embebida en las llaves pública y privada propias, para no arriesgarse a comunicar esa clave si es que se le dice a otro la frase con que se cifró tradicionalmente un archivo determinado.
- ♦ Porque se puede cifrar para quien no ha generado sus llaves, sea que se trate de una sola persona o de un grupo. Con el problema de comunicar de una manera segura la frase-clave sin que la conozcan otros.
- ♦ Porque cifrar y descifrar varios archivos le va dando a uno destreza en el manejo del PGP.

B. Comandos para cifrar y descifrar

- ♦ El comando para cifrar es:

pgp -c archivo.???

En el proceso de cifrar pedirá PGP que escribas la clave, y que la vuelvas a escribir como comprobación. Y el resultado será: archivo.pgp. Si al cifrar añades el parámetro -w, el archivo original no cifrado se borrará de tu disco. Sólo quedará en él el archivo.cifrado

El archivo que vas a cifrar puede tener cualquier terminación menos .pgp y conviene que tenga alguna, porque cuando descifres el archivo.pgp el resultado será archivo. (sin terminación).

- ♦ El comando para descifrar es:

pgp archivo.pgp

Y el resultado será: archivo. (sin terminación) igual a archivo.???

Puedes cifrar cualquier clase de archivo: un texto en ASCIIs simples; un texto con formato de word, o de wordperfect, o de cualquier procesador por el estilo; un archivo de programa, aunque no se le ve mucha utilidad a esto último a no ser que sea un programa original tuyo o de otro que quieras proteger para que otros no lo copien. En el proceso de descifrar te pedirá el programa la clave, o frase contraseña.

C. Cifrar y codificar en ASCIIs simples

Los archivos .pgp no son aptos para ser tratados como texto por un manejador de correo electrónico. Hay programas de manejo de correo (yo conozco el llamado pine) que sí pueden transmitir archivos que no sean de ASCIIs simples, pero sometiéndolos a un proceso de codificación a ASCIIs simples, y mandándolos no como simple correo, sino como archivo pegado (attachement). Pero esto sólo funciona cuando quien recibe el correo tiene también un manejador de correo con la capacidad de descodificarlo. El codificador y descodificador que utiliza pine se llama mime.

El PGP tiene integrado un codificador que transforma cualquier archivo a uno equivalente en ASCIIs simples. Se añade el parámetro -a del siguiente modo:

cifras con el comando: pgp -ca archivo.???

Y el resultado será: archivo.asc. El archivo.asc estará cifrado y codificado en ASCIIs simples, y podrá pasar por cualquier manejador de correo electrónico.

Descifras con el comando: pgp archivo.asc

Y el resultado será: archivo. (sin terminación) y será igual a archivo.???

D. Guardar tu llave pública en un archivo que sea tu_nombre.asc

Ya que hemos hecho ejercicio de generar archivos cifrados en ASCIIs simples, es conveniente también hacer un archivo en ASCIIs simples cuyo contenido sea el pubring.pgp. Así podrán pasarse por correo electrónico las llaves públicas.

Por haber generado tu par de llaves, tienes los archivos pubring.pgp y secring.pgp. Ahora sólo trataremos del primero, pues no tiene caso trabajar así con el secring.pgp, pues es un archivo que no se pasará nunca ni por correo ni de ninguna otra manera.

El comando es: pgp -a pubring.pgp

En el proceso se te preguntará que si ese archivo fue generado por PGP. Le dices que sí y el proceso proseguirá. El resultado será el archivo pubring.asc

Como hasta ahora el contenido de pubring.asc es sólo una llave pública, la tuya, puedes copiarlo como tu_nombre.asc. Y esa es tu llave pública en aptitud para transitar por correos electrónicos. Pero en realidad, a la corta o a la larga, no guardarás así simplemente tu llave pública para pasarla por correo-e, porque querrás mandarla hasta que otro de fe de ella o la autentifique con su firma. Asunto posterior del capítulo IV.

E. El PGP habla de "armadura" de ASCIIs

Porque utiliza la imagen de que ha metido un archivo en una armadura. Se metían en las armaduras para protegerse de los proyectiles enemigos (piedras, flechas, balas de arcabuz). Ahora se meten los astronautas en su traje espacial para protegerse del ambiente hostil que es la falta de atmósfera y de gravitación. Metemos los archivos .pgp en una armadura (.asc) para que puedan transitar por ese medio específico que son los correos electrónicos que dejan intactos los caracteres de la tabla de ASCIIs simples y o no entienden o modifican los demás. Si en el config.txt activamos la línea "armor = on", estamos instruyendo a PGP que siempre que se pueda meta en armadura de ASCIIs simples los archivos que genere.

IV. Intercambiar llaves públicas

A. Conservar la propia llave pública

Si quiere uno conservar aparte por sí sola la propia llave pública se hace una copia del pubring.pgp con otro nombre. Por ejemplo jhdez.pgp. Así, si alguien pide la llave pública se le copia el archivo jhdez.pgp en un disquete suyo. Es la manera directa de pasar la llave pública a otra persona. El tendrá la seguridad de que es la llave verdadera porque está seguro que se la copió el dueño en su disquete.

NOTA: Si tu pubring.pgp ya tiene varias llaves, hay otra manera de que extraigas de allí tu llave pública. Pero de esto se tratará más adelante.

El archivo jhdez.pgp, que es la copia de pubring.pgp cuando sólo tiene la llave pública de Juan Hernánez, no es capaz de transitar por el correo electrónico porque es un archivo en el que hay caracteres que no son lo que hemos llamado ASCIIs puros (del 33 al 127 de la tabla de ASCIIs). Pero se puede meter jhdez.pgp en una armadura de ASCIIs puros con el comando:

pgp -a jhdez.pgp. El resultado será: jhdez.asc

Y este archivo sí puede transitar por el correo electrónico. Y es la manera como podemos pasar nuestras llaves públicas a otros por correo electrónico.

B. Guardar llaves públicas en el llavero propio

Quien recibe en su correo una llave pública la puede meter a su llavero público con la orden: pgp archivo.???, donde "archivo.???" es el nombre del archivo de su correo.

Por ejemplo: Yo recibí un correo que pasé a mi computadora como correo.txt. Ese correo es una llave pública. Con la orden pgp correo.txt me dirá el programa que el contenido es una llave pública y que si la quiero añadir a mi llavero. Si le digo que sí, aumentará mi pubring.pgp con esa nueva llave.

C. Certeza sobre que la llave es de quien dice que es

Pero nos topamos con un problema de seguridad. ¿Estoy cierto de que esa llave pública es verdaderamente de quien dice que es? ¿No se metió en la cuenta del amigo que me escribe otra persona y me está mandando una llave pública que no es la de él como si sí lo fuera? Cualquiera puede generar un par de llaves asumiendo cualquier nombre.

