```
I-gnupg: (0.18) (potato)
        miniGuia de instalación/uso de GnuPG: firma y cifrado de clave pública


CAMBIOS:
14/1/01 0.13
 1/2/01 0.14 mutt de potato rev2
17/6/01 0.15 firmar en archivo separado
 5/7/01 0.16 firma de claves, añadir nueva dir correo (identidad), teclas mutt
 6/7/01 0.17 dudas resueltas tras tener la clave firmada
11/9/01 0.18 importar clave desde mutt


ATENCIÓN:
8/2000 No usar claves DH son susceptibles de manipulación (aviso de kriptopolis)


ANTES DE EMPEZAR:
    Mira si hay una actualización de seguridad más reciente del paquete gnupg
    en Debian (Potato trae 1.0.1 y ya existe por lo menos el 1.0.4-1)
    Cuando trabajas con criptografía es necesario estar pendiente de esas
    actualizaciones y estar al día


PASO A PASO:

1) Instalar paquetes:
    gnupg
    gpgp        Frontend para gnome

2) Instalar paquetes (solo si se desea compatibilidad con PGP 2.0 y ojo a las
 patentes... parece que solo es posible el uso "no comercial" de IDEA)

    gpg-rsa     Modulo RSA para GPG (algoritmo expatentado) (está en non-us)
    gpg-idea    Modulo IDEA para GPG (algoritmo patentado)

    _NOTA_: La patente de RSA expiró el 20 de Septiembre de 2000 y actualmente
    y viene con gnupg "de serie"


    Añadir a ~/.gnupg/options   *POR VERIFICAR QUE FUNCIONA*

    ---8<---
    # PGP2 (IDEA)
    load-extension idea
    #load-extension rsa         # Actualmente parece no necesario
    --->8---


3) Inicializar gnupg (dos opciones)
    a) Lanzar el frontend  gpgp
    b) Lanzar gpg --gen-key     # Si no existe ~/.gnupg/* solo los crea


4) Crear una clave
    gpg --gen-key
        1       (1) DSA y ElGamal (tipo de clave deseado)
        1024    ¿De qué tamaño quiere la clave?
        0       ¿Validez de la clave (0)? --- 0 = la clave nunca caduca ---
        s       ¿Es correcto (s/n)?
        NOMBRE APELLIDOS        Nombre y apellidos:
        EMAIL@PROVEEDOR.ES      Dirección de correo electrónico (permanente):
        [enter]                 Comentario:
        v                       ¿Cambia ... o (V)ale/(S)alir?
        fRasE d5e pas$O         Introduzca contraseña (para proteger tu clave)
        fRasE d5e pas$O         Repita contraseña:

            *** Usar el teclado ratón y disco para conseguir suficiente ***
            *** entropía para generar los bytes aleatorios necesarios   ***


5) Generar certificado de revocación y guardarlo en floppy e impreso "lejos"
    (por si olvidas la contraseña o te roban la clave)
    y no dejar copia en este sistema (para evitar que un intruso te lo revoque
    tras conseguir comprometer tu sistema)

    gpg --gen-revoke tu_identidad_de_usuario
        # Envía el certificado a la pantalla, corta y pega e imprime

    shred -fuz archivo          # Borrado seguro del archivo utilizado
        # shred = hacer trozos pequeños (sobreescribe 25 veces ;-)
        # -f = cambiar permisos para poder escribir si es preciso
        # -z = sobreescritura final con ceros para ocultar la operación
        # -u = borrar el archivo al final


6) Copia a floppy también (copia de seguridad, ante fallos de disco):

        ~/.gnupg/secring.gpg    #Aqui está tu clave privada (protegida)
        ~/.gnupg/pubring.gpg    #Aqui están las claves públicas
        ~/.gnupg/trustdb.gpg    #Base de datos de confianza


7) Exportar clave para enviar por correo o ponerla en pagina web

    gpg --armor --export id_clave > mi-clave-publica-en-formato-ASCII
        # sin --armor se exporta en formato binario

    gpg --armor --export > todas-mis-claves-publicas-en-formato-ASCII


8) Envia tu clave a un servidor de claves públicas en Internet

    gpg --keyserver certserver.pgp.com --send-key Manel

    NOTA: Asegurate de que tu cortafuegos te permite trafico hacia el puerto
        de destino 11371

*POR ACLARAR*
    ¿Otros servidores de claves?
        pgp.rediris.es
        wwwkeys.nl.pgp.net
        pgp.uoc.es


9) Añade a tu firma de correo y/o a tu página web personal el identificador de
 tu clave y el fingerprint

    El identificador es necesario para que consigan tu clave del servidor
    El fingerprint es preciso para verificar que es realmente tu clave

    Ejemplo:
        gpg --fingerprint Manel

        pub  1024D/F9BC34B5 2000-12-07 Manel Marin <manel3@apdo.com>
        Key fingerprint = 2F60 43D5 A297 5458 9067  5A50 0029 9C8D F9BC 34B5
        sub  1024g/C896EE40 2000-12-07


    El identificador (keyID) es: F9BC34B5
    El fingerprint es: 2F60 43D5 A297 5458 9067  5A50 0029 9C8D F9BC 34B5
