Conceptos de

Firma Digital

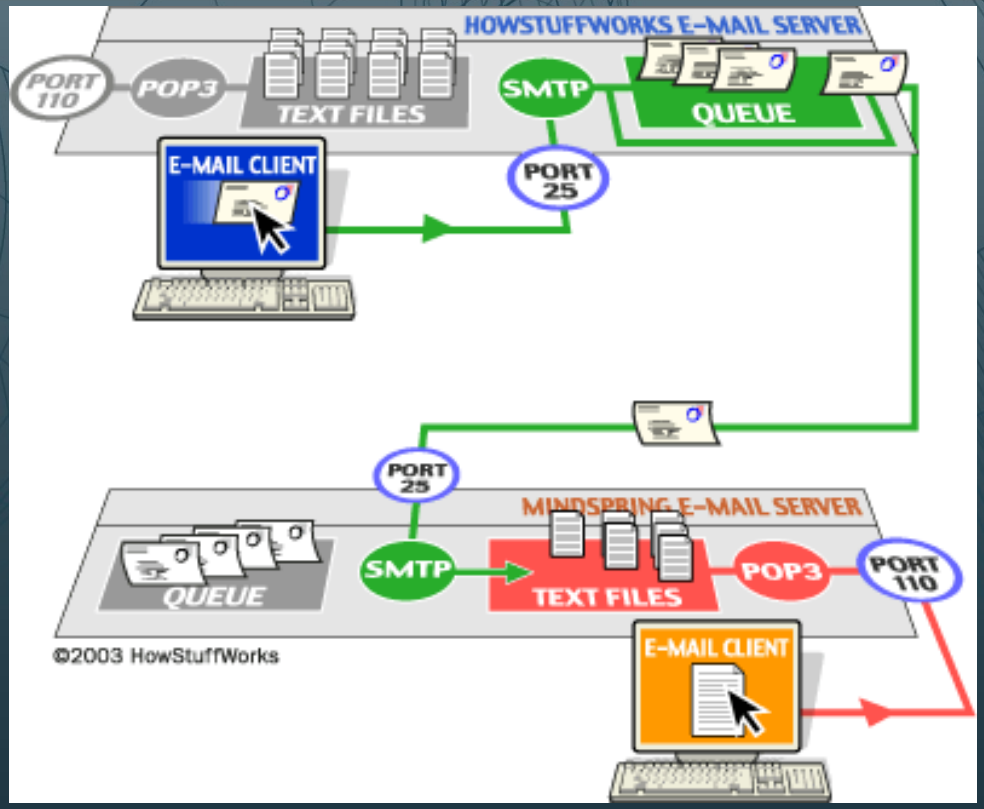
Mg. Javier Echaiz

Dpto. de Cs. e Ing. de la Computación

je@cs.uns.edu.ar

http://cs.uns.edu.ar/~jechaiz

Servidores de e-mail



http://www.howstuffworks.com/email.htm

POP3: Post Office Protocol, port 110
IMAP: Internet Mail Access Protocol, port 143
SMTP: Simple Main Transfer Protocol, port 25

SMTP (RFC 822)

- **Simple Mail Transfer Protocol**.
- Limitaciones de SMTP – No puede transmitir o tiene problemas con:
  - archivos ejecutables/binarios (ej. imagen jpeg).
  - caracteres no US, por ejemplo nuestras vocales acentuadas o ñ.
  - problemas de traducción de ASCII a EBCDIC.
  - líneas de más de *n* caracteres, *n*=72 of *n*=128.

Seguridad en e-mails

- El servicio de e-mails es uno de los dos servicios de red más usados.
  - Junto con la WWW son las killer apps!!!
- Generalmente el contenido de los mensajes no está protegido.
  - Puede ser interceptado en tránsito o en destino.
- El servicio de e-mails es como el servicio postal clásico:
  - Se intercepta y se lee… ☹

¿Qué opciones tenemos?

- Hacer seguras la conexiones entre el cliente y el servidor (lo más sencillo).
  - POP e IMAP sobre ssh/SSL.
  - Acceso https para webmail.
  - Fácil para configurar.
  - Protección frente a las conexiones wireless inseguras.
- Hacer seguro el *end-to-end*.
  - PGP.
  - Es necesario que la otra persona tenga PGP.
  - Práctico en intranets (e.g. empresas, UNS, …).

Mejoras de Seguridad en E-mails

- Confidencialidad.
  - protección de divulgación.
- Autenticidad del origen.
  - Del emisor del mensaje.
- Integridad.
  - Protección contra modificación.
- No repudio del origen.
  - Protección contra el “yo no fui” del emisor.

¿Cómo lo logramos?

¿Cómo lo logramos?

Combinando técnicas de encripción, protocolos y controles de integridad para proteger e-mails.

Requerimientos y Soluciones

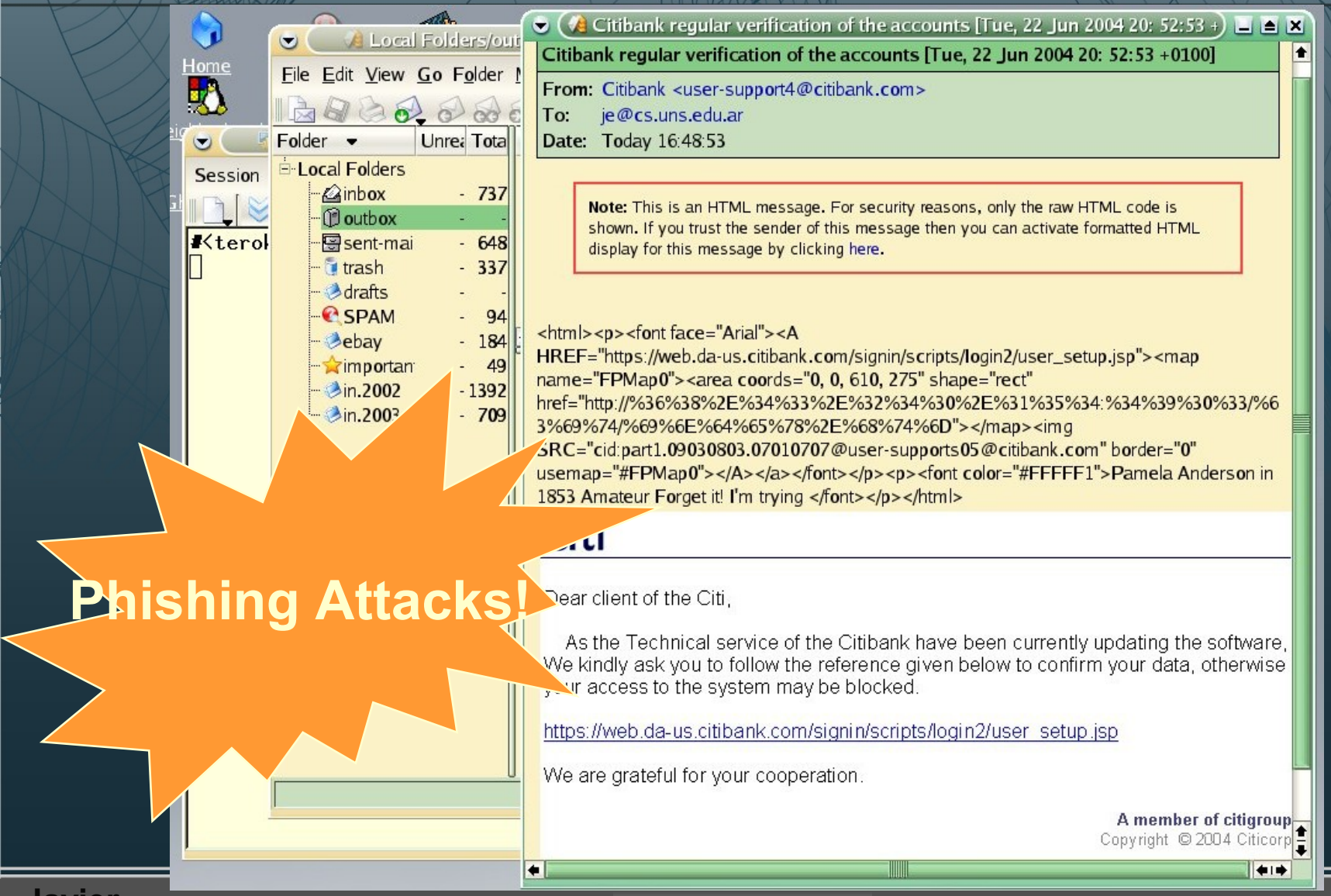
Las rupturas a la confidencialidad y la falsificación de contenidos usualmente se previene con **encripción**. La encripción también puede ayudar a evitar ataques de *replay* si cada mensaje contiene algo único (número de secuencia) que se encripta junto con el mensaje.