1. Solución por medio de la huella digital de la llave

Pedro Pérez me mandó por correo su llave pública y yo ya la añadí a mi pubring.pgp. Veo la huella digital de la llave de Pedro con el comando:

pgp -kvc pedro pubring.pgp.

la respuesta será una serie de números y letras que apunto y luego le telefoneo a Pedro Pérez. Estando seguro que es él porque le conozco la voz, y porque le llamé a su casa y por lo que platicamos, le digo que recibí su llave pública y que la huella digital es tal y tal. El checa con la misma orden de arriba en su llavero público y me dirá que sí o que no es, según el caso.

[La huella digital consiste en dos series de ocho números hexadecimales de dos guarismos cada uno]

2. Solución por medio de llaves firmadas, o autenticadas

Hablemos de Pedro Pérez, de Fernando Fernández y de Juan Juárez.

Pedro Pérez recibió por correo-e la llave de Fernando Fernández, y venía firmada (autenticada) por Juan Juárez. Por otra parte Pedro tiene en su llavero la llave pública de Juan recibida directamente del mismo Juan. Al descifrar con la orden pgp correo.txt el programa le dirá que es la llave de Fernando y que viene firmada por Juan. Si no tiene en su llavero la llave de Juan el informe que recibirá es que viene firmada por un firmante desconocido. Cuando añada a su llavero la llave de Juan todo se arreglará y Pedro tendrá la garantía dada por Juan de que la llave de Fernando es realmente de Fernando. Si Pedro le cree o no a Juan, es su bronca.

D. Guardar una llave recibida en formato llave.pgp y no en llave.asc

Pero yo no recibí por correo una llave. Me la dio en disquete su dueño y no viene en formato .asc así que no puedo simplemente descifrarla con la orden pgp jhdez.pgp. ¿Cómo la añado a mi llavero?

La añado con la orden: pgp -ka jhdez.pgp pubring.pgp

Aquí supongo que "jhdez.pgp" es la llave pública tal como me la dieron quizá ya firmada por otro.

E. Cómo se firman las llaves

Cada uno puede firmar la o las llaves que tenga en su llavero. Para esto lo primero es incorporar al propio llavero la o las llaves que va uno a firmar. Por ejemplo: en mi llavero tengo las llaves mía y las de Pedro Pérez, Fernando Fernández y Juan Juárez. Y quiero firmar la de Juan. Procedo: pgp -ks Juan pubring.pgp

y va uno contestando las preguntas que vaya haciendo el programa. Al final queda firmada por mí la llave de Juan.

F. Cómo aislo la llave de Juan Juárez

Se aisla una llave para poder tenerla sola y pasarla a otros y al mismo Juan Juárez para que él la tenga (y pueda pasarla) con todo y mi firma.

con el comando: pgp -kx juan juan.pgp pubring.pgp

o si quiero el archivo en armadura de ASCIIs puros:

pgp -kxa juan juan.asc pubring.pgp

Y según la orden usada tendré la llave pública de Juan Juárez ya sea como juan.pgp, o como juan.asc.

- Sugerencia:
  Varios que estén relacionados por su trabajo, o por cercanía geográfica, o por otros motivos pueden nombrar a uno que sea algo así como su notario oficial para autentificar las firmas de ese grupo. Ese "notario" debe ser conocido por varias de las gentes conocidas a su vez en su sistema de comunicación electrónica. El notario consigue en directo que los conocidos del sistema le autentifiquen su llave pública en forma directa con presencia personal; él, a su vez, irá autentificando por conocimiento directo las llaves públicas de los de su grupo. Pero sin que la firma de ese notario se convierta en obligatoria. Cada uno puede hacer certificar su llave pública de la manera que quiera. Los "notarios" son para ayudar, pero no para adquirir el control de la autenticidad de las llaves de determinado grupo.

G. Examinar nuestro llavero público

Con las llaves públicas que vamos recibiendo de otros se va enriqueciendo nuestro llavero pubring.pgp. Podemos, si así lo deseamos, asomarnos dentro del llavero para saber cuántas llaves tenemos y de quiénes.

La orden pgp -kv pubring.pgp

da como resultado el contenido del llavero;

La orden pgp -kvm pubring.pgp
o bien: pgp -kv pubring.pgp | more

da el contenido del llavero, pero parte por parte. No pasa por la pantalla todo el contenido, sino que se va deteniendo pantalla por pantalla.

La orden pgp -kv pubring.pgp > archivo.txt

Da por resultado un archivo de texto con el contenido del llavero.

NOTA: En cualquiera de estos casos se puede añadir otra "v" a los parámetros del comando (-kvv, -kvvm). El resultado dará más detalles de información: también aparecerá con qué firmas está autenticada cada llave. Si en tu config.txt has instruido a PGP que siempre actúe con muchas palabras (verbose = 2), -kv funcionará como si fuera -kvv
-kv = ver el llavero
-kvv = ver el llavero con respuesta de más palabras ("verbose")

NO NOS CONTENTEMOS CON SABER ESTO. Practiquémoslo. Tómese cada uno el tiempo para examinar el contenido de su llavero público, pubring.pgp; para verlo en pantalla y para generar un archivo de texto, con menos y más información.

H. Observaciones

- Algunos han puesto como identificador un nombre: Mauricio, javier. Esto dificultará cifrar documentos si es que hay otro Mauricio u otro Javier. ¿Cómo distinguirlos? Es bueno que pongan también su apellido. Pueden modificar su identificador con el comando:

  pgp -ke identificador [llavero], o en concreto:
  pgp -ke Mauricio pubring.pgp

  Y seguir las instrucciones que vayan saliendo. Quedará una llave con dos identificadores. Luego habrá que quitar el identificador incompleto con la orden: pgp -kr mauricio pubring.pgp, y asímismo ir siguendo las instrucciones de PGP.

- Alguno sólo ha puesto como identificador su nombre institucional. No es buena política la de que varios individuos compartan una misma llave con su clave (contraseña). La llave privada y la frase con que se abre ésta son secretos que deben ser muy bien guardados.

- El formato de los identificadores suele ser:

  fulano de tal <dirección electrónica>
  Juan Hernandez <jhdez@servidor.unam.mx>

Si alguno no tiene dirección electrónica puede poner dentro de los paréntesis angulares, un número de teléfono, o un apodo, o algo que lo identifique a él solo.

I. Digresión de preguntas y respuestas

Esto no forma parte del instructivo en directo. Son preguntas y respuestas que tienen que ver con lo que va del instructivo.

(P/): Hoy recibí tu mensaje y ya estuve practicándolo, primero saqué todas las firmas que venían en el programa (Zimmerman y demás).

(R/): Te sirvió de ejercicio y eso es bueno. Supongo que te refieres a las que vienen en el programa como keys.asc. Son las de los productores y mantenedores del PGP. Yo lo que hice fue guardarlas como un llavero aparte. Hasta ahora no me he metido a explicar como hacer diversos llaveros. De hecho yo hice un llavero que llame "pgpjefes.pgp", con su poco de humor y puse allí todas esas llaves además de la mía.

