Mutt-i, GnuPG y PGP Como

Andrés Seco *AndresSH@ctv.es* y J.Horacio M.G. *homega@ciberia.es* v1.2, 7 de Febrero de 2000

Este documento explica la forma de configurar *Mutt-i*, *PGP* y *GnuPG* en sus diferentes versiones (2.6.x, 5.x y GnuPG) de forma rápida indicando los posibles problemas que pueden surgir al enviar correo firmado o cifrado para ser leído por clientes de correo que no se ajustan a las especificaciones PGP/MIME definidas en la RFC2015 y por otros sistemas operativos. Incluye además un ejemplo de configuración de procmail para enviar las claves públicas de forma automática a peticiones recibidas por correo electrónico, como si de un servidor de claves se tratara.

Índice General

1 Introducción

Este documento explica la forma de configurar *Mutt-i* y *PGP* en sus versiones `2.6.x`, `5.x` y `GnuPG` para poder tener en marcha de forma rápida un lector de correo con seguridad de firmas y cifrado digital.

Para esto se incluirán ficheros de configuración de ejemplo que servirán para la puesta en marcha. Para obtener el máximo rendimiento y usar todas las características de los programas que se utilizarán será necesario leer la documentación adjunta a cada programa y reconfigurar dichos ficheros de configuración que se incluyen como ejemplos.

Además, se comentarán algunos problemas derivados de la falta de seguimiento de la recomendación RFC2015 sobre PGP/MIME por parte de muchos programas de correo electrónico, tanto en Linux como en otros sistemas operativos.

Será presentado también un ejemplo de configuración de procmail de forma que puedan ser enviadas las claves públicas de manera automática a la recepción de mensajes que las soliciten.

Nos gustaría dar las gracias a Roland Rosenfeld ***roland@spinnaker.de***, Christophe Pernod ***xtof.pernod@wanadoo.fr***, Denis Alan Hainsworth ***denis@cs.brandeis.edu*** y Angel Carrasco ***acarrasco@jet.es*** por sus correcciones y sugerencias.

2 Copyright y descarga de responsabilidad

Por supuesto, estamos abiertos a todo tipo de sugerencias y correcciones sobre el contenido de este documento.

3 Nota sobre el intercambio de correo desde/hacia internet

Este documento no trata sobre el intercambio de mensajes de correo electrónico entre el equipo local y otros equipos (en red local o en internet). Dicho intercambio debe ser realizado mediante programas que actúen como Agentes de Transferencia de Mensajes (MTAs) como Sendmail, *http://www.sendmail.org*, qmail, *http://www.es.qmail.org*, Exim, *http://www.exim.org*, Smail, *ftp://ftp.planix.com/pub/Smail*, etc.

En este documento se presupone que este método de envío/recepción de mensajes fuera del equipo local ya está instalado y funcionando de forma correcta. Si puede enviar un mensaje y leer su correo con el comando `mail` desde el indicador de linea de comando de su equipo,

```
$ mail -s <asunto> <usuario@dominio.net>
escribir aquí el texto, y finalizar con un punto en la línea siguiente
.
```

es que debes tener instalado algún tipo de MTA que se encargue de la transferencia de mensajes. En caso contrario, puedes encontrar documentación al respecto en las páginas de manual de *smail*:

```
$ man smail
```

o del MTA que tengas instalado, y las de *fetchmail*:

```
$ man fetchmail
```

o en otro documento similar a estos que haga referencia a dichos programas.

4 Configuración de mutt

El siguiente fichero de ejemplo es válido para comenzar a usar *Mutt* de forma básica incluyendo caminos para los ficheros de alias, mensajes enviados y mensajes postpuestos. Se puede realizar una mayor personalización atendiendo a las indicaciones del manual de *Mutt* en `/usr/doc/mutt/` o `/usr/doc/mutt-i/`

Ejemplo simple de `~/.muttrc`:

```
set folder=~/Mail
set alias_file=.alias
set postponed=.postponed
set record=MensajesEnviados
set signature=.signature
my_hdr From: Nombre Apellido <Nombre@dominio.com>
source =.alias
```

Es necesario que exista el directorio `~/Mail`, que es el que aparece como un signo de «igual que» en el fichero de configuración `.muttrc` (esto es, `=.alias` quiere decir para *Mutt* `~/Mail/.alias`, y `=.postponed` quiere decir para *Mutt* `~/Mail/.postponed`). No obstante es posible tener estos ficheros en otro directorio siempre y cuando indiquemos el camino completo en `~/.muttrc`, y tengamos los permisos necesarios para trabajar en dicho directorio.

También hay que personalizar la linea `my_hdr` con el nombre y la dirección de correo electrónico adecuados. En el fichero `~/Mail/.signature` se puede incluir la firma que aparecerá en todos los mensajes que se envíen.