La criptografía simétrica no nos sirve para proteger contra falsificación por parte del receptor, pues el receptor y el emisor comparten la misma clave. Sin embargo los esquemas de clave pública le permiten al receptor desencriptar pero no encriptar. Los e-mails viajan por puntos de la red que no controlamos por lo que es virtualmente imposible para el emisor y receptor evitar el *bloqueo de entrega*.

Claves y Pares de Claves

- La encripción es una forma de codificar algo en algo distinto.
  - E.g. simple corrimiento de 3 letras (algoritmo del Cesar).
- …pero el destinatario debe conocer la "clave".
  - ¿Cómo le decimos cuál es la clave de forma segura?
- La respuesta es: claves asimétricas.
- Claves públicas/privadas.
  - "Fingerprint" para verificación.
  - Password protege clave privada por seguridad.
  - Incluye dirección(es) de e-mail(s).

Ejemplos de e-mails Fraudulentos



Ejemplos de e-mails Fraudulentos

Amenazas a la seguridad en e-mails

- SPAM.
  - Falsificación de “From: dirección”. Relación con virus.
  - Filtros anti-SPAM pueden bloquear e-mails legítimos.
- Phishing / Hoax / Spoof e-mail.
  - El e-mail que recién vimos que parecía ser del Citibank te redirige a una página web en Rusia…

    (James Bond: Desde Rusia con Amor).

- Hipótesis:
- Las amenazas ACTUALES sobre los e-mails pueden resolverse simplemente agregando firma digital.

Firmas Digitales

La firma digital es una transformación de un mensaje utilizando un sistema de cifrado asimétrico de manera que la persona que posee el mensaje original y la clave pública del firmante, pueda establecer de forma segura, que dicha transformación se efectuó utilizando la clave privada correspondiente a la pública del firmante, y si el mensaje es el original o fue alterado desde su concepción.

Firmas Digitales

La firma digital es una secuencia de números generados mediante un algoritmo matemático. Para la generación y verificación de una firma digital, se requiere:

Mensaje a Firmar

Claves Pública y Privada

Certificados Digitales

- El ***Certificado Digital*** es un documento electrónico infalsificable que contiene los datos del ***Usuario*** y su ***Clave Pública***.

Por ejemplo los certificados X.509.

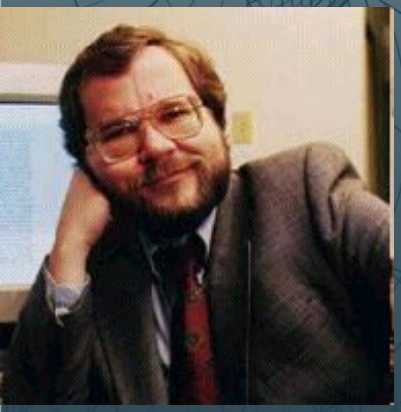
E-Mails Seguros

A comienzos de los años 90 hacen su aparición dos sistemas de correo electrónico seguro:

- PEM (Private Enhanced Mail)
- PGP (Pretty Good Privacy)

De los dos, PGP fue quien se convirtió en el estándar de facto en clientes de e-mail seguro.

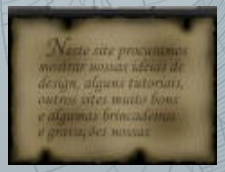
Pretty Good Privacy



- Philip R. Zimmermann es el autor del PGP.
- PGP provee confidencialidad, integridad y autentificación. Puede ser utilizado como sistema de e-mail seguros o para almacenar archivos encriptados.

¿Por qué es el PGP tan Popular?

- Se encuentra disponible para muchas plataformas.
- Basado en algoritmos conocidos y probados.
- Aplicabilidad de amplio espectro.
- No fue creado ni por el gobierno ni por organizaciones.



Pretty Good Privacy (PGP)

- Zimmermann publica la versión 1.0 de PGP en 1991 con mínimos requisitos de hardware y software.
- En 1992 aparece la versión 2.0 en la que ya participan programadores de todo el mundo. Su código se escribe fuera de USA para evitar las leyes restrictivas respecto al software criptográfico y sus problemas legales.
- En 1993 aparece la versión 2.3a muy popular en sitios FTP y válida para varias plataformas de sistemas operativos.
- En 1994 participa en el proyecto el Massachusetts Institute of Technology MIT y aparecen las versiones 2.4, 2.5 y 2.6.
- La versión 2.6.3i se populariza a nivel mundial.

Características de PGP 2.6.3i

- PGP, en su versión 2.6.3i (internacional) se convirtió a mediados de la década de los 90 en un estándar de facto. De hecho, muchos usuarios “siguen fieles” a esta versión.
- Encripta todo tipo de datos en entornos MS-DOS y UNIX. Su orientación principal es el cifrado de los datos y la firma digital en correo electrónico.
- Los algoritmos básicos que usa son:
  - IDEA para cifrar con sistema de clave secreta.
  - RSA para intercambio de claves y firma digital.
  - MD5 para obtener la función hash de la firma digital.