¿Como se hace? Con pgp -ka pubring.pgp pgpjefes.pgp, y cuando el PGP te pregunte si vas añadiendo tal y tal llave al llavero pgpjefes.pgp le vas contestando sí o no. Para esto debes tener activado en tu config.txt la linea interactive = on, porque si no, te pasa de golpe todas las llaves contenidas en keys.asc y tendrás que ir quitando una por una las que no quieras en el llavero pgpjefes.pgp.

Te sirven para checar los programas de PGP que hayas bajado o los que puedas bajar en el futuro. También por si alguna vez le escribes en cifra a Zimmermann o algún otro de ellos. Yo he estado usando la de Armando Ramos, que es el que se ha encargado de hacer las traducciones al castellano. Pero mayor utilidad no creo que tengan.

P/: ¿Cómo firma mi llave pública otro con quien puedo estar en contacto directo?

R/: Fulano tienen tu llave publica. Tú eres sutano. Entonces la firma él en su llavero:

pgp -ks sutano pubring.pgp

Luego la extrae de su llavero:

pgp -kx sutano sutano.pgp pubring.pgp, o:
pgp -kxa sutano sutano.asc pubring.pgp

Luego te da la copia del archivo sutano.pgp o sutano.asc

- Si es sutano.asc, con el simple descifrarlo irá a tu pubring.pgp. Si tu sutano.asc está en un disquete puedes usar la orden: pgp a:\sutano.asc.

Esto sucede si tienes en tu autoexec.bat el set pgppath = c:\pgp o el directorio correspondiente.

Y luego distribuyes por el correo electrónico ese archivo sutano.asc, que es tu llave publica firmada por fulano, en armadura de ASCIIs simples.

- Si es sutano.pgp, metes su contenido a tu llavero con:

pgp -ka sutano.pgp pubring.pgp

Y después extraes tu llave para el correo electrónico:

pgp -kxa pubring.pgp sutano.asc

P/: En la llave de perengano que le firmé personalmente, hay varias firmas que aparecen como desconocidas, ¿a qué se debe eso, a que yo no las tengo como registradas o que las firmas en sí no están validadas? ¿Me expliqué o me hice bolas?

R/: Para que el PGP reconozca una firma, tiene que estar la llave pública del que la firmó, en el mismo llavero. O sea, que deben estar en tu pubring.pgp las llaves públicas de los que firmaron la de perengano. Pero lo importante es que haya una a la que tú le confíes y entonces la firmas para ti en tu llavero. Luego podrás quitarle luna firma inútil con la orden::

pgp -krs perengano pubring.pgp

P/: ¿cómo puedo confirmar tu llave pública?

R/ La bronca está en qué tan cierto estás de que yo soy la persona que digo ser en el correo electrónico. Que no me estoy inventando una personalidad para el correo electrónico, ni que haya otra persona que se ha metido a mi cuenta de correo electrónico simulando que soy yo y así ha generado un par de llaves pública-secreta y ha estado comunicando la pública como si fuera mía. Podemos andar en trabajos y actividades por la paz, por los derechos humanos, etc., que no les caen bien a muchos sectores del gobierno y a otros con capacidad de organizar sus servicios de inteligencia. Supón que alguno de ellos logra hacerse pasar por mí, y al principio en muy buena onda da tiempo y trabajo a ayudar a los grupos de los que desea extraer información y vigilar. Y luego, ganada la confianza, saca la información que anda buscando.

Por tanto la única manera de que puedas estar seguro de que mi llave pública que tienes es realmente mía es que te la dé en persona. O que alguien en quien tú confías te la dé en persona y la haya recibido de mí en persona. O que ese alguien en quien confias y de quien recibiste en persona su llave pública te mande, firmada por él, mi llave pública.

V. Cifrar archivos al modo del PGP

A. Cifrar para otros

Se cifra para otros con la orden:

pgp -e archivo.??? fulano [otros], o como está en la ayuda:
pgp -e fnormal identificador_des

Ejemplo: pgp -e carta.doc fulano

Así cifras "carta.doc" para fulano. El resultado, o sea, el archivo cifrado, será carta.pgp; De haber cifrado con pgp -ea, el resultado habría sido carta.asc. Y si hubieras usado el parámetro -w el archivo original no cifrado se habría borrado de tu disco.

B. Pero salen varias dudas

1. Hay dos o más que tienen el mismo identificador

Para evitar identificadores repetidos se usa como identificador o bien una palabra que no se repita en la lista con ningún otro, o emplear algo más largo que una sola palabra.

Por ejemplo, para Juan Hernández <jhdez@servidor.unam.mx> el identificador puede ser juan o hernandez o jhdez. Usar el que no se repita con otros. Puede haber una llave de Juan Martinez o de Pedro Hernandez. Ya no puedo usar ni 'Juan', ni 'Hernandez' como identificador.

Las comillas vienen en nuestra ayuda. Puedo escribir "juan hernandez" (con las comillas) como identificador. (Hay otro modo de identificar llaves. Ver más adelante: IX D)

2. Cifrar para varios

Sabemos que se pueden poner uno tras otro muchos identificadores. Pero hay un límite: el renglón del comando no puede ser de más de 127 caracteres. Si la lista de aquellos para los que quiero cifrar es grande, y más si he de usar identificadores largos agrandados por las comillas y de varias palabras, me paso del límite permitido en una sola línea de comando.

La versión 2.6.3i tiene una solución, que no está en la versión 2.6.2i. En ésta no queda más remedio que cifrar varias veces y luego tener cuidado de mandar lo cifrado precisamente a aquellos para los que se cifró cada vez. Bien molesto y con peligro grande de equivocaciones. Y no se puede mandar lo mismo a toda la lista. Y por esta razón se aconseja que cambien a la versión 2.6.3i.

En la versión 2.6.3i se pone como identificador el nombre de un archivo de texto precedido por -@ Y por otro lado se escribe un archivo de texto con los identificadores, uno en cada renglón. Así, la orden para muchos será: pgp -e carta.doc -@llaves.txt

Con el supuesto de que llaves.txt es un archivo con la lista de los identificadores. Un ejemplo:

Juan hernandez (sin necesidad de comillas)
pedro hernandez (sin necesidad de comillas)
fulano
sutano
perengano
armando flores (sin necesidad de comillas)

3. Cómo firmar el texto además de cifrarlo

Se añade el parámetro -s. Enviando un archivo cifrado sé que sólo el o los destinatarios podrán leerlo. Recibiendo un archivo firmado sé que realmente proviene de quien dice que me lo manda, puesto que tengo su llave pública. Sin ella recibiría el mensaje de "firmante desconocido".

pgp -es archivo.??? identificador [otros identificadores]

Y para meterlo en armadura de ASCIIs puros:

pgp -esa archivo.??? identificador [otros identificadores]

4. Si mi archivo por cifrar es muy grande

Si el archivo que se va a cifrar y meter en armadura de ASCIIs simples es grande, pueden resultar dos o mas archivos cifrados y numerados. Si cifré el archivo cartas.doc podrán resultar: cartas.as1, cartas.as2,

5. Un escrito muy confidencial

¿Puedo cifrar mi carta confidencial de tal modo que quien la reciba sólo la pueda leer en su pantalla y no se grabe en su computadora?