Este fichero de configuracion puede llegar a hacerse muy grande, por lo que es común separar algunos de sus comandos en ficheros diferentes. Por lo pronto, son facilmente separables las lineas de configuración de *PGP* o *GnuPG* y las macros de teclado que personalicemos. Para ello seria necesario añadir las lineas siguientes al fichero `~/.muttrc`:

```
source = ~/Mail/.mutt.macros
source = ~/Mail/.gnupgp.mutt
```

y utilizar los ficheros `~/Mail/.mutt.macros` y `~/Mail/.gnupgp.mutt` para introducir en ellos las macros de teclado y la configuración de *PGP* o *GnuPG* que más adelante se comentan.

Para una información más extensiva y completa sobre el uso y configuración de *Mutt*, así como de características avanzadas, ver el Manual de Mutt en ***http://www.lucas.org/manual_mutt.html***.

5 PGP y GnuPG

Para usar cualquiera de las versiones de *PGP* con *Mutt-i*, primero será necesario configurar adecuadamente *PGP* de modo que existan el fichero con las claves públicas (anillo de claves públicas) y el fichero con las claves privadas (anillo de claves privadas). Conviene probar previamente PGP desde la línea de comandos para asegurarse de que firma y cifra correctamente.

Recordemos que para *Unix* las versiones de *PGP* existentes son `2.6.3(i)` y `5.0(i)`, que de ahora en adelante llamaremos **PGP2** y **PGP5** respectivamente. **GnuPG** es un sistema de codificación nuevo, todavía en desarrollo aunque muy avanzado, de código abierto y libre, en muchos aspectos superior a **PGP** (ver GnuPG mini como en ***http://www.insflug.org/documentos/GPG-Mini-Como/***).

Aclararemos también que *PGP*, al ser un sistema desarrollado en los EE.UU., está sujeto a ciertas leyes sobre la exportación de programas que incluyen código criptográfico; por esta razón existe una versión internacional para casi todas las versiones numéricas, y ésta vienen denotadas por la letra «**i**» (**pgp - pgpi**).

5.1 PGP2

PGP2 genera claves con el algoritmo RSA (***http://www.rsa.com***), y como algoritmo de cifrado usa IDEA (***http://www.ascom.ch***).

Ambos son algoritmos propietarios y su uso está restringido por sus respectivas patentes.

Para su correcto funcionamiento, una vez instalado el programa, deberemos tener un directorio `~/.pgp`, en el que se encuentren el fichero de configuración `pgp-i.conf` y los ficheros con los anillos de claves públicas y privadas, `pubring.pgp` y `secring.pgp` respectivamente.

5.2 PGP5

Las claves generadas por *PGP5* son del tipo **DSS/DH** (Digital Signature Standard / Diffie-Helman). PGP5 usa como algoritmos de cifrado **CAST**, **Triple-DES**, e **IDEA**. PGP5 puede trabajar con datos cifrados y/o firmados con *RSA* (PGP2), e incluso usar estas claves para cifrar y/o firmar digitalmente (con claves generadas por PGP2, ya que PGP5 no puede generar dichas claves). Por contra, PGP2 no reconoce las claves *DSS/DH* de PGP5; esto crea un problema de incompatibilidad, pues en *Unix/Linux* todavía muchos usuarios tienen instalado sólo la versión de PGP2.

Para el correcto funcionamiento de PGP5, en el directorio `~/.pgp`, se encontrarán los anillos de claves públicas y privadas (`pubring.pkr` y `secring.skr` respectivamente), así como el fichero de configuración `pgp.cfg`.

En caso de tener instaladas las dos versiones de *PGP* (por tanto, habremos instalado y configurado PGP2 antes que PGP5), el fichero de configuración `~/.pgp/pgp.cfg` de PGP5 lo crearemos como un enlace simbólico al fichero de configuración `~/.pgp/pgp-i.conf`,

```
~/.pgp$ ln -s pgp-i.conf pgp.cfg
```

añadiendo las siguientes líneas al final del fichero `~/.pgp/pgp-i.conf`:

```
PubRing = "~/.pgp/pubring.pkr"
SecRing = "~/.pgp/secring.skr"
RandSeed = "~/.pgp/randseed.bin"
```

Los ficheros con los anillos de claves de las diferentes versiones pueden coexistir sin ningún problema en el mismo directorio.

5.3 GnuPG

GnuPG es un programa con las mismas funciones que el anterior. A diferencia de *PGP*, *GnuPG* evita el uso de algoritmos con patentes restrictivas. *PGP* puede ser usado libremente con fines personales, pero no comerciales, y su desarrollo es cerrado. *GnuPG* es de libre uso, y de desarrollo abierto, al igual que nuestro sistema operativo favorito (además de que su implementación y desarrollo es principalmente en *Linux*).