Características del cifrado local

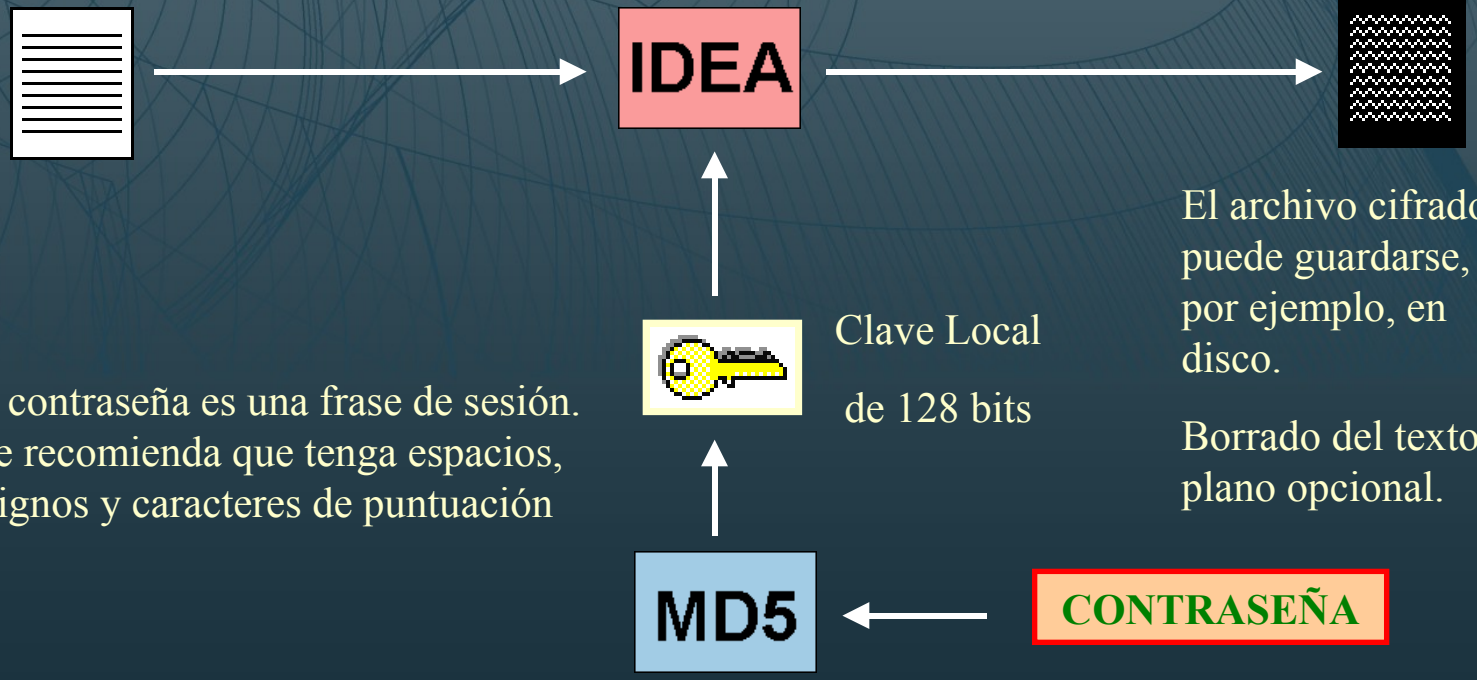
- Esta operación sirve para mantener los archivos protegidos, por ejemplo en el disco rígido.
- El acceso al texto plano sólo será posible si se conoce una clave o contraseña que es la frase usada al cifrar.
- Recuerde que si después de cifrar el archivo borra físicamente el texto plano -operación que realiza una grabación de unos y ceros aleatorios en la zona de almacenamiento del disco- le será **imposible** recuperarlo si olvida el password.

Pasos del cifrado local con IDEA

Pasos:

1. PGP solicita una frase de sesión: ésta debe ser lo suficientemente larga como para evitar ataques por combinaciones.
2. Se aplica el algoritmo de hash MD5 a esa contraseña, generando así una clave de 128 bits.
3. PGP cifra el documento con el algoritmo IDEA y le agrega la extensión .pgp.
4. Permite luego hacer un borrado físico del archivo plano.

Esquema de cifrado local con IDEA



Cada nuevo cifrado requiere una contraseña.

Operaciones con claves asimétricas

- Las operaciones de PGP para cifrar, descifrar, firmar y la comprobación posterior de la firma digital, usan los algoritmos de funciones hash, de clave pública y de clave secreta.
- Para poder enviar y recibir correo seguro, es necesario contar al menos con las siguientes claves:

  Clave pública del destinatario.

  Par de claves asimétricas del emisor.

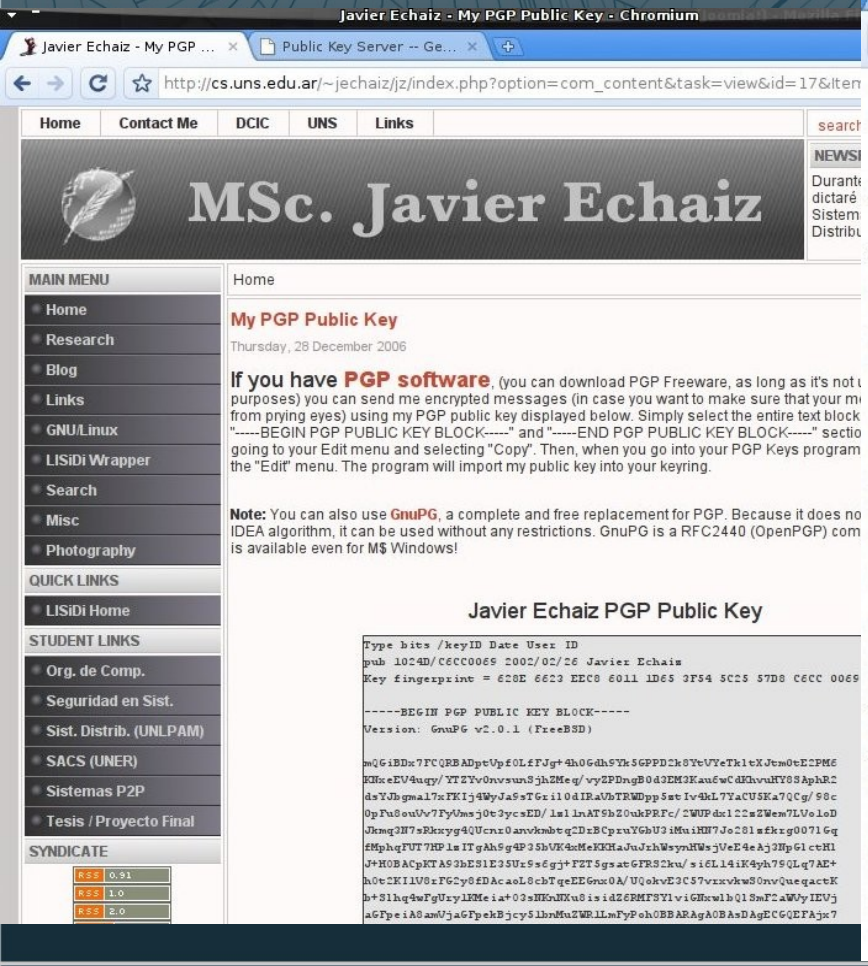
  Generación de claves con RSA, etc.