Sí. Es como decirle: "este texto se borrará una vez que lo leas. No se grabará. Si quieres volver a verlo lo tienes que descifrar de nuevo." Se supone que se trata de un texto legible directamente en pantalla.

La orden es con el parámtero -m:

pgp -e[s][a][w]m carta.txt identificador.

Va la [s] entre corchetes para indicar que puede o no firmarse el documento según uno quiera poniendo o no ese parámetro. Lo mismo la [a] por si quiere uno o no usar la armadura de ASCIIs.

Si quieres que tu archivo original se borre en tu máquina añades el parámetro -w{wipe}

C. Otra digresión de preguntas y respuestas

P/: Ahora bien, más importante aún es continuar utilizando este sistema entre nosotros/as para que llegue el momento en que sea la forma común de enviarnos TODOS los mensajes. Esto nos evitará llamar la atención a los "ojos" y "oídos" invisibles.

R/: Sin duda que lo que sugieres es una buena política de seguridad. Es un poco más de trabajo, esto de cifrar todo antes fuera de línea. Existe un programa para manejar el PGP en combinación con lectores fuera de linea. Hay también programas que cifran lo que se va a mandar por correo-e y lo descirfran al recibirlo.

P/: No me funcionó el comando: pgp -esa carta.doc -@llaves.txt. El que sí me funciona es: pgp -esa carta.doc fulano sutano (etc.)

R/: No te funciona ese parámetro (-@llaves.txt) porque no tienes la ultima versión del PGP que es la 2.6.3i. Esta posibilidad de usar este parámetro es de las cosas nuevas de la versión última.

El comando pgp -esa carta.doc fulano sutano (etc) tiene la limitante de que la línea de comando del PGP admite un máximo de 127 caracteres. Además, claro, de la que dices de que no siempre se acuerda uno de poner todos los identificadores. Por eso la insistencia en que consigamos esta nueva versión.

VI. Descifrar los archivos recibidos

A. Caso ordinario

Conocemos ya la orden, que es la más sencilla del PGP, puesto que no lleva parámetro ninguno. La conocemos por haber ejercitado el cifrado y descifrado a la manera tradicional. Y es: pgp archivo.???

La extensión del archivo por descifrar será generalmente o bien pgp o bien asc. Pero puede ser otra, como cuando bajas de tu correo un archivo con un nombre dado por ti. El resultado del descifrado será un archivo sin extensión ninguna. Descifras, por ejemplo, el archivo carta.asc, y el resultado será el archivo carta.

Si tienes un archivo híbrido en el que hay texto abierto, y texto o textos cifrados, o alguna llave pública, el PGP es lo suficientemente ducho como para ir distinguiendo los diversos componentes. Y te irá guiando en lo que hay que hacer.

B. Situaciones especiales al descifrar

1. Quiero leer el archivo descifrado en mi pantalla, sin que se grabe en mi disco.

Esto evidentemente sólo se aplica si estoy seguro de que lo que estoy por descifrar es un archivo de texto legible directamente en pantalla. Se utiliza el parámetro -m. La orden es:

pgp -m carta.txt

Y de todos modos tendrás la opción de guardar en un archivo lo que hayas leído pantalla por pantalla. Podrás darle un nombre que incluya una ruta para que se grabe en otro directorio.

2. *Me llegaron varios archivos cifrados con el mismo nombre y extensiones en serie: carta.as1, carta.as2, carta.as3...*

Esto quiere decir que el pgp al cifrar un archivo produjo más líneas que las marcadas como límite en el config.txt y fue entonces construyendo uno o varios archivos de ése número de líneas y el último más corto.

Por correo-e los recibo de varias formas:

- En una sola sesión bajo todo mi correo en un solo archivo. Éste puede traer texto ordinario y mensajes cifrados. Para el manejo de estos, simplemente descifro ese archivo que es todo mi correo. El programa sabrá qué hacer con él, y si vienen varios trozos del mismo archivo cifrado en su orden natural, el PGP lo descifrará sin dificultad, Mandará un mensaje de error y advertencia si no están en su orden debido.
- En la misma sesión ví que había varios mensajes pegepeados y los "subjects" me lo avisan. Los voy bajando con el mismo nombre: carta.as1, carta.as2... Y luego para descifrarlos los uno en solo archivo con la orden de DOS:

  copy carta.as1+carta.as2+carta.as3 carta.asc

  Y así me resultará el archivo carta. asc, que es el que descifraré.
- No hay ninguna indicación. Entonces yo examino el correo pegepeado, y allí vendrá, dentro de cada correo, la indicación de qué número de serie es. Y esto antes o después de bajarlo. Fuera de línea hago la fusión como arriba con copy y sumas.

3. *Al descifrar me dice el PGP que el archivo está firmado, pero que el firmante es desconocido*

Esto quiere decir que no tengo en mi llavero la llave pública de quien cifró para mí el archivo recibido. De todos modos lo podré descifrar.

VII. Firmar archivos de texto abierto

En ocasiones lo que nos importa no es que no se pueda leer un texto que escribimos, pero sí que quien lo recibe tenga la seguridad de que ese texto salió de nosotros y no tiene absolutamente ningún cambio. O sea, mandar un texto firmado, pero no cifrado.

A. Para qué firmar sin cifrar

Para comunicar algunas instrucciones que otro con quien sólo nos comunicamos por correo electrónico las ejecute, con la seguridad de nuestra parte de que no fueron modificadas por nadie, ni en el camino, ni por el mismo que las recibió. Que no haya eso de que tú me lo escribiste así.

Para mandar disquete con instrucciones o cualquier otro escrito del que queremos que quien lo reciba esté seguro dee que nada ha cambiado. El correo que transporta el disquete podrá cambiar el texto, pero si lo hace, el PGP dará el mensaje de firma inválida.

Para que un convenio quede escrito y las dos o más partes involucradas en él tengan la certeza de que todos manejan el mismo texto y precisamente el texto en el que quedaron de acuerdo. Y esto sin necesidad de que hagan la firma del convenio en el mismo lugar y al mismo tiempo. Puede hacerlo cada uno en su lugar a su conveniencia. Supongamos que tú yo hemos estado hablando sobre un convenio que queremos quede escrito y firmado por ambos. Pero cuando hablamos no tuvimos todavía la posibilidad de perfilar todos los detalles, porque había que consultar a otras personas o datos en el lugar donde vivimos o hemos estado negociando los términos del convenio sólo por correo electrónico. Cuando yo ya llegué a la redacción del convenio a mi satisfacción y pienso que también a la tuya, escribo el convenio, lo pgpfirmo y te lo mando. Si tú estás de acuerdo lo pgpfirmas y me lo mandas así. Los dos tenemos el mismo texto firmado por ambos. Cuando tú compruebas mi firma el PGP te genera un texto sin firmar cuyo nombre de archivo queda sin extensión. Recibo "convenio.txt" firmado por ti, leo el aviso de que la firma es buena y de que tengo un archivo que se llama "convenio.". (Es igual a convenio.txt pero sin la firma.) Éste es el que firmo y te mando.