Las claves generadas por *GnuPG* son del tipo **DSA/ElGamal** (*Digital Signature Algorithm*, también conocido como *DSS*). Es totalmente compatible con *PGP*, excepto en el uso de los algoritmos con patentes restrictivas *RSA* e *IDEA*. No obstante, es posible implementar cierta compatibilidad al respecto (ver GnuPG mini como en *http://www.insflug.org/documentos/GPG-Mini-Como/* para su configuración e interacción con PGP2 y PGP5).

6 Integración de PGP y Mutt

La operación a realizar en los mensajes salientes (firmar, cifrar o ambas) se elige justo antes de pulsar «`y`» para enviar el mensaje, en el menú de opciones que aparece con la opción «`p`». Una vez elegida la operación a realizar simplemente cambiará la linea de *PGP* en la cabecera que se muestra en pantalla, pero hasta que no enviemos el mensaje con «`y`» no se nos pedirá introducir la frase de paso para activar la firma del mensaje o las claves publicas a utilizar para el cifrado en caso de que no coincida algún destinatario con los que tienen clave pública en nuestro anillo de claves.

NOTA: En caso de que cometamos un error al introducir la frase de paso cuando nos es solicitada, parecerá que *Mutt* se queda «colgado», pero no será así, si no que estará esperando que la volvamos a introducir. Para ello pulsaremos la tecla <`Intro`> y borraremos la contraseña de la memoria con el juego de teclas <`Ctrl`>`F`. A continuación repetiremos la operación de enviar el mensaje («`y`») y tendremos que volver a introducir la contraseña.

Mutt, en este caso, usará *PGP/MIME* para enviar el mensaje, de modo que aparecerá un nuevo adjunto al mensaje con la firma (si sólo hemos elegido firmar) o cifrará el mensaje completo (todas sus partes *MIME*, adjuntos incluidos) y dejará tan solo 2 partes MIME, la primera con la versión de PGP/MIME y la segunda con el mensaje total (adjuntos incluidos) cifrado (y firmado si así lo solicitamos).

Nota: Por varios motivos, como que el lector de correo del destinatario del mensaje no sea capaz de reconocer el formato *MIME*, podemos necesitar que la firma no vaya acompañando al mensaje como un adjunto, sino que aparezca en el mismo cuerpo del mensaje. Para este modo, ver la sección que trata sobre *application/pgp* con 7.1 (PGP5) y con 7.2 (GnuPG).

En los mensajes entrantes que vengan firmados o cifrados con *PGP/MIME*, *Mutt* intentará verificar la firma y/o descifrarlo automáticamente. Ver la sección sobre 8.2 (recetas para procmail), en donde se comenta cómo cambiar de forma automática el tipo *MIME* para los mensajes recibidos cifrados o firmados, que no indican su tipo *MIME* correctamente.

6.1 Ficheros de configuración opcionales

En las siguientes secciones se habla sobre las modificaciones al fichero de configuración de *Mutt* para poder usar 6.3 (PGP2), 6.4 (PGP5), y 6.5 (GnuPG) con comodidad.

Para ello se usa un fichero de configuración llamaremos `.gnupgp.mutt` (el nombre es inventado y le podríamos llamar de cualquier otro modo, siempre y cuando **indiquemos el origen** de este fichero de configuración en el fichero de configuración principal `~/.muttrc`.

Esto se hace incluyendo el origen (dónde está situado) completo del fichero de configuración `.gnupgp.mutt`, en una línea al final del fichero `~/.muttrc`. El directorio en el que situemos éste u otros ficheros opcionales de configuración puede ser cualquiera, siempre y cuando tengamos los permisos correctos de acceso a él (en un capítulo anterior lo incluimos dentro del directorio `~/Mail/`), o podemos crear un directorio dentro de nuestro directorio de usuario, con un nombre aleatorio:

```
~$ mkdir mutt.varios
```

en el que copiaremos (o crearemos) el fichero de configuración opcional `.gnupgp.mutt`, y a continuación indicaremos el origen en el fichero `.muttrc` con el comando `source`, del siguiente modo:

```
source ~/mutt.varios/.gnupgp.mutt
```

De este modo *Mutt* aceptará las variables de configuración en `.gnupgp.mutt` como si estuvieran directamente especificadas en `.muttrc`.

Este sistema es útil para evitar tener un fichero de configuración desordenado y demasiado grande, y puede ser usado para poner cualquier otro grupo de variables de configuración en otro fichero aparte. Por ejemplo, de modo parecido aunque no igual, si usamos *vim* como el editor por defecto en *Mutt*, le podemos indicar a `.muttrc` que use un fichero de configuración `.vimrc` distinto al que usamos cuando usemos *vim* sobre la línea de comandos. Para ello, copiaremos primero `~/.vimrc` a nuestro directorio ficticio para ficheros de configuración opcionales `~/mutt.varios/` y le daremos un nombre distinto (vg. `vim.mutt`):

```
$ cd /home/usuario
~$ cp .vimrc mutt.varios/vim.mutt
```

a continuación cambiaremos las variables de configuración que deseamos que sean distintas en *vim* como editor de *Mutt*, y finalmente modificaremos `.muttrc` para que refleje este cambio:

```
set editor="/usr/bin/vim -u ~/mutt.varios/vim.mutt"
```

Con esta última línea estamos indicando a `Mutt` que, en lugar de usar su editor interno por defecto, use un editor externo, *Vim*, con las opciones de configuración que deseamos.