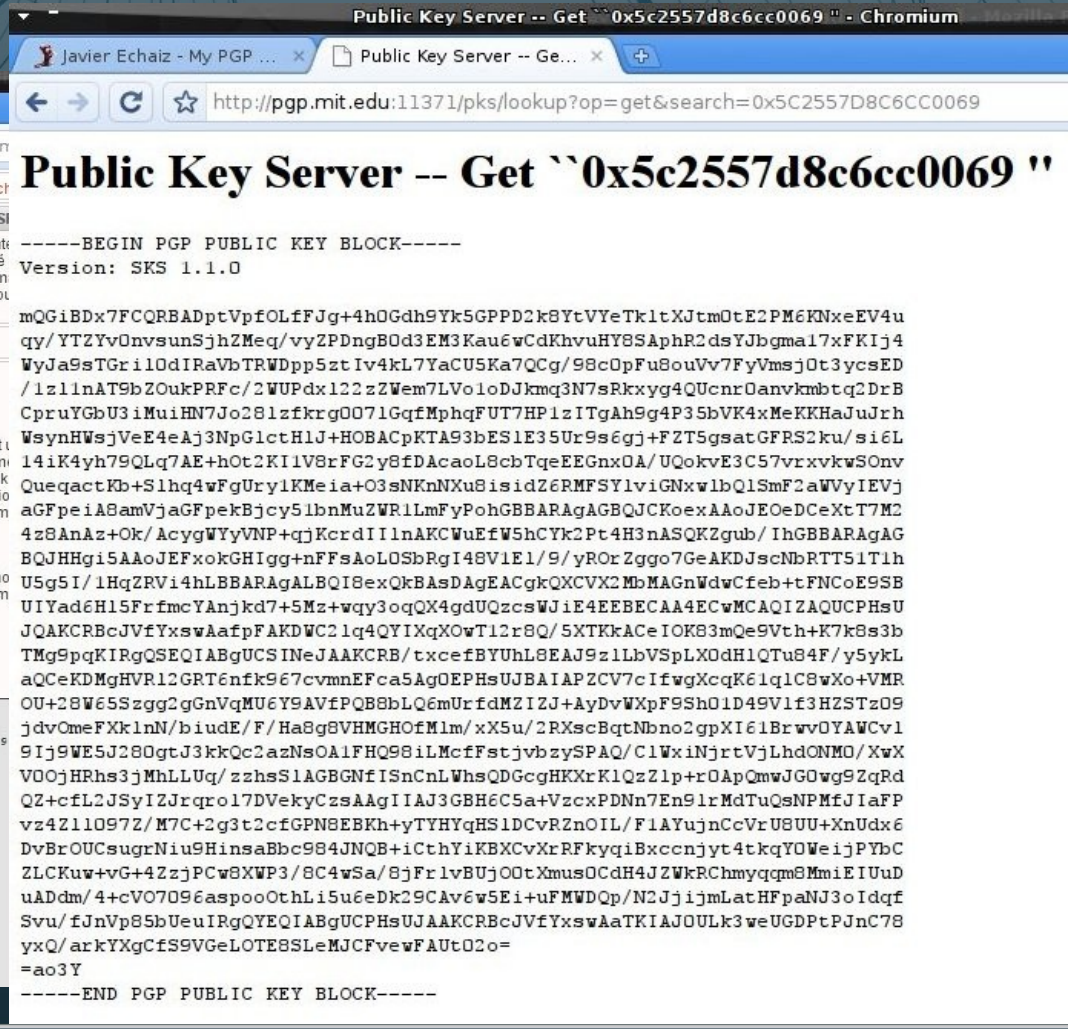
Anillos de claves asimétricas

- Con las claves pública y privada generadas y otras claves públicas que podrá importar de otros usuarios, se crean dos anillos de claves:
  - Anillo de claves públicas: archivo pubring.pkr en el que se guardan las claves públicas del usuario propietario (puede tener más de una identidad) y las claves públicas de importadas.
  - Anillo de claves privadas: archivo secring.skr en el que se guarda la o las claves privadas del usuario propietario.

Mi Clave Pública PGP

Autopropaganda… ☺





Mi Clave Pública PGP

Sin importar como obtenga mi clave pública,

Por favor verifique el fingerprint de la misma

(e.g. mediante el comando

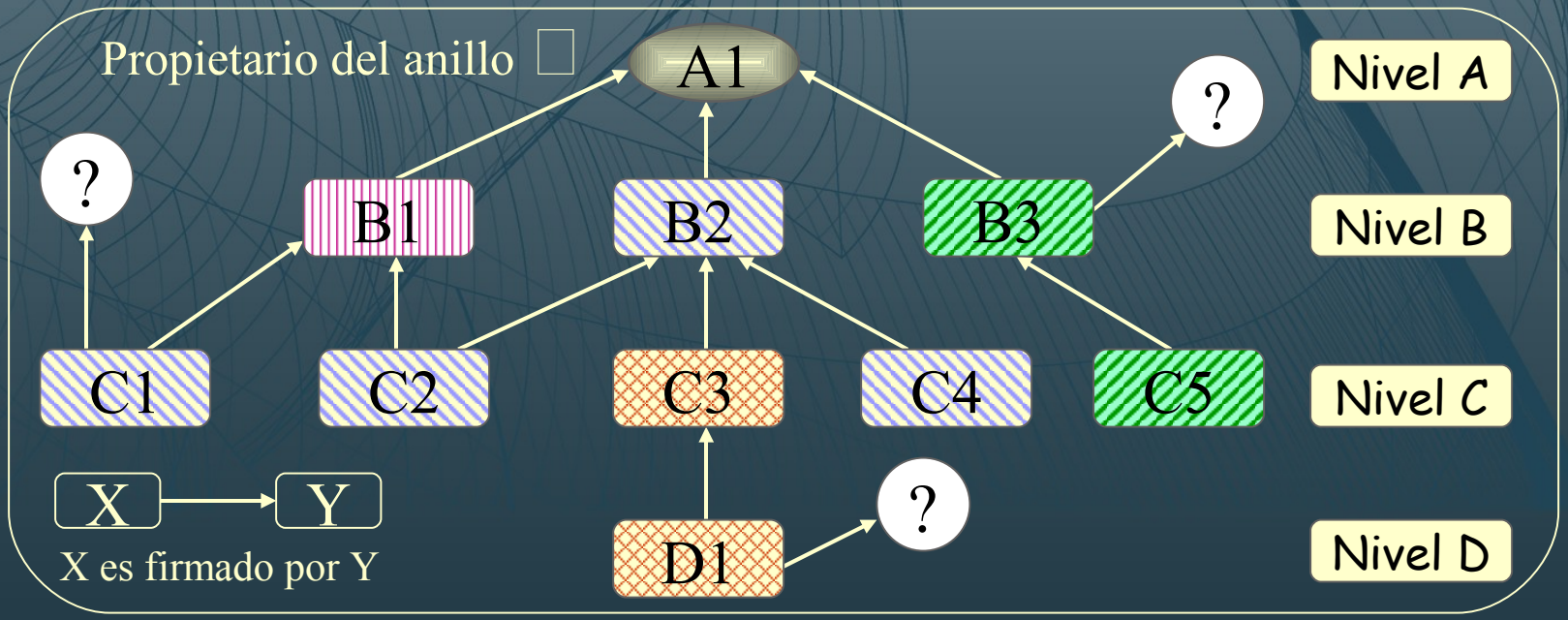
gpg --fingerprint C6CC0069)

Debería obtener lo siguiente:

```
$univac:/tmp> gpg --fingerprint C6CC0069
pub   1024D/C6CC0069 2002-02-26
      Key fingerprint = 628E 6623 EEC8 6011 1D65  3F54 5C25 57D8 C6CC 0069
uid                  Javier Echaiz <jechaiz@cs.uns.edu.ar>
sub   2048g/1990D9F6 2002-02-26
```