```

```
10) Importar las claves públicas de los desarrolladores de Debian

    Instala el paquete debian-keyring

    gpg --import /usr/share/keyrings/debian-keyring.gpg


11) Para importar/firmar la clave pública de un amigo

    ATENCIÓN: Es VITAL asegurarse que la clave es realmente la de tu amigo, y
     no ha sido sustituida por un tercero

    ATENCIÓN: Se firman las identidades (Nombre-Correo) de la firma así que
     si cambias de correo (añades una identidad) hazlo antes de que te firmen,
     o te tedrán que firmar esa identidad después


  SISTEMAS:
    Tienes que ver físicamente un documento identificativo (DNI, pasaporte) y:


    1) Intercambiar keyIDs y fingerprints en mano
        gpg --fingerprint id_usuario | lpr


    2a) Tu amigo te envia su clave publica adjunta por correo

        gpg --import archivo.asc        # Esto importa la clave pública


    2b) O utilizar el servidor de claves publicas de Internet
        ***Necesitas el keyID de la clave de tu amigo***

        NOTA: Asegurate de que tu cortafuegos te permite tráfico hacia el
        puerto de destino 11371

        gpg --keyserver certserver.pgp.com --recv-key F9BC34B5

        gpg --sign-key id_del_amigo # Esto firma la clave de nuestro amigo
                                    # con la nuestra, muestra el fingerprint,
                                    # las identidades y...
        ¿Firmar realmente todos los identificativos de usuario?
        ¿Firmar de verdad?


    3) Le devuelves la clave firmada a tu amigo, para que el la importe y
        añada así tu firma a su clave


    4) Tu amigo puede ver las firmas añadidas a su clave con:

        gpg --edit-key keyID
        > check         # Muestra todas las identidades y las firmas que tienen

        > quit          # Esto para salir...

        Deberá también:
        - Hacer copia de seguridad de la clave (punto 6)
        - Volver a subir la clave al servidor de claves (punto 8)
        - No necesita generar otro certificado de revocación, vale el anterior


POSTERIORMENTE:

1) Añadir una nueva dirección de correo (identidad)

    ATENCION: Si tu clave ya estaba firmada te tendrán que firmar la nueva
     identidad para que tenga confianza

    gpg --edit-key id_usuario
    > adduid
    Nombre y apellidos:
    Dirección de correo electrónico:
    Comentario:
    ¿Cambia (N)ombre, (C)omentario, (D)irección o (V)ale/(S)alir? -> V
    Introduzca contraseña:
    > save

    - Tienes que volver a subir tu clave al servidor de claves


MANIOBRAS VARIAS:

    Para cifrar/descifrar y firmar/verificar es más práctico usar mutt o el
    frontend gpgp...

    Lo importante es entender que:
        - Para que solo el destinatario pueda descifrar, cifras con _SU_
         clave pública
        - Para firmar utilizas _TU_ clave privada, se podrá verificar por
         cualquiera que disponga de tu clave pública


  LINEA DE COMANDO:
    gpg --list-keys     # Muestra lista de claves públicas o solo las pedidas

    gpg -sa archivo     # Firmar archivo en ASCII para enviar por correo
    gpg --verify archivo                # Verificar archivo firmado
    gpg -ea -r id_usuario archivo       # Cifrar en archivo.gpg en ASCII
    gpg --clearsign archivo             # Firmar en archivo.asc en ASCII
    gpg --decrypt archivo               # Descifrar


    gpg -sb archivo                     # Firmar en archivo.sig separado
    gpg -sba archivo                    # Firmar en archivo.asc ASCII separado
    gpg --verify archivo.asc            # Verificar los archivos firmado
                                        #  con adjunto (no vale *.asc ...)
        NOTA: El comando devuelve 0 si la firma es correcta
                y 1 si _NO_ lo es


    gpg --edit-key id_usuario           # Gestión de claves
    gpg --print-md '*' archivo          # Muestra MD5, SHA1 y RMD160 de archivo



  GPGP:
    gpg de Potato está en una versión muy baja y aunque es práctico para
    cifrar, descifrar y firmar tiene algunas limitaciones:

        - La gestión de claves debe de hacerse con la linea de comandos
        - Hay muchas opciones y botones que no funcionan
        - No da la opción de cifrar si no has firmado la clave pública del
         destinatario (gpg -ea avisa y pide confirmación)


USO DESDE CLIENTE DE CORREO:

    Hay varios clientes de correo integrables con gpg (mutt y pine)

    Netscape parece haber elegido otro sistema, certificados B2 (creo) de los
```