6.2 Variables de Configuración General

Hay un número de variables que nos servirán para el uso de cualquiera de los tres sistemas de cifrado público con *Mutt* por igual. Éstas variables son del tipo booleano, y aceptan las opciones **set** (activada) o **unset** (desactivada).

En el fichero de configuración (bien sea `~/.muttrc`, o `~/mutt.varios/.gnupgp.mutt`, o el que decidamos), el signo de almohadilla (#) es un comentario libre y no se interpreta. Por tanto, lo usaremos aquí delante de la aclaración a cada una de las variables:

unset pgp_autosign

si esta variable está activada, *Mutt* nos pedirá firmar todo el
correo saliente. 6.2 ((1))

unset pgp_autoencrypt

si esta variable está activada, *Mutt* nos pedirá cifrar todo el
correo saliente. 6.2 ((1))

set pgp_encryptself

guardar una copia cifrada de todos los mensajes que se envíen cifrados
(precisa de la variable de configuración general `set copy=yes`).

set pgp_replysign

al responder a un mensaje firmado, requerir que el mensaje de respuesta
sea también firmado.

set pgp_replyencrypt

al responder a un mensaje cifrado, requerir que nuestra respuesta
también vaya cifrada.

set pgp_verify_sig=yes

¿queremos que se verifique automáticamente las firmas de los mensajes
entrantes? ¡por supuesto que sí!

set pgp_timeout=<n>

eliminar la contraseña de la memoria intermedia cada <n>
segundos.6.2 ((2))

set pgp_sign_as="0xABC123D4"

¿qué clave quiero usar por defecto para firmar los mensajes salientes?
Nota: es posible especificar un id de usuario en lugar de un id de
clave, pero esto podría confundir si tenemos el mismo id de usuario
para distintas claves.

set pgp_strict_enc

usar codificación «quoted-printable» siempre que PGP la
requiera.

unset pgp_long_ids

no usar identificadores de claves de 64 bits, usar de 32 bits.

set pgp_sign_micalg=<algo>

algoritmo de comprobación de la integridad de un mensaje, en donde
<algo> es uno de los siguientes:6.2 ((3))

- **pgp-mda5**
  para claves RSA
- **pgp-sha1**
  para claves DSS (DSA)
- **pgp-rmd160**

En las tres subsecciones siguientes se especificarán las variables a configurar para cada una de las versiones. La cuarta subsección explica los cambios en las variables en caso de que usemos más de una versión.

(1) dado que el contínuo requerimiento por parte de *Mutt* para que firmemos o cifremos todos los mensajes salientes puede representar un inconveniente, es deseable dejar esta variable desactivada. Esto es especialmente así en el caso del cifrado, ya que no dispondremos de las claves públicas de todos los destinatarios.

(2) depende del número de mensajes que firmemos o descifremos generalmente, nos interesará mantener la contraseña en la memoria durante más o menos tiempo. Esta opción nos evita que tengamos que introducir la contraseña por cada mensaje que firmemos, o por cada mensaje cifrado que leamos. **Aviso:** mantener la contraseña en la memoria es inseguro, especialmente en sistemas conectados a una red.

(3) esto sólo es necesario para la clave con la que hayamos configurado para firmar. Cuando la clave la escojamos desde el menú de composición, *Mutt* se encargará de calcular el algoritmo.

6.3 Variables de Configuración para PGP2

Para utilizar PGP2 con *Mutt-i* es necesario añadir las siguientes líneas al fichero `~/mutt.varios/.gnupgp.mutt`:

```
set pgp_default_version=pgp2
set pgp_key_version=default
set pgp_receive_version=default
set pgp_send_version=default
set pgp_sign_micalg=pgp-md5
set pgp_v2=/usr/bin/pgp
set pgp_v2_pubring="/.pgp/pubring.pgp
set pgp_v2_secring="/.pgp/secring.pgp
```

Evidentemente, deberán existir los ficheros `~/.pgp/pubring.pgp` y `secring.pgp`. Más información sobre PGP2 con el comando `man pgp`.

6.4 Variables de Configuración para PGP5

Para utilizar PGP5 con *Mutt-i* es necesario añadir las siguientes líneas al fichero `~/mutt.varios/.gnupgp.mutt`:

```
set pgp_default_version=pgp5
set pgp_key_version=default
set pgp_receive_version=default
set pgp_send_version=default
set pgp_sign_micalg=pgp-sha1
set pgp_v5=/usr/bin/pgp
set pgp_v5_pubring="/.pgp/pubring.pkr
set pgp_v5_secring="/.pgp/secring.skr
```

Evidentemente, deberán existir los ficheros `~/.pgp/pubring.pkr` y `secring.pkr`. Más información sobre PGP 5 con el comando `man pgp5`.