http://www.cs.uns.edu.ar/~jechaiz/pubkey.html

Gestión del anillo de claves públicas



A1 cree en el propietario de la clave para firmar otra clave

A1 cree parcialmente en el propietario de la clave para firmar otra clave

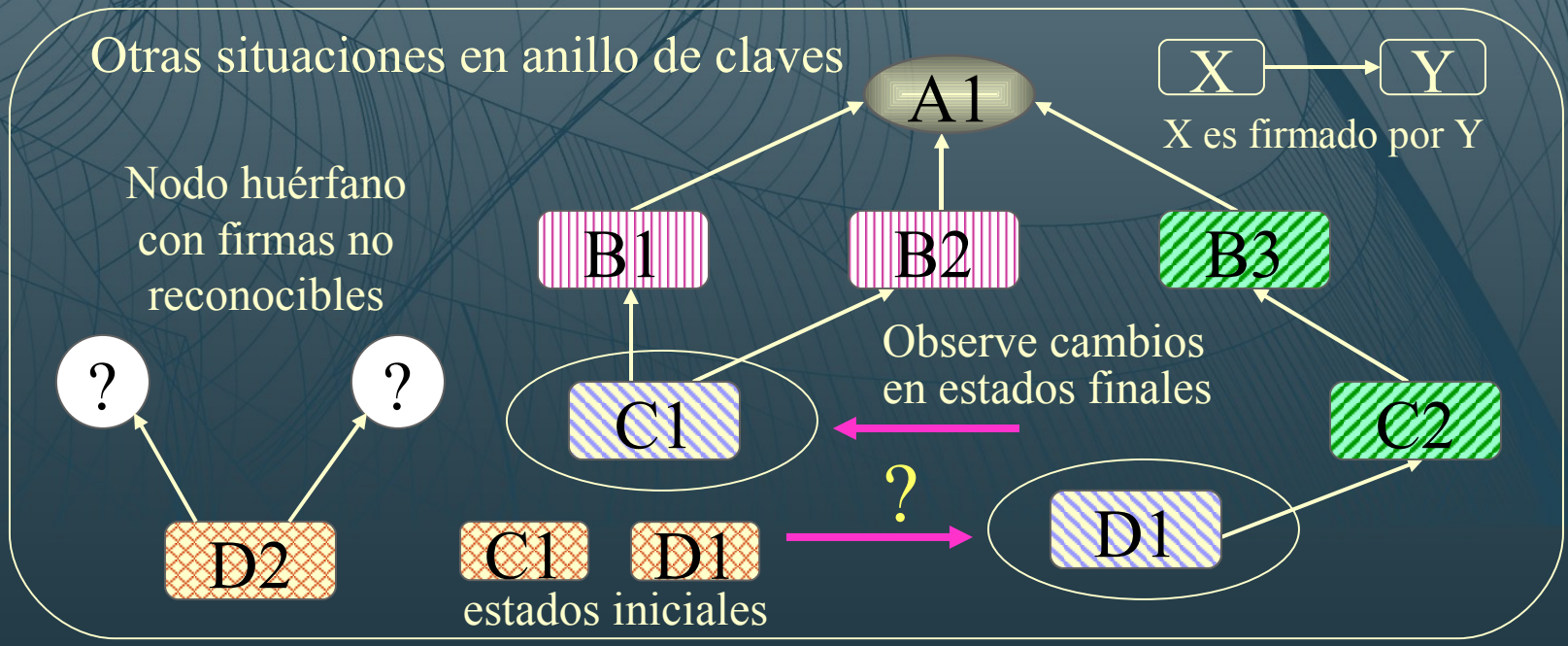
A1 cree en legitimidad de clave

A1 no cree que la clave sea legítima

Más...

? La clave está firmada por un usuario o Autoridad que no está en anillo de claves de A1

Otros escenarios de confianza en PGP



A1 cree en el propietario de la clave para firmar otra clave

A1 cree parcialmente en el propietario de la clave para firmar otra clave

PGP hace que A1 crea en la legitimidad de las claves pues tienen al menos dos firmas parciales (B1-B2) o una completa (C2) pero no da confianza para firmar

A1 no cree que la clave sea legítima

Problema en estos Escenarios de Confianza

La gestión de claves en PGP se basa en la confianza mutua:
¡Los amigos de tus amigos son mis amigos!

- En un sistema abierto como Internet su uso para comercio electrónico, bajo esta gestión de claves de confianza mutua, **resulta inaceptable**.
- La solución que PGP contempla actualmente, es la aceptación de **Autoridades Certificantes** como certificadores de claves públicas.

Modelo de Confianza y Gestión de Claves

- Cualquier usuario puede certificar a cualquier otro (modelo anárquico).
- Cada usuario decide en quién confiar y qué tanto confiar.
- “Anillo de claves públicas”: ED que almacena las claves públicas de un usuario, junto con su nivel de confianza.
- ¿Cómo obtener claves públicas de forma “segura”?
  - Offline (en persona, por teléfono, etc.).
  - Páginas web personales (e.g. http://www.cs.uns.edu.ar/~jechaiz/pgp.html)
  - Amigo en quien confiamos (“web of trust”) .
  - CA en quien confiamos.

Revocación de Claves Públicas

- El dueño envía un certificado de revocación de clave.
- Es un certificado de firma normal con un indicador de revocación.
- Se utiliza la clave privada correspondiente para firmar la revocación.
- Comando: gpg --gen-revoke

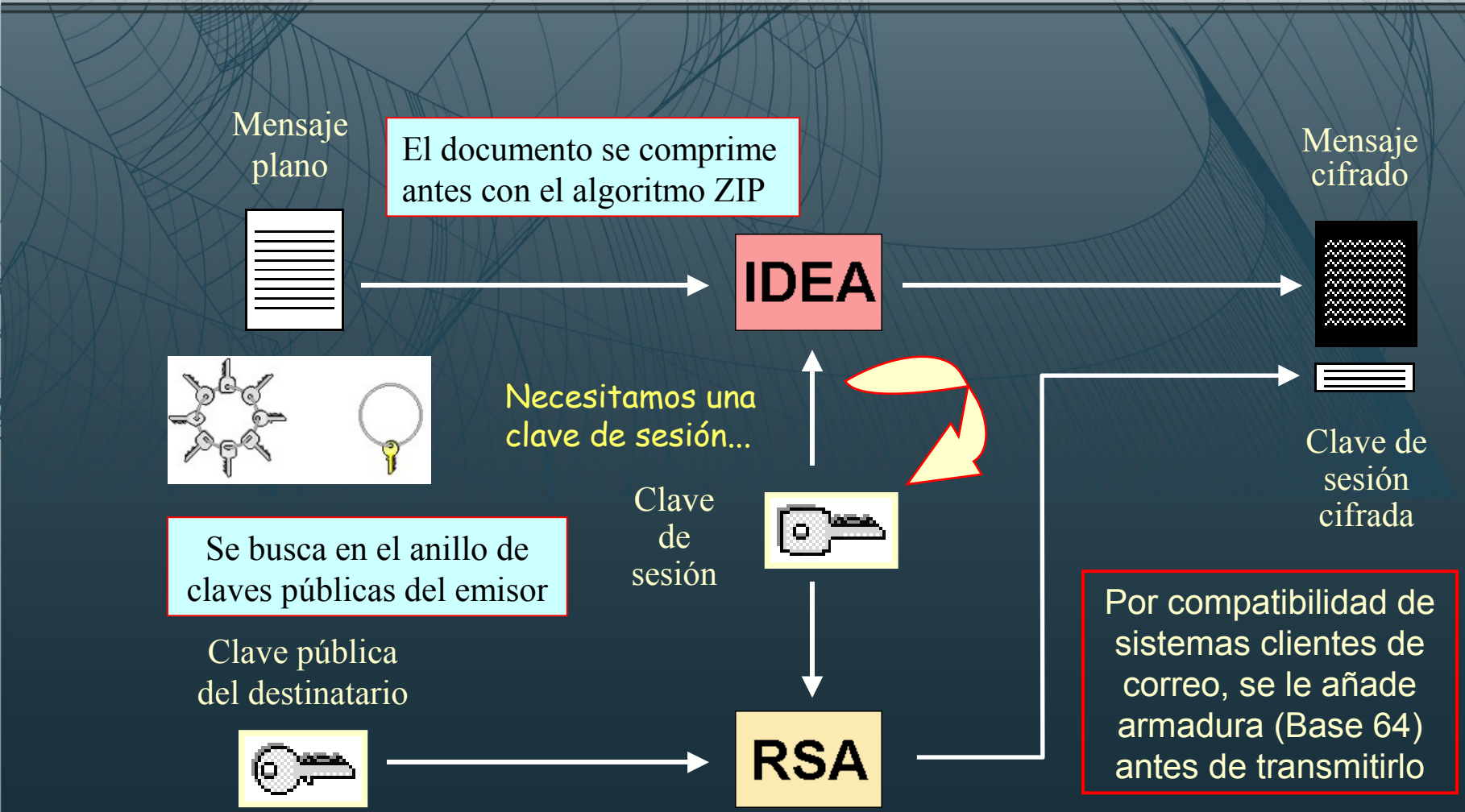
Pasos Cifrado con Clave Pública de Destino

Pasos:

1. PGP genera un número aleatorio de 128 bits que será la clave de sesión.
2. Se cifra el mensaje con dicha clave usando IDEA.
3. Se cifra la clave de sesión con la clave pública RSA del destinatario y se añade al criptograma.
4. Se añade el identificador ID de la clave pública del destinatario a la clave de sesión cifrada en el paso 3 como indicador de la identidad del receptor.

Recuerde que el correo electrónico no es en general una comunicación en tiempo real por lo que, aunque se envía una clave para descifrar el criptograma en recepción, no se trata de una clave de sesión en los mismos términos que se usa, por ejemplo, en una comunicación SSL.