En general usaremos la pgpfirma de igual manera que la firma manuscrita. Claro que no exactamente igual porque ni los bancos ni los notarios ni los tribunales aceptan todavía estas firmas como buenas. Siempre las legislaciones van atrás y a veces muy atrás de la vida.

B. Condiciones para pgpfirmar un texto abierto

- Que el texto esté en ASCIIs simples. Con la versión 2.6.3i pueden usarse los caracteres de texto de los conjuntos de caracteres de 7 y 8 bits para textos. Esto concretamente significa para nosotros que los textos que se irán a firmar pueden llevar acentuadas, ñs y signos de abrir interrogación y admiración. Y el PGP se encargará de avisar cuál es el conjunto de caracteres que está uno usando. Será el que esté declarado en el config.txt. Si revisan el config.txt que copié en I E verán que existe la línea: CharSet = cp850.
- Que decidamos si vamos a querer la firma con o sin armadura de ASCIIs. En el fondo creo que será bueno que siempre sea en armadura porque puede uno necesitar de pronto enviarlo por correo electrónico, y por otra parte la función de firmado es la misma.
- Que tengamos nuestra llave privada (que es secreta) en el secring.pgp

C. Procedimiento

Utilizar la orden: pgp -tas carta.txt. El PGP usa tu llave secreta y por eso necesita abrirla y te pregunta tu clave. El resultado será carta.asc, o, si no usaste el parámetro -a, carta.pgp. Vale la pena que por vía de ejercicio pgpfirmes algunos documentos en una forma y en otra y luego los examines y veas los diversos resultados.

Tener en tu config.txt activados los renglones:

```
Armor = on        # Use -a flag for ASCII armor whenever applicable
TextMode = on     # Attempt to use -t option where applicable
```

equivale a escribir los parámetros -at. Si PGP detecta que hay caracteres que no pertenecen a ningún conjunto para textos, meterá a la armadura no sólo la firma, sino también el texto.

D. Comprobar la firma

El mismo procedimiento que para descifrar: pgp carta.asc, o: pgp carta.pgp

El programa te dirá después de una comprobación si la firma es buena o no y te dará un pequeño informe. Además te generará un archivo sin terminación que es lo que te llegó firmado, pero sin la firma. Y éste es el que firmas tú para mandármelo a mi si se trata de un convenio. Así, yo tendré el convenio firmado por ti y tú por mí.

E. La pgpfirma en archivo separado del texto

Es posible pgpfirmar un texto de modo que resulte un archivo con la sola firma separada del texto. Al comprobar una firma así con la orden pgp firma.asc pedirá el programa que se indique el archivo al que se refiere esa firma. Si se lo escribimos correctamente y tenemos ese archivo sin ninguna modificación, dirá que la firma es buena.

La ventaja principal es para cuando se quiere firmar un convenio de más de dos. Si el convenio es entre cinco, no habrá que tener cinco textos firmados cada uno con cada una de las cinco firmas. Un sólo texto y cinco archivos de firmas. Pero ¡Ojo! Nos irán llegando las firmas con el mismo nombre: carta.asc. Tener cuidado de renombrarlos, por ejemplo, carta01.asc, carta02.asc... , Pero es mejor que cada uno al firmar d´´e un nombre específico a la firma separada. Por ejemplo, cartajh.asc (jh son las iniciales de Juan Hernández) en vez de sólo carta.asc.

Se obtiene la firma por separado con la orden pgp -sba carta.txt y el resultado es carta.asc, que es la firma sola. Para que el resultado sea otro la orden será: pgp -sba carta.txt -o cartajh.asc (Ver IX B)

Entre los archivos del directorio \pgp distribuido en la versión 2.6.3I hay uno que es firma por separado: pgp263ii.asc. Intenten descifrarlo como ejercicio. Es la firma de pgp263ii.zip.

Como se ve esto de las firmas separadas es un poco complicado y sobre todo no ha sido hasta ahora de uso frecuente y por eso se olvida fácilmente cómo funciona. Cada uno verá si le invierte tiempo ahora en la práctica de esto, o mejor cuando vea que lo va a necesitar.

VIII.Manejo de llaveros y de llaves

Ya hemos visto algunas de las funciones del programa que tienen que ver con el manejo de llaves y llaveros. Pero hasta ahora hemos estado en el supuesto de que sólo tenemos un llavero público, el puring.pgp y uno privado, el secring.pgp. Pero puede ser que queramos tener varios llaveros públicos (y quizá varios secretos, lo cual creo que tiene menos utilidad y complica mucho las cosas). El o los llaveros públicos contendrán varias o muchas llaves. El privado suele tener sólo una, a no ser que hayamos generado varias llaves con identificadores y claves (frase-contraseña) distintas, o que compartan varios el mismo programa..

Manejar llaveros muy grandes hace todas las operaciones muy lentas. Zimmerman lo dice en el documento número 2 [1]. Por muy grande piensa en más de un par de cientos. Y no creo que por lo pronto queramos tener más. Creo que la utilidad de tener varios llaveros públicos es más bien para agrupar las llaves públicas por categorías de personas. En todos los llaveros es bueno que esté la propia llave pública. Por vía de ejemplo: yo tengo tres llaveros además del pubring.pgp: uno con las llaves que vienen en el archivo keys.asc y que son las de los que desarrollan de alguna manera el PGP, otra con las llaves de un grupo de amigos y otro con las de un grupo que trabajamos en común , y en mi pubring.pgp he sumado los dos últimos.

Viendo con pgp -h el archivo de ayuda del programa se darán cuenta que muchas veces el llavero (en ese archivo es llamado "anillo") viene entre paréntesis cuadrado para denotar que es opcional. Si no se escribe en el comando, el programa usará el llavero por omisión (por defaul, decimos a veces), que es el pubring.pgp o el secring.pgp según el caso. Y con el nombre que esté declarado en el config.txt.