6.5 Variables de Configuración para GnuPG

Para utilizar *GnuPG* con *Mutt-i* es necesario añadir las siguientes líneas al fichero `~/mutt.varios/.gnupgp.mutt`:

```
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
set pgp_gpg_pubring="~/.gnupg/pubring.gpg
set pgp_gpg_secring="~/.gnupg/secring.gpg
```

Evidentemente, deberán existir los ficheros `~/.gnupg/pubring.gpg` y `secring.gpg`. Más información sobre GnuPG com el comando `man gpg.gnupg`, `man gpgm`, y `man gpg`.

6.6 Variables de Configuración Mixta

En el caso que deseemos tener más de uno de los tres sistemas a la vez, es preciso modificar algunas de las variables que hemos comentado anteriormente. En realidad tan sólo se trata de eliminar la redundancia en la versión configurada por defecto.

Si, por ejemplo, decidiéramos usar GnuPG como el sistema a utilizar por defecto, todas las teclas relacionadas con PGP/GnuPG de *Mutt* llamarían a éste sistema para cualquier operación de cifrar, descifrar, firmar, verificar, etc...
Para ello debemos definir la variable de configuración **`$set_pgp_default`** **una sóla vez**, así:

```
set pgp_default_version=gpg
```

con lo cual, y para el uso conjunto de los tres sistemas, la parte correspondiente al fichero `~/mutt.varios/.gnupgp.mutt` podría quedar del siguiente modo:

```
set pgp_default_version=gpg     # versión a usar por defecto

set pgp_key_version=default     # clave que se usará por defecto, en este
                                # caso la definirá gnupg

set pgp_receive_version=default # para descodificar usará la versión que se
set pgp_send_version=default    # haya configurado por defecto (gpg)

set pgp_gpg=/usr/bin/gpg        # localización del binario de GnuPG
set pgp_gpg_pubring="~/.gnupg/pubring.gpg        # fichero de claves pub. GnuPG
set pgp_gpg_secring="~/.gnupg/secring.gpg        # fichero de claves secr. GnuPG

set pgp_v2=/usr/bin/pgp         # localización del binario de PGP2
set pgp_v2_pubring="~/.pgp/pubring.pgp           # fichero de claves pub. PGP2
set pgp_v2_secring="~/.pgp/secring.pgp           # fichero de claves secr. PGP2
```

```
set pgp_v5=/usr/bin/pgp         # localización del binario de PGP5
set pgp_v5_pubring="/.pgp/pubring.pkr           # fichero de claves pub. PGP5
set pgp_v5_secring="/.pgp/secring.skr           # fichero de claves secr. PGP5
```

7 Macros interesantes para mutt

Mutt es altamente configurable y puede modificar su forma de trabajo de modo muy flexible si se hace un uso adecuado de las opciones de `.muttrc`.

Aquí se presentan algunas macros que ayudarán a generar firmas evitando seguir el estándar *PGP/MIME*, para enviarlas a destinatarios que sabemos de cierto que no pueden verificar las firmas *PGP/MIME*, y para editar el fichero de alias y volver a cargarlo sin salir de *Mutt* (aunque esto último no está relacionado con *PGP/GnuPG*, se presenta como ejemplo para mostrar el poder de las macros en *Mutt*).

Es posible indicar a Mutt las combinaciones de teclado que queremos utilizar para las opciones de *PGP/GnuPG* que deseamos implementar en el correo electrónico. Aun cuando algunas de estas opciones vienen configuradas por defecto, podemos fácilmente cambiarlas o añadir otras modificando la configuración.

7.1 Firma sobre el propio texto del mensaje sin usar PGP/MIME con PGP5

Antes de la existencia de *PGP/MIME*, la firma de un mensaje iba incluida en el cuerpo del mensaje. Este caso sigue siendo común con muchos lectores de correo.

En caso de que queramos firmar de esta forma, tenemos dos opciones, dejar el tipo *MIME* del mensaje intacto o modificarlo como `application/pgp`.

Para implementar estas dos formas de firmar en *Mutt*, añadiremos las líneas siguientes al fichero `~/mutt.varios/mutt.macros`. Previamente, habremos indicado el origen de este fichero de configuración opcional al fichero de configuración principal `.muttrc`; ver 6.1 (ficheros de configuración opcionales):

```
macro   compose \Cp     "F/usr/bin/pgps\ny"
macro   compose S       "F/usr/bin/pgps\ny^T^Uapplication/pgp; format=text; x-action=sign\n"
```

y de esta forma, pulsando <`Ctrl`>`p` o `S` podremos incluir la firma digital en el adjunto del mensaje sobre el que esté situado el cursor en la pantalla, en la que se encontrará el mensaje listo para ser enviado.