Cifrado con clave pública de destino

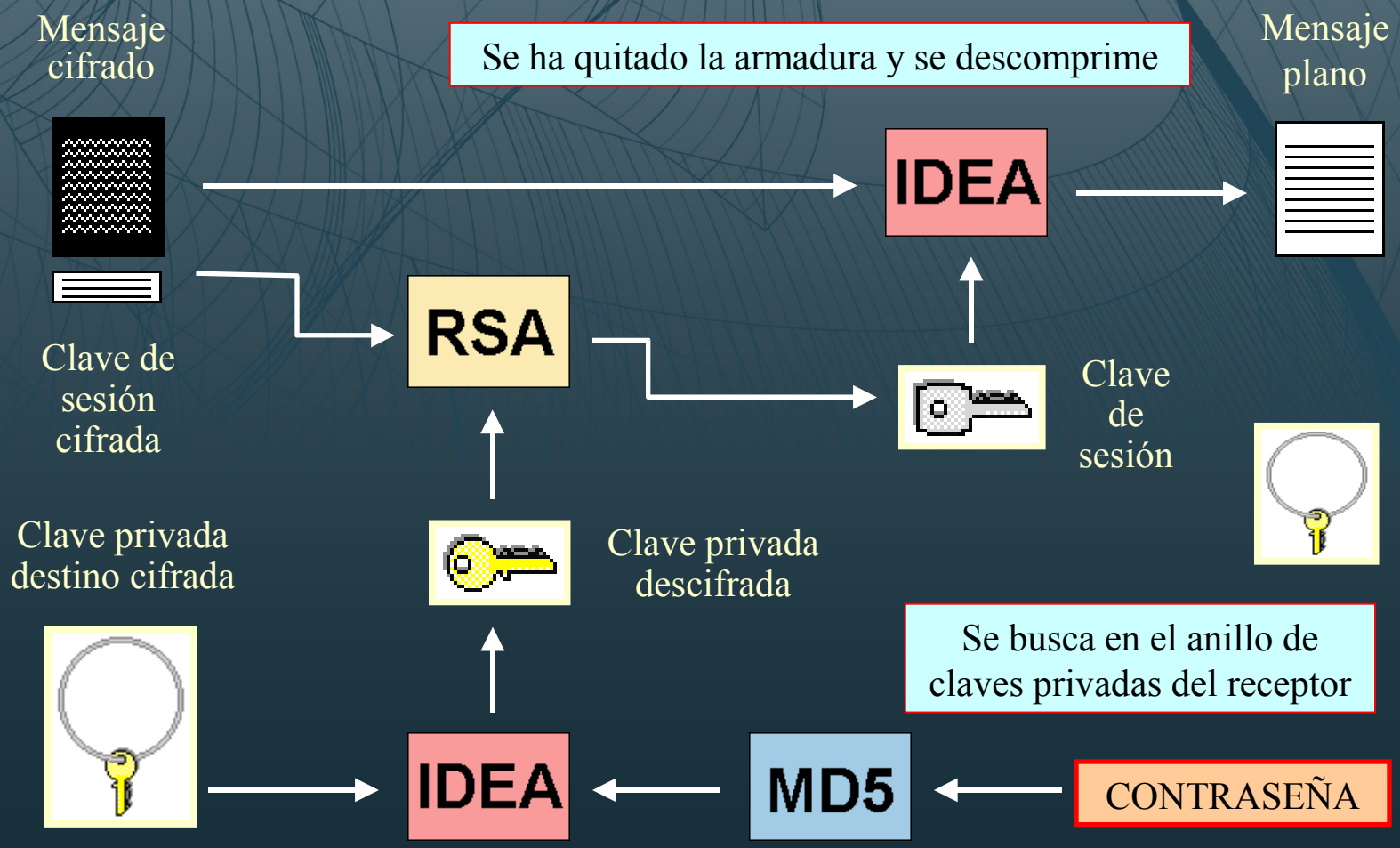


Pasos descifrado con Clave Privada Destino

Pasos:

1. PGP busca en la cabecera del criptograma el identificador de usuario ID (receptor) que se ha añadido en la clave de sesión cifrada.
2. Se busca la clave privada del identificador ID en el anillo de claves privadas del receptor.
3. Se accede a la clave privada plano, descifrándola con IDEA al introducir el propietario ID su frase de paso. Sólo en ese momento está plano.
4. Con la clave privada se descifra la clave de sesión.
5. Con la clave de sesión se descifra el criptograma.

Pasos descifrado con Clave Privada Destino



Firma digital RSA

Mensaje plano

Se va a firmar un mensaje plano

Mensaje plano

MD5

RSA

Clave privada descifrada

Necesitamos nuestra clave privada...