En esa ayuda se habla de fdeclaves (archivo de llaves) y de anillo (llavero). ¿Son lo mismo? No exactamente siempre, aunque a veces sí. Los llaveros son un conjunto de llaves con terminación pgp y el formato propio de las llaves. A veces un llavero cumple también la función de un archivo de llaves. El archivo de llaves puede tener terminación y formato .asc; puede tener cualquier otra terminación y estar revuelto con textos. Como cuando nos llega en el correo

---

[1] Manejar llaveros muy grandes:

> PGP fue diseñado originalmente para manejar llaveros pequeños para tener las llaves de las amistades, a la manera de un rolodex pequeño. Un par de cientos de llaves es un tamaño razonable para un llavero así. Pero al hacerse PGP más y más popular las gentes tratan de meter llaveros muy grandes a su llavero. Lo cual a veces significa añadir miles de llaves al llavero. La actual versión de PGP no es capaz de manejar tales llaveros en un tiempo razonable.
>
> Si quieres importar a tu llavero otro llavero de muchísimas llaves pero sólo porque te interesan unas cuantas docenas de ellos, sería más eficiente ir extrayendo las pocas llaves que te interesan y sólo meter ésas a tu llavero.
>
> La verdadera solución es mejorar el programa PGP con técnicas avanzadas de manejo de base de datos. Estamos trabajando en ello y esperamos tenerlo pronto. Pero hasta que esto suceda tendrás que usar llaveros chicos, o tener paciencia.

electrónico un mensaje y dentro de él una o varias llaves. O porque bajamos del correo todos nuestros mensajes en unsólo archivo y allí se encontrarán textos abiertos, textos pegepeados y llaves. En estos casos al descifrar ese archivo el pgp dirá si se encuentra una llave. Para las otras funciones de manejo de llaves archivo de llaves será un conjunto de llaves en formato .pgp o en formato .asc.

El cuadro que sigue nos dice lo que ya sabemos, aunque viene con información adicional, porque puede uno tener varios llaveros públicos e incluso varios secretos

Generar nuestro par de llaves pública-privada	pgp -kg El resultado son pubring.pgp, secring.pgp y randseed.bin (Luego veremos para qué es este archivo)
Añadir llaves nuevas a un llavero. Al generar tu par de llaves te resultó un par de llaveros de una llave cada uno: pubring.pgp y secring.pgp	pgp -ka archivo_origen [llavero_destino] Si no se escribe el llavero_destino va al defaul, que es pubring.pgp, o el que se haya indicado en el config.txt, o el secring.pgp según que el contenido del archivo_origen tenga por contenido llaves públicas o privadas.
Meter nuestro llavero público, con una o con muchas llaves, a una armadura de ASCIIs simples con la que es capaz de sobrevivir en el medio inhóspito de las máquinas de 7 bits.	pgp -a pubring.pgp El resultado será pubring.asc, que no es llavero, sino archivo de llaves. Es la colección de llaves en formato .asc, pero no funciona como llavero.
Sacar copia de cualquier llave que esté en nuestro llavero público (e igual se hace con el llavero privado o secreto) y conservarla en un archivo cuyo contenido es sólo una llave.	pgp -kx ident archivo_destino.pgp llavero_origen. O bien: pgp -kxa ident archivo_destino.asc llavero_origen
Examinar cualquier llavero para saber de qué llaves consta. La segunda "v" opcional se usa para ver también qué firmas tiene cada llave (v=verbose). Si no se escribe ningún ident se verán todas.	pgp -kv[v] [identif] [llavero]
Suprimir una llave o sólo un identificador, de un llavero	pgp -kr ident [llavero]
Añadir un identificador o modificar la clave en el llavero privado	pgp -ke ident [llavero]
Firmar la llave de un llavero público	pgp -ks ident [-u propio_ident] [llavero] El parámetro -u propio_ident sirve para cuando en el mismo llavero privado hay varias llaves. O porque uno mismo se ha generado varias llaves, o porque dos o más comparten el mismo llavero secreto, cada uno con su llave.

Y siguen más funciones de manejo de llaves y llaveros que no han salido todavía en esta guía:

Examinar la huella digital de las llaves	pgp -kvc [ident] [llavero]
Modificar el valor de mi firma en la llave de otro, presente en un llavero público mío.	pgp -ke ident [llavero]
Desactivar (o activar si estaba desactivada) una llave pública.	pgp -kd ident [llavero]
Suprimir una o más firmas de una llave.	Pgp -krs ident [llavero]
Revocar una llave propia. Esto no lo he hecho nunca. Se emite un certificado de revocación. No sé si al mismo tiempo queda desactivada para siempre la propia llave.	Pgp -kd mi_ident

Tomo (no a la letra):del archivo pgpdoc1.txt

Revocar una llave pública

Supongamos que tu llave secreta y tu clave (frase contraseña) ya están en manos de otros. Hay que avisar a todo el mundo para que dejen de usar esa llave pública tuya. Para esto se expide un certificado de revocación. Se obtiene esto con:

pgp -kd tu_ident

Este certificado lleva tu firma hecha con la misma llave que estás revocando. Deberás distribuir ampliamente el certificado de revocación lo antes posible. Quienes lo reciban lo podrán añadir a sus llaveros públicos y su PGP impedirá que se use esa llave si es que por error la quieres usar. Te tocará generar un nuevo par de llaves y volver a difundir tu nueva llave pública.

Y puede ser que quieras revocar tu llave por alguna otra razón. Pero el procedimiento es el mismo.

Casos conocidos:: robo de la computadora, olvido de la frase clave que abre tu llave secreta...

IX. Complementos

Quedan aún algunos temas por tratar que si se hubieran explicado junto con los anteriores, quizá hubieran complicado de más cada tema. Sin embargo, precisamente porque queremos ayudarnos todos en esto, vuelvo a pedir sus sugerencias para mejorar las explicaciones o el orden, o todo lo que les parezca que ayuda a que otros se animen y usen el PGP.

A. El parámetro -p

Al descifrar un archivo es posible restablecer el nombre original del archivo. No lo he usado porque es fácil que varios le den el mismo nombre, por ejemplo, juan.txt (porque le escriben a Juan Hernández). Y como se puede determinar el nombre resultante como uno quiera con el parámtero -o prefiero escogerlo yo.

B. El parámetro -o

Con el parámetro -o señala uno el nombre resultante, o sea el nombre del archivo descifrado. Si no se usa ni el -o ni el -p el archivo resultante es, como ya hemos visto, el mismo nombre del que se descifra sin ninguna terminación. El parámetro -o también sirve para dar un nombre con todo y ruta (path) y así cae lo descifrado en el directorio que uno quiere.

Supongamos que quiero tener lo que voy recibiendo en un directorio que se llama \cartas\Juan y Juan me manda una carta cifrada carta.asc; entonces con el comando:

pgp -o c:\cartas\juan\c960717.txt

el archivo descifrado será c960717.txt en el subdirectorio \cartas\juan. Se supone que el subdirectorio \juan ya existe en mi compu.

C. El parámetro -u

Si hay sólo una llave en el llavero privado (secreto) el parámetro -u no es necesario. Si varios comparten un mismo llavero privado y por tanto hay en él varias llaves, con el parámetro -u se le indica al programa cuál de esas llaves debe usar para firmar archivos o llaves.

D. El número hexadecimal de la llave como identificador

Se puede usar como identificador el número hexadecimal de la llave precedido por el prefijo 0x (cero x). Así, mi identificador puede ser 0xAF32327D; el de fulano, 0x3E66DFD1. El comando:

pgp -e archivo.??? 0xaf32327d 0x3e66dfd1

cifrará archivo.??? para fulano y para mí.

Cuando examinamos con -kv un llavero vemos el número hexadecimal de la(s) llave(s).