7.2 Firma sobre el propio texto del mensaje sin usar PGP/MIME con GnuPG

Igual que en el caso anterior, pero con GnuPG. Las macros quedan de la siguiente manera:

```
macro   compose \CP     "Fgpg --clearsign\ny"
macro   compose \CS     "Fgpg --clearsign\ny^T^Uapplication/pgp; format=text; x-action=sign\n"
```

7.3 Edición del fichero de alias y recarga del mismo

Con esta macro para incluir en `~/mutt.varios/macros.mutt` puedes editar con *vi* (modificando la linea puedes usar otro editor) el fichero de alias sin salir de *Mutt* pulsando <`Alt`>`a`.

```
macro   index   \ea     "!vi ~/Mail/.alias\n:source =.alias\n"
```

7.4 Más ejemplos de macros

El siguiente listado ha sido obtenido de Roland Rosenfeld y presenta macros para cambiar el «sistema de firmado/cifrado por defecto» y firmar sin PGP/MIME con GnuPG:

```
# ~/Mail/.muttrc.macros
# Fichero de configuración del teclado en Mutt-i
# copiado, modificado y traducido del original:
#
###############################################################
# The ultimative Key-Bindings for Mutt                        #
#                                                             #
# (c) 1997-1999 Roland Rosenfeld <roland@spinnaker.rhein.de>   #
#                                                             #
# $ Id: keybind,v 1.36 1999/02/20 19:36:28 roland Exp roland $ #
###############################################################
#
# Para su correcto funcionamiento, añadir al fichero de
# configuración ~/.muttrc la línea:
# source ~/Mail/.muttrc.macros
#

# Generic keybindings
# (¡para todos los menús de Mutt excepto el del paginador!)
# Con las tres siguientes podemos cambiar el sistema de cripto que
# estemos utilizando por defecto:

# <ESC>1 para GnuPG
macro   generic \e1     ":set pgp_default_version=gpg ?pgp_default_version\n"\
                        "Switch to GNU-PG"

# <ESC>2 para PGP2
macro   generic \e2     ":set pgp_default_version=pgp2 ?pgp_default_version\n"\
                        "Switch to PGP 2.*"

# <ESC>5 para PGP5
macro   generic \e5     ":set pgp_default_version=pgp5 ?pgp_default_version\n"\
                        "Switch to PGP 5.*"


# index, OpMain, MENU_MAIN
# (Menú principal)
# La siguiente sólo funciona desde el menú principal (el que nos
# encontramos nada más abrir Mutt).  La combinación de las teclas
# <CTRL>K nos permite extraer las claves públicas de un mensaje si las
# hubiere (esto se sabe porque el mensaje viene precedido por la letra
# K):

macro  pager   \Ck     ":set pipe_decode pgp_key_version=pgp2\n\e\ek:set pgp_key_version=pgp5\n\e\ek:set pgp_key_version=gpg\n\e\ek:set pgp_key_version=default nopipe_decode\n"\ "Extract PGP keys to PGP2, PGP 5, and GnuPG keyrings"


# pager, OpPager, MENU_PAGER
# (Menú del Paginador)
# Permite las mismas operaciones que en el primer caso, y con las mismas
# combinaciones de claves, pero en este caso desde el menú del
```

```
# paginador:

macro   pager   \e1     ":set pgp_default_version=gpg ?pgp_default_version\n"\
                        "switch to GNUPG"

macro   pager   \e2     ":set pgp_default_version=pgp2 ?pgp_default_version\n"\
                        "switch to PGP 2.*"

macro   pager   \e5     ":set pgp_default_version=pgp5 ?pgp_default_version\n"\
                        "switch to PGP 5.*"


# compose, OpCompose+OpGerneric, MENU_COMPOSE
# (Menú de composición)
# Las siguientes operaciones se realizan desde el menú de composición.
# Esto es, una vez hemos compuesto el mensaje y lo cerramos para
# enviarlo, justo antes de presionar la tecla "Y" que nos permita
# pasárselo al MTA.

# En este caso creamos un menú que se abre al presionar la tecla "P".
# Las opciones de este menú las ligamos a MENU_PGP. Estas son las
# opciones de uso principal (codificación y firma).

bind    compose p       pgp-menu

# Como muchos programas no son capaces de implementar las
# espicificaciones de MIME/PGP (especialmente los de M$), las teclas
# <CTRL>P nos permitirán firmar los mensajes "a la antigua"
# (Application/PGP):

macro   compose \CP     "Fgpg --clearsign\ny"

# A continuación, <CTRL>S nos permitirá firmar "a la MIME/PGP" con la
# clave que tengamos definida por defecto.  Esta macro no es necesaria
# ya que lo mismo podemos hacer desde el menú "P":
macro   compose \CS     "Fgpg --clearsign\ny^T^Uapplication/pgp; format=text; x-action=sign\n"
```

Es posible añadir más macros, y algunas otras vienen ya configuradas por defecto en nuevas versiones de Mutt. Algunas otras opciones incluyen:

- <CTRL>K (extraer claves públicas adjuntas a un mensaje)
- <ESC>K (adjuntar una clave pública a un mensaje)
- <CTRL>F (al usar la contraseña para firmar o descodificar un mensaje, ésta queda grabada en memoria. De este modo podemos borrar la contraseña de la memoria)
- etc...