Bloque de firma digital

Si se desea se puede enviar también cifrado

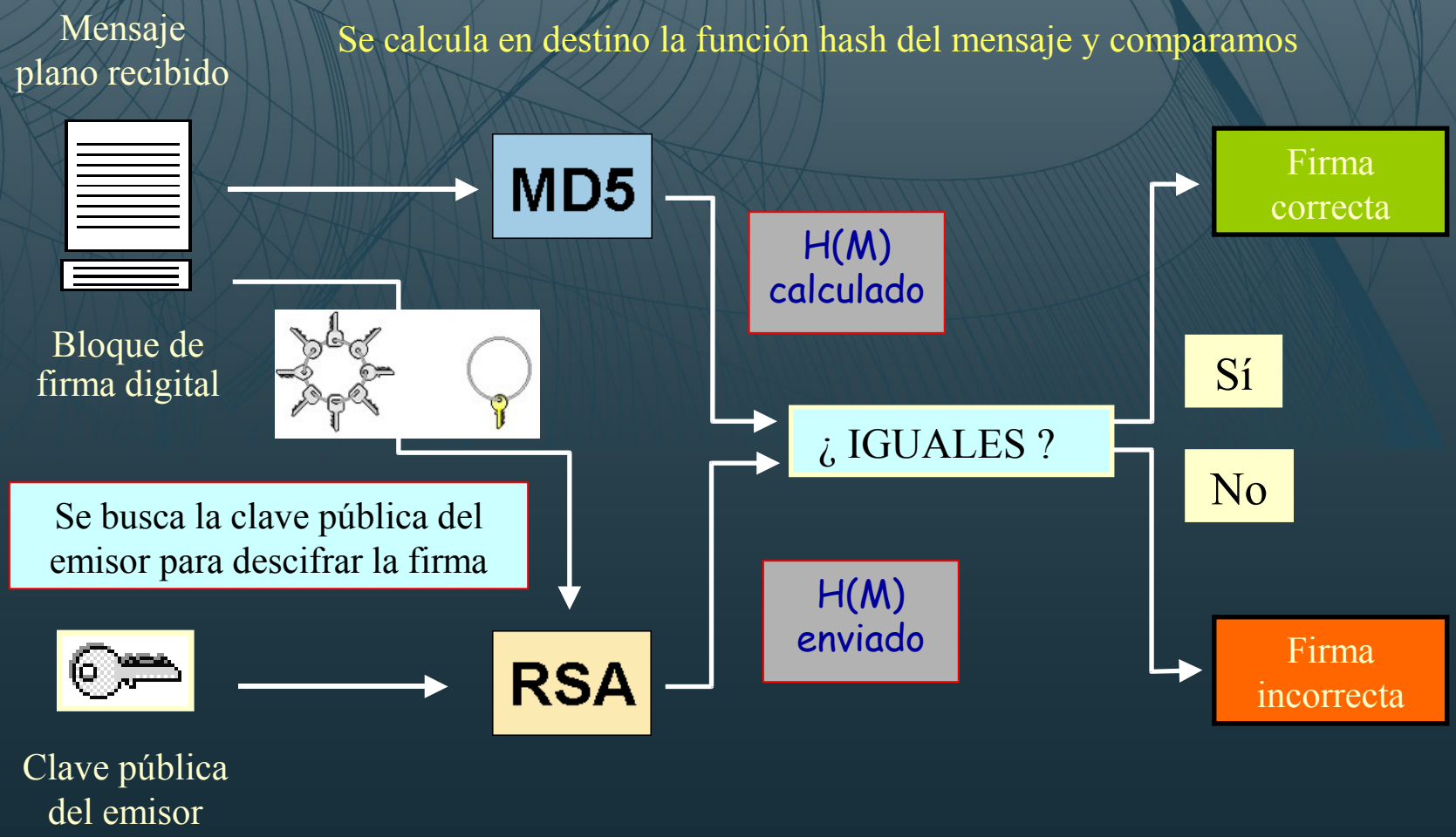
Clave privada cifrada IDEA

IDEA

MD5

CONTRASEÑA

Comprobación de la firma digital RSA

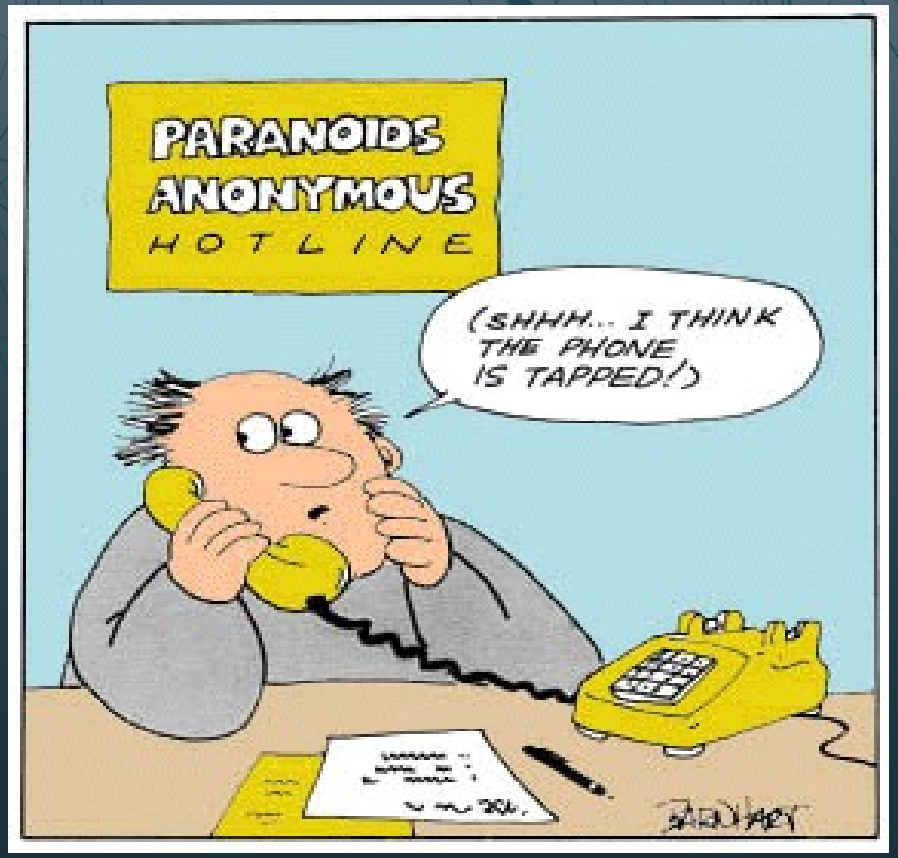


Firmado de mensajes

- Se usa tu propia clave **privada**.
- Si el destinatario está seguro de que tiene tu clave pública entonces puede estar seguro de que el mensaje vino de vos.
- Tu clave pública está MUY disponible.

Paranoico…?

- El mensaje se encripta con la clave pública del destinatario y se firma con tu propia clave privada.
- De esta forma el destinatario no sólo es el único que puede desencriptarlo sino que además puede verificar la integridad/autenticidad.



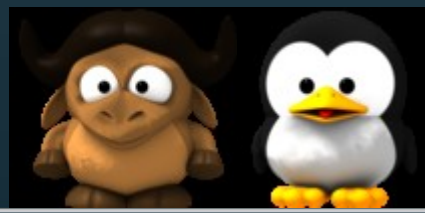
¿Cómo sabe que es MI clave?

- Cualquier persona puede generar una clave de cualquier otra persona.
- La Firma de una clave confirma que la misma pertenece a la persona correcta.
  - Verificar la identidad por voz, pasaporte, carnet, etc.
  - Usar *fingerprint* para asegurarse.
- Crear cadena de confianza.
- A veces se dan eventos de “firmado”.
  - http://www.ox.compsoc.net/compsoc/events/pgp-keysigning.html
- Buscar Smith y God en pgp.mit.edu! ☺

Keysignings

- Los Keysignings son eventos donde las claves pueden intercambiarse y subirse a los Key Servers.
- Son una oportunidad para que tu clave sea firmada por otros… y que vos firmes las claves de otros.
- Los encargados del evento te pueden ayudar a generar claves y certificados de revocación.
- Estos “eventos de firmado” podrán (ojalá) ser parte de los meetings de nuestro LUG!!!

BBLUG



Algoritmos soportados por GPG

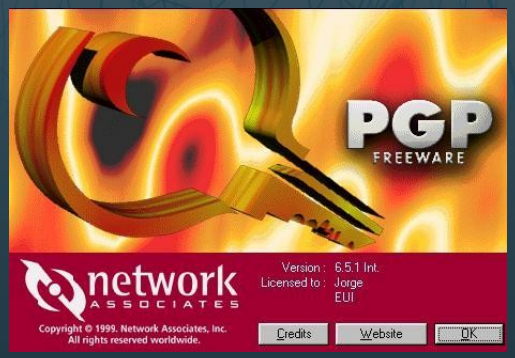
Últimas:
1.4.9 y 2.0.12

```
$univac:/tmp> gpg --version
gpg (GnuPG) 2.0.7
Copyright (C) 2007 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: ~/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ELG
Cipher: 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH
Hash: MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512
Compression: Uncompressed, ZIP, ZLIB, BZIP2
```

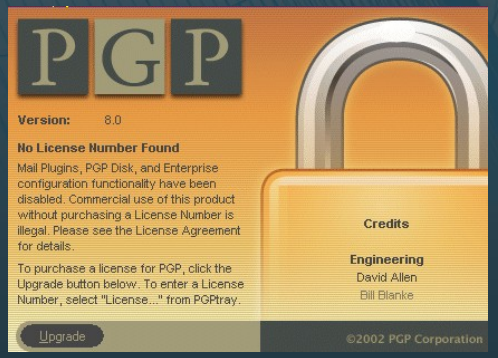
Algunas versiones de PGP en Windows

Desde la versión 5.0 hasta la actual 8 (freeware) los esquemas de cifrado y firma digital han cambiado muy poco aunque presentan mayores prestaciones. No obstante, recuerde que algunas prestaciones sólo estarán activadas en versiones comerciales.