Esta es la manera de evitar ambigüedades cuando varios tienen identificadores iguales. El caso extremo será si alguno ha generado dos pares de llaves con el mismo identificador pues es la misma persona. Pero el número hexadecimal de cada par de llaves es distinto.

Una observación a propósito de los identificadores. PGP maneja como identificador cualquier palabra o parte de ella que esté en el nombre de la llave. Mi llave es: pub 512/AF32327D 1994/07/15 Luis del Valle <lvalle@mixcoac.uia.mx>. Pueden ser identificadores las palabras completas: luis del valle lvalle mixcoac.uia.mx, o cualquier parte de ellas. Y por eso fácilmente se pueden dar ambigüedades. También puede ser identificador cualquier combinación de letras y espacios de mi nombre si se escriben entre comillas: "luis del" "is de" "lv", o también el número hexadecimal de mi llave, o todo o una parte de él, precedido por el prefijo 0x. 0xaf32327d, 0xaf32, 0x327d identifican mi llave, aunque si no se escribe todo el número puede identificar otras llaves también. La manera de ver si varias llaves coinciden en el mismo identificador es viendo el contenido de un llavero con la orden pgp -kv ident [llavero]. Resultarán todas las llaves que tengan ese identificador. Si sólo resulta una, no hay ambigüedad en ese identificador.

E. El archivo randseed.bin

Cada vez que PGP cifra algo genera primero la clave para ello. Es un número construido con un número de tantos bits cuantos tiene la llave. Con esa clave cifra el archivo. Luego cifra la clave recién generada con ayuda de la llave pública del destinatario. Al descifrar, PGP descifra primero la clave de todo lo demás del archivo y con ella como instrumento descifra el archivo mismo. Randseed.bin es un archivo que utiliza PGP para generar al azar el número que será la clave de lo que esté cifrando en ese momento.

F. Archivos por cifrar en otros directorios que no son \PGP

Sin duda que ya han cifrado archivos que no estén en el directorio \PGP. No es tema éste directamente del uso de este programa, sino del uso general del DOS. Dado que el directorio \PGP está en la ruta (path) en el comando path del autoexec.bat, se puede invocar el comando pgp desde cualquier directorio.

También se puede cifrar cualquier archivo que esté en cualquier directorio si estando en el directorio PGP se da el nombre del archivo con todo y su ruta. Por ejemplo:

c:\pgp> pgp -esa c:\doc\cartas\fulano.txt ident

cifra el archivo fulano.txt que está en el subdirectorio cartas del subdirectorio doc del directorio principal. Con el parámetro -o pedimos que el resultado tenga el nombre que queramos en el directorio que deseemos.

c:\pgp> pgp -esa c:\doc\cartas\fulano.txt ident -o c:\modem\subir\fulano.asc

pone el mensaje cifrado en el subdirectorio subir del subdirectorio modem del directorio principal.

X. Frentes para PGP en DOS y en windows

Tan sólo aludo a que existen programas que manejan más amigablemente, como dicen, el PGP. No es que se haya hecho el programa de esa otra manera, sino que son presentaciones por medio de menús, o desde windows y que se encargan da ir a DOS y dar los comandos con los parámetros correspondientes. Mi experiencia ha sido que facilitan el manejo, pero sólo de las principales funciones. Por otra parte son las más ordinarias.

Se pueden conseguir por ftp o en la página web del pgp. No entro a explicarlos porque sería, creo, una tarea semejante a la que asumí con este instructivo.

Órdenes para PGP 2.6.3(i)

(a_normal = archivo normal; adellaves = archivo de llaves,...)

Para cifrar un archivo normal con la llave pública del destinatario:
pgp -e a_normal identificador [otros identificadores]

Para firmar un archivo normal con mi llave secreta:
pgp -s a_normal [-u mi_identificador]

Para firmar un archivo normal con mi llave secreta y después cifrararlo con la llave pública del destinatario:
pgp -es a_normal identificador [otros id.] [-u mi_identificador]

Para cifrar un archivo normal sólo a la manera tradicional:
pgp -c a_normal

Para descifrar un archivo, o comprobar la integridad de la firma en un archivo firmado:
pgp acifrado [-o a_normal]

Para descifrar un archivo con armadura ASCII en varias partes: grabe todas las partes por orden en un archivo con extensión .asc y escriba:
pgp farmadura [-o a_normal]

Funciones para la gestión de llaves:

Para generar tu propio par único de llaves pública/secreta:
pgp -kg

Para añadir el contenido de un archivo de llaves públicas o secretas al llavero correspondiente:
pgp -ka adllaves{origen} [llavero{dest}]

Para extraer (copiar) una llave del llavero de llaves públicas o secretas:
pgp -kx identificador adllaves [llavero]

o: pgp -kxa identificador adllaves {destino} [llavero{origen}]

Para visualizar el contenido del llavero de llaves públicas:
pgp -kv[v] [identificador] [llavero]

Para ver la huella digital de una llave pública, y poder verificarla por
teléfono con su dueño:
pgp -kvc [identificador] [llavero]

Para visualizar el contenido y comprobar las firmas de certificación en el llavero de llaves públicas:
pgp -kc [identificador] [llavero]

Para modificar el identificador o la frase de contraseña (clave) de la llave secreta:
pgp -ke identificador [llavero]

Para modificar los parámetros de confianza de una llave pública:
pgp -ke identificador [llavero]

Para suprimir una llave, o sólo un identificador, del llavero de llaves públicas:
pgp -kr identificador [llavero]

Para firmar la llave pública de alguien en el propio llavero de llaves públicas:
pgp -ks identificador_des
[-u su_identificador] [llavero]

Para suprimir de un llavero ciertas firmas de un usuario:
pgp -krs identificador [llavero]

Para desactivar una llave pública, para que PGP no la utilice, o para reactivarla:
pgp -kd identificador [llavero]

Para revocar permanentemente su propia llave, emitiendo un certificado de compromiso de llave:
pgp -kd su_identificador

Ordenes esotéricas:

Para descifrar un mensaje y dejar su firma intacta:

pgp -d acifrado

Para crear un certificado de firma separado del documento:

pgp -sb atexto [-u mi_identificador]

Para separar un certificado de firma del mensaje firmado:

pgp -b acifrado

Opciones que pueden utilizarse en combinación con otras:

Para producir un archivo cifrado en formato ASCII radix-64, añada simplemente la opción -a cuando cifre o firme un mensaje, o cuando extraiga una llave:

pgp -sea ftexto identificador_des

o: pgp -kax identificador fdllave [llavero]

Para eliminar automáticamente el archivo normal después de crear el archivo cifrado, añada simplemente la opción -w (wipe {limpiar})
cuando cifre o firme un mensaje:

pgp -wes mensaje.txt identificador [otros ident.]

Para indicar que un archivo normal contiene texto ASCII, no binario, y que debe ajustarse a los convenios de línea del destinatario, añada la opción -t (texto) a las otras opciones:

pgp -seta mensaje.txt identificador [otros ident.]