Para ver qué otras opciones tenemos activadas basta con ir al menú de ayuda (?) desde el menú en que nos encontremos.

8 Algunas «recetas» para Procmail

8.1 Configuración de Procmail para devolver las claves públicas automáticamente

Aunque no es el objetivo de este Como, comentaremos que la forma más segura de obtener la clave pública de una persona es que ella misma nos la provéa en mano.

Como en muchas ocasiones este método no es posible (distancia que separa a los interesados) las personas se pueden enviar por correo electrónico las claves públicas o buscarlas en un servidor de claves, pero ninguno de los métodos garantiza que la clave obtenida sea realmente de quien parece ser su poseedor, a menos que mediante una comunicación de otro tipo que consideremos «seguro» (buscar en la guia de teléfonos al propietario y pedirle que nos lea la «huella» de su clave pública para confirmar que es la misma de la clave que hemos obtenido por el metono no seguro).

Lo que presentamos a continuación es un ejemplo de «receta» para añadir en `.procmailrc` del procesador de correo Procmail para devolver automáticamente su clave pública al remitente cuando reciba un mensaje con un determinado texto en la línea `Asunto`:

```
:0 h
* ^Subject:[    ]+\/(|enviar)[  ]+clave pub\>.*
| mutt -s "Re: $MATCH" ‘formail -rtzxTo:‘ </clau/miclave.asc
```

Lo que aquí dice es lo siguiente: tenemos una copia de nuestra clave pública en ASCII, en un directorio (en este caso el directorio `/clau`), en un fichero llamado `miclave.asc`; así, cuando procmail reciba un mensaje que lleve en la línea de `Asunto:` (`Subject:`) la frase «**enviar clave pub**», enviar el fichero al remitente.

IMPORTANTE: lo que va entre los corchetes es **un espacio** y **un tabulador**.

8.2 Verificación y descifrado automáticos de mensajes firmados sin PGP/MIME

Cuando recibimos un mensaje firmado digitalmente del tipo MIME/PGP y lo abrimos en nuestro lector de correo preferido (Mutt, ¿cuál si no?), éste nos lo reconoce como tal y comprueba la veracidad de la firma siempre que tengamos la clave pública del firmante. Estos mensajes son los que llevan la letra «S» al lado:

```
36  S  05/09 Andres Seco Her ( 12K) Al fin
```

mientras los mensajes codificados llevan la letra «P»:

```
12  P  03/24 Andres Seco Her (6,3K) Re: FW: Re: Mutt - pgp/gnupg
```

Pero si el mensaje viene firmado al estilo «application/pgp», nos encontraremos que al abrir el mensaje no habrá verificación, y éste vendrá con el texto rodeado con la firma digital, así:

```
-----BEGIN PGP SIGNED MESSAGE-----

Date: Tue, 25 May 1999 13:04:26 +0200
From: La Corporación <bill@reboot.com>
Subject: Actualización S.O.
To: Sufrido Usuario <pepe@casa.es>
```

```
Sufrido usuario:

le comunicamos que puede usted adquirir la última actualización del
programa O.E. con la adquisición de nuestro sistema operativo reboot99
por el módico precio de ... etc.


-----BEGIN PGP SIGNATURE-----
Version: 2.6.3ia
Charset: noconv

iKBGNpUBX0235VapRBUy1KklAQGl9wQA3SBMio0bbbajHAnyKMOlx3tcgNG7/UVC
AbqXcUnyGGOo13Nbas95G34Fee3wsXIFo1obEfgiRzqPzZPLWoZdAnyTlZyTwCHe
6ifVpLTuaXvcn9/76rXoI6u9svN2cqHCgHuNASKHaK9034uq81PSdW4QdGLgLoeB
vnGmxE+tGg32=
=Xidf
-----END PGP SIGNATURE-----
```

Para poder verificarlo tendríamos que guardarlo en un fichero y hacerlo desde la línea de comandos. Sin embargo, es posible convertir estos mensajes con *Procmail* para que *Mutt* los reconozca como *MIME/PGP*. Basta con añadir a `.procmailrc`:

```
:0
* !^Content-Type: message/
* !^Content-Type: multipart/
* !^Content-Type: application/pgp
{
    :0 fBw
    * ^-----BEGIN PGP MESSAGE-----
    * ^-----END PGP MESSAGE-----
    | formail \
        -i "Content-Type: application/pgp; format=text; x-action=encrypt"

    :0 fBw
    * ^-----BEGIN PGP SIGNED MESSAGE-----
    * ^-----BEGIN PGP SIGNATURE-----
    * ^-----END PGP SIGNATURE-----
    | formail \
        -i "Content-Type: application/pgp; format=text; x-action=sign"
}
```

Como se puede ver esto sirve tanto para los mensajes firmados como para los cifrados con application/pgp.