que tienes que pagar cada año...

Esto no impide que puedas escribir algo en gpgp, cifrarlo, cortarlo, pegarlo en el Messenger y enviarlo por correo y al revés ;-)

MUTT:

Teclas ya disponibles:

"p" = Menu GPG (co(d)ificar, f(i)rmar...)
"Esc + k" = Adjuntar una clave pública como añadido (pide keyID)

Para importar una clave pública adjunta:

"Ctrl + K" = Extraer claves públicas

Otro sistema:
- salva el adjunto como archivo p.e. archivo-x
- gpg --import archivo-x

Para firma en el cuerpo (en lugar de en un archivo adjunto) añadir a ~/.muttrc

```
---8<---

# Configuración para usar GnuPG con mutt
##NOTA: todas las lineas con ## no son necesarias (y dan error) con potato rev2
##set pgp_default_version=gpg
##set pgp_key_version=default
##set pgp_receive_version=default
##set pgp_send_version=default

set pgp_sign_micalg=pgp-sha1
##set pgp_gpg=/usr/bin/gpg


# CTRL+F para firmar en el mismo cuerpo sin usar adjuntos PGP/MIME
# el "s" es para que no pregunte si desea sobreescribir /tmp/mutt-host-XXXX-Y
macro   compose \CF    "Fgpg --clearsign\ns"

# CTRL+E para cifrar en el mismo cuerpo sin usar adjuntos PGP/MIME
# el "s" es para que no pregunte si desea sobreescribir /tmp/mutt-host-XXXX-Y
macro   compose \CE    "Fgpg -ea\ns"

# CTRL+V para verificar la firma sin usar adjuntos PGP/MIME
macro   pager \CV    "|gpg --verify\n"

# CTRL+D para descifrar sin usar adjuntos PGP/MIME
# uso dd porque la primera d no aparece (borra la siguiente letra si no es d)
macro   pager \CD    "|gpg --ddecrypt\n"

--->8---
```

FALLOS AL VERIFICAR UN ARCHIVO FIRMADO

Para que la firma sea correctamente verificada es necesario que no se haya cambiado ni una sola letra.

Si usas Netscape Mail y cortas y pegas en gpgp piensa que los tabuladores son sustituidos por espacios, con lo que la verificación dará INCORRECTA

Para sortear esto salva el correo como un archivo (ALT+S) y haz

```
gpg --verify archivo            # Verificar archivo firmado
```

POR DENTRO:

Esto que sigue es una simplificación, puedes profundizar más en el handbook en castellano en http://www.gnupg.org/docs.html

- GnuPG es conforme al estándar propuesto OpenPGP descrito en RFC2440
- Está libre de algoritmos patentados y es software libre

CLAVE PRIVADA, CLAVE PUBLICA ¿COMO FUNCIONA?

Hay dos claves, una privada y la otra pública, y son "complementarias": utilizando nuestra clave privada podemos descifrar lo que otros han cifrado con nuestra clave pública.

Como efecto lateral _NO_ podemos descifrar un mensaje que acabamos de cifrar con la clave pública de otro, ya que para eso se necesita la clave privada.

La base del invento es que no se puede conseguir la clave privada a partir de la pública :-)

FIRMAR CLAVES

Firmar las claves públicas de otros usuarios es la forma de demostrar que estamos absolutamente seguros de que su propietario es quien dice ser

Cuantas mas firmas de personas tengas en tu clave, mayor es tu circulo de confianza

Cada dirección de correo (identidad) de la firma va firmada por separado

MIME/PGP

Hay un estándar para añadir las firmas y los mensajes cifrados como adjuntos MIME del tipo "application/pgp-signature", si el agente de correo receptor no lo soporta (Windows, Netscape) se puede enviar firmado/cifrado en el cuerpo del mensaje

CONFIANZA

Parece que gnupg evalúa una "cadena" de confianza basada en las claves firmadas que poseemos para decidir si una clave pública merece confianza o no (si podemos estar seguros de que realmente pertenece a quien creemos)

PGP/GnuPG Y PATENTES:

PGP 2.X usa los algoritmos patentados RSA e IDEA

MAS INFO:

http://www.gnupg.org/docs.html	Hay handbook y miniCOMO en castellano
man gpg	
/usr/doc/gnupg/README.gz	
/usr/doc/gnupg/FAQ.gz	Incluye interacción con PGP2 y PGP5
http://www.kriptopolis.org	Hay un tutorial "GnuPG en una hora"
Mutt-GnuPG-PGP-HOWTO	Viene con Potato en /usr/doc/HOWTO
/usr/doc/mutt/html/manual.html	