Para visualizar el texto descifrado en la pantalla (como con la orden "more" {más} de Unix), sin grabarlo en un archivo, utilice la opción -m (more) al descifrar:

pgp -m acifrado

Para indicar que el texto descifrado para el destinatario SÓLO se puede mostrar en la pantalla de éste, sin grabarse en el disco, añada la opción -m:

pgp -metas mensaje.txt identificador [otros identificadores]

Para recuperar el nombre original del archivo normal al descifrar, añada la opción -p:

pgp -p acifrado

Para utilizar en modo filtro, como en Unix, leyendo de la entrada estándar y escribiendo en la salida estándar, añada la opción -f:

pgp -festa identificador [otros identificadores]< aentrada > asalida

Para incluir identificadores de un archivo al cifrar un mensaje, utiliza la opción -@:

pgp -e archivo un_identificador -@masids.txt

NOTA: Aunque no esté en la documentación, también me ha resultado con:

pgp -e archivo -@identificadores.txt

Como identificador se puede usar el número de cualquier llave (el que sale al ver un llavero, no la huella digital) precedido de 0x (cero, x), por ejemplo 0xAF323227D, o sólo parte de ese número. Cuidar que no haya ambigüedad. Se comprueba viendo el llavero con el identificador (pgp -kv identificador llavero) y comprobar que sólo resulta uno. A no ser que uno quiera cifrar para los varios que resulten.

Revisado el escrito noto que no traté lo que se dice del modo de filtro al estilo unix.

Órdenes para PGP 2.6.3i

Cifrar a la manera tradicional:

pgp -c[a][m][w] archivo.??? [-o ruta\archivo.???]

Opciones:

[a] armadura de ASCIIs,

[m] Sólo lectura en pantalla sin que lo puedan grabar.

[w] Que se borre el archivo original no cifrado

[o ruta\archivo.???] Darle nombre con todo y ruta al archivo resultante

Resulta: archivo.asc ó archivo.pgp ó ruta\archivo.???

Cifrar al modo del pgp

pgp -e[s][a][m][w] archivo.??? Ident [ident] [ident] [...][-o ruta\archivo.asc] [-u tu_ident],
o bien: pgp -e[s][a][m][w] archivo.??? -@llaves.txt [-o ruta\archivo.asc] [-u tu_ident],

Opciones:

- [s] firmar
- [a] armadura de ASCIIs
- [m] Solo lectura sin que lo puedan grabar
- [w] Que se borre el archivo original no cifrado
- [-o ruta\archivo.???] Darle nombre con todo y ruta al archivo resultante.
- [-u tu_identi] se debe usar si firmas lo cifrado y tienes en tu secring.pgp varias llaves.

Llaves.txt es un archivo de texto con los identificadores de aquellos para quien se cifra. Un identificador en cada renglon. Pueden ir espacios y sin comillas.

Resulta: archivo.asc ó archivo.pgp ó ruta\archivo.???

Firmar un texto abierto (se puede firmar también cualquier archivo, con cualquier terminación):

pgp -st[a][b] archivo.txt

Opciones:

- [a] armadura de ASCIIs
- [b[producir la firma en archivo separado

Resulta: archivo.asc ó archivo.pgp

Descifrar:

pgp -[m][p] archivo.??? [-o ruta\archivo.???]

Opciones:

- [m] Sólo lectura, sin que se grabe el archivo descifrado en mi disco.
- [p] El resultado tendrá el nombre original del archivo que se cifró.
- [-o ruta\archivo.???] Dará nombre con todo y ruta al archivo que resulte

Observaciones:

PGP va distinguiendo qué tipo de archivos está descifrando.

- ♦ una o varias llaves. Preguntará si las añade al llavero pubring.pgp.
- ♦ un texto cifrado a la manera tradicional. Pedirá la frase-clave con que se cifró.
- ♦ un texto cifrado la manera del pgp. Dirá si está firmado y luego lo descifrará.
- ♦ una firma separada del texto. Dirá que necesita el nombre del archivo a que se refiere y luego checará la integridad del documento.
- ♦ un texto abierto pero firmado. Checará la firma.

Resultados diversos.

* Recuérdese que el config.txt puede hacer que las opciones [a][t] estén activadas siempre que sea posible aunque no se escriban expresamente en la línea de comando.

MANEJAR LLAVES

- Generar nuestro par de llaves pública-privada

  `pgp -kg`

  El resultado son pubring.pgp, secring.pgp y randseed.bin

- Añadir llaves nuevas a un llavero:

  `pgp -ka archivo_origen [llavero_destino]`

  Si no se escribe el llavero_destino va al defaul, que es pubring.pgp, o el que se haya indicado en el config.txt, o el secring.pgp según que el contenido del archivo_origen tenga por contenido llaves públicas o privadas

- Meter nuestro llavero público, con una o con muchas llaves, a una armadura de ASCIIs simples con la que es capaz de sobrevivir en el medio inhóspito de las máquinas de 7 bits.

  `pgp -a pubring.pgp`

  El resultado será pubring.asc, que no es llavero, sino archivo de llaves. Es la colección de llaves en formato .asc, pero no funciona como llavero.

- Sacar copia de cualquier llave que esté en nuestro llavero público (e igual se hace con el llavero privado o secreto) y conservarla en un archivo cuyo contenido es sólo una llave.

  ```
  pgp -kx ident archivo_destino.pgp llavero_origen
  pgp -kxa ident archivo_destino.asc llavero_origen
  ```

- Examinar cualquier llavero para saber de qué llaves consta. La segunda "v" opcional se usa para ver también qué firmas tiene cada llave (v=verbose). Si no se escribe ningún ident se verán todas.

  `pgp -kv[v] [identif] [llavero]`

- Suprimir una llave o sólo un identificador, de un llavero

  `pgp -kr ident [llavero]`

- Añadir un identificador o modificar la clave en el llavero privado

  `pgp -ke ident [llavero]`

- Firmar la llave de un llavero público

  `pgp -ks ident [-u propio_ident] [llavero]`

  El parámetro -u propio_ident sirve para cuando en el mismo llavero privado hay varias llaves. O porque uno mismo se ha generado varias llaves, o porque dos o más comparten el mismo llavero secreto, cada uno con su llave.

- Examinar la huella digital de las llaves

  `pgp -kvc [ident] [llavero]`

- Modificar el valor de mi firma en la llave de otro, presente en un llavero público mío.

  `pgp -ke ident [llavero]`

- Suprimir una o más firmas de una llave.

  `pgp -krs ident [llavero]`

- Desactivar (o activar si estaba desactivada) una llave pública.

  `pgp -kd ident [llavero]`

- Revocar una llave propia.

  `pgp -kd mi_ident`

  Se emite un certificado de revocación. Este certificado, incluido en un llavero público, impide que se use la llave revocada. Esto no lo he hecho nunca. No sé si al mismo tiempo queda desactivada para siempre la propia llave.