8.3 Cambio del tipo MIME para mensajes con claves públicas sin PGP/MIME

Cuando recibes una clave pública desde un MUA sin soporte PGP/MIME, debes salvar la clave en disco y despues insertarla en tu anillo público de llaves, pero, incluyendo estas lineas en `.procmailrc` puedes realizar la inserción directamente desde el propio *Mutt*.

```
    :0 fBw
    * ^-----BEGIN PGP PUBLIC KEY BLOCK-----
    * ^-----END PGP PUBLIC KEY BLOCK-----
    | formail -i "Content-Type: application/pgp-keys; format=text;"
```

Gracias a Denis Alan por esta nota de *Procmail*.

9 Intercambio de mensajes firmados/cifrados entre diferentes clientes de correo y plataformas

Inicialmente, las firmas PGP se añadian al final del texto que deseaban firmar. Posteriormente, se incluyo el tipo MIME `application/pgp` para indicar que el bloque a continuación contenia firma o cifrado PGP, y finalmente con la especificación de PGP/MIME se consiguió separar la firma de los bloques a los que afectaba, de modo que no fueran modificados en absoluto y alguien que no tuviese PGP pudiese ver el mensaje exactamente como fue enviado (no cuando está cifrado, evidentemente) sin añadidos al principio y al final de PGP.

La situación actual es tal que existen pocos clientes de correo capaces de integrarse con PGP para ofrecer PGP/MIME, de modo que suele ser necesario enviar mensajes firmados digitalmente de modo que con clientes de correo que no soportan PGP/MIME puedan ser verificadas las firmas.

En Linux, los clientes de correo disponibles que cumplen la especificación PGP/MIME son mutt-i y pine. En Windows, solo los clientes de correo Eudora en sus versiones 3.x y 4.x soportan PGP/MIME. Si conoce otros clientes que lo soporten esperamos su mensaje indicándonoslo, para incluirlo aquí.

10 Programas y versiones utilizados

Para el desarrollo de esta documentación hemos utilizado las siguientes versiones de mutt:

- Mutt 0.93i - con esta versión de mutt no se puede utilizar GnuPG.
- Mutt 0.95.3i - todas las versiones de PGP y GnuPG son utilizables.

Y las siguientes versiones de PGP y GnuPG:

- PGPi 5.0
- GnuPG 0.4.3
- GnuPG 0.9.4

11 Más información

La documentación original a partir de la cual ha sido obtenido este documento puede ser encontrada en las paginas man de `mutt`, `pg`, `pgp5`, `gnupg`, `procmail`, en los directorios correspondientes de /usr/doc y en las paginas de los paquetes en la world wide web:

- Página Oficinal de Mutt - ***http://www.mutt.org***
- Página Pricipal de GnuPG - ***http://www.gnupg.org***
- Página internacional de PGP - ***http://www.pgpi.com***
- Página Oficinal de Procmail - ***http://www.procmail.org***

Las recomendaciones (request for comments, RFC) que hacen referencia a temas tratados en el documento son las siguientes:

- 1847 - Security Multiparts for MIME: Multipart/signed and Multipart/encripted
- 1848 - MIME Object Security Services
- 1991 - PGP Message Exchange Formats
- 2015 - MIME Security with Pretty Good Privacy (PGP)
- 2440 - OpenPGP Message Format

y pueden ser encontradas en /usr/doc/doc-rfc y en diversos lugares de la world wide web, como *http://metalab.unc.edu* y *http://nic.mil*. Puedes solicitar información sobre las RFCs en *RFC-INFO@ISI.EDU*

12 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos, así como de la producción de documentos originales en aquellos casos en los que no existe análogo en inglés, centrándose, preferentemente, en documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. **:)**), etc.

Diríjase a la sede del Insflug para más información al respecto.

En ella encontrará siempre las **últimas** versiones de las traducciones «oficiales»: *www.insflug.org*. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Además, cuenta con un sistema interactivo de gestión de fe de erratas y sugerencias en línea, motor de búsqueda específico, y más servicios en los que estamos trabajando incesantemente.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

En *http://www.insflug.org/insflug/creditos.php3* cuenta con una detallada relación de las personas que hacen posible tanto esto como las traducciones.

¡Diríjase a *http://www.insflug.org/colaboracion/index.php3* si desea unirse a nosotros!.

«Cartel» Insflug, *cartel@insflug.org*.