PGP 6.5.1

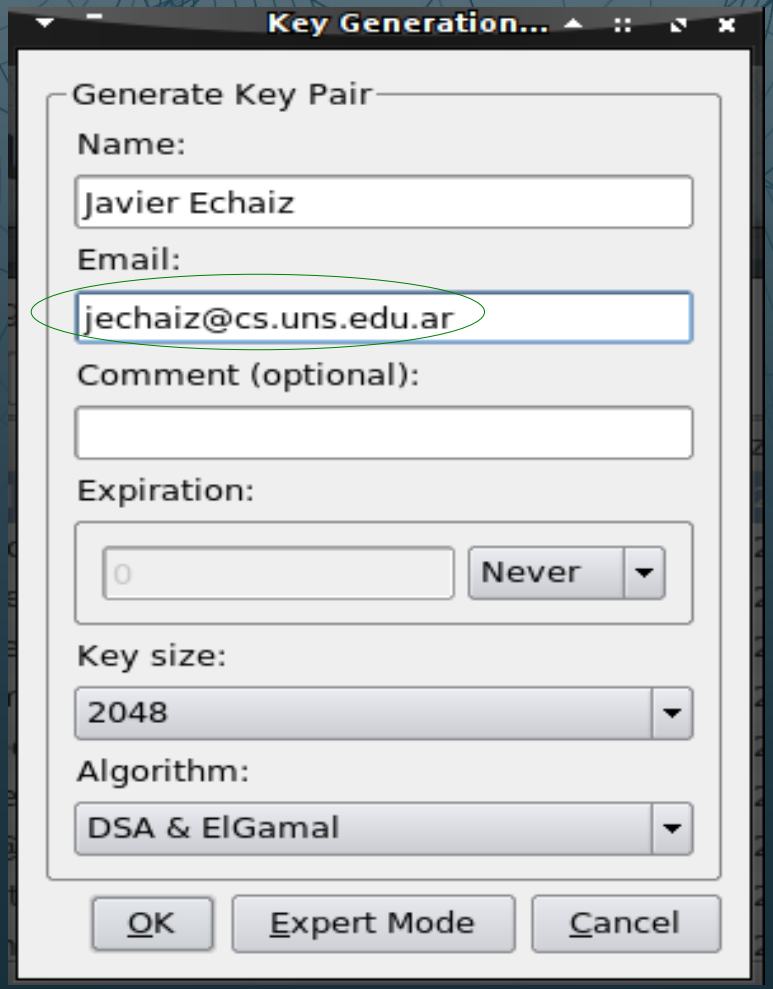
PGP 7.0.3



PGP 8.0

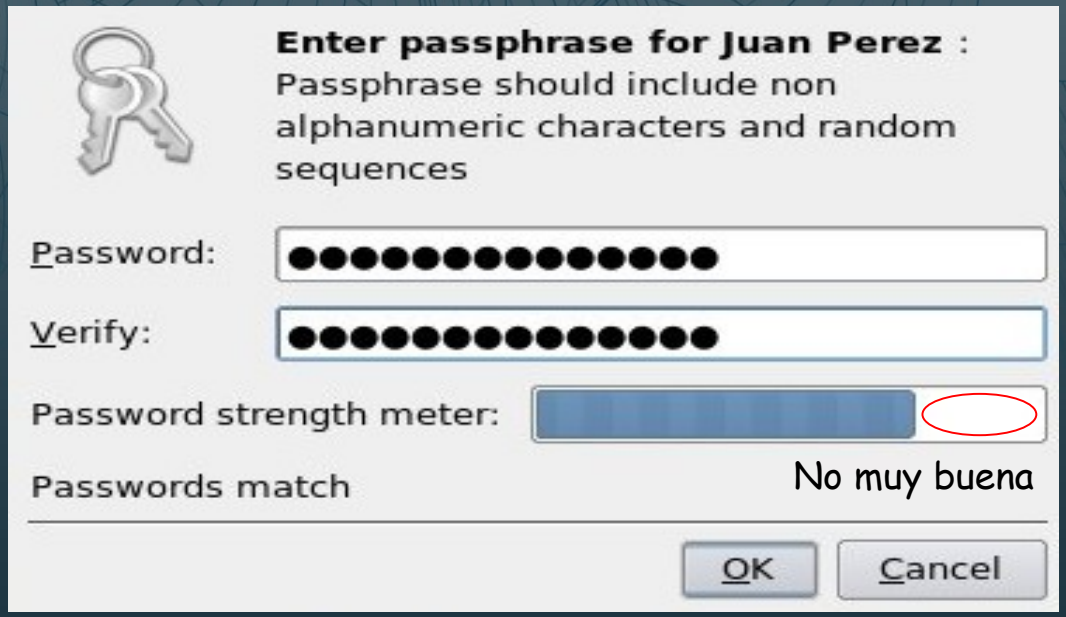
Veremos algunas operaciones de estas tres versiones con mayor detalle. Recuerde, eso sí, que la versión 7.0.3 no tiene su código fuente abierto.

Nombre del usuario y su correo



- ✓ No es necesario que la dirección de correo sea la real... No obstante sirve para que los que se comunican con nosotros sepan que esa clave pertenece a esa dirección de email.
- ✓ Además “colabora” con los clientes de correo electrónico.

Frase de sesión para cifrar la clave privada

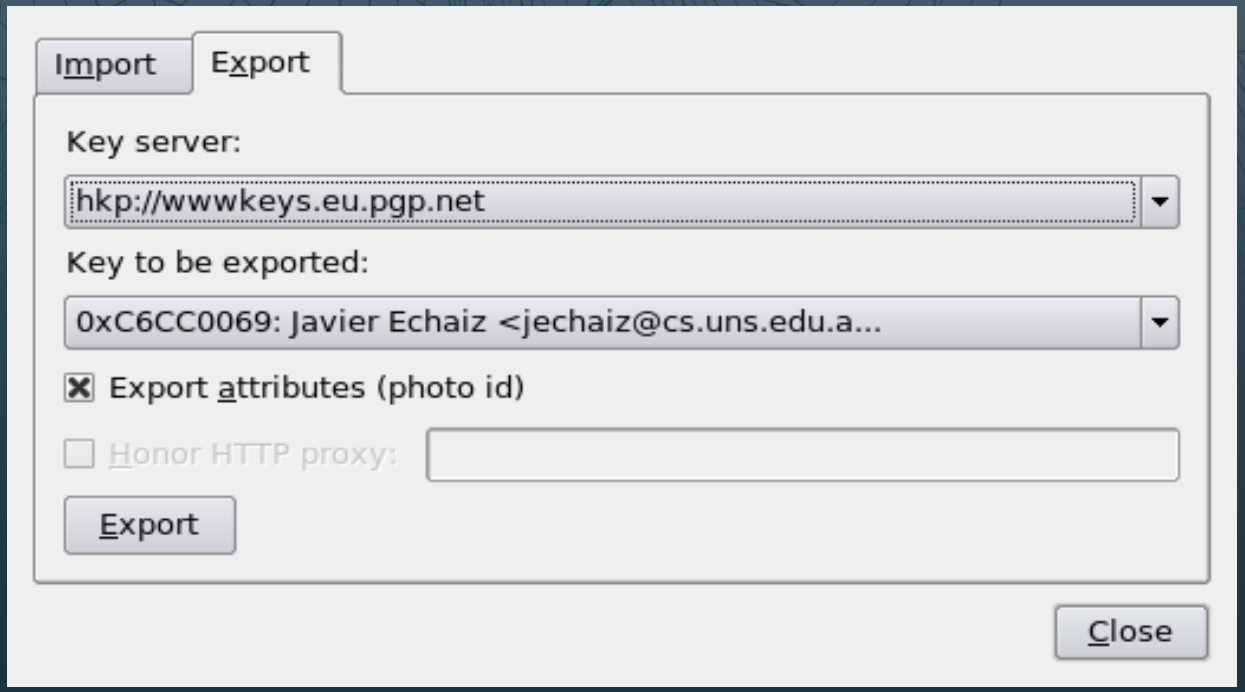


- La frase de sesión debe tener varias palabras para que sea difícil un ataque por diccionario.
- Si quita la opción Hide Typing podrá ver lo que escribe.
- Por seguridad, no está permitido usar el clipboard para copiar la frase de arriba en la casilla de confirmación.

Generación de clave concluida

- Se ha concluido satisfactoriamente la generación del par de claves pública y privada que se guardarán en los anillos **pubring.pkr** y **secring.pkr** (~/.gnupg/).
- La clave privada se guarda cifrada con una clave de sesión que se genera al aplicar una función hash a la frase de sesión del propietario.

Envío de la clave al Servidor de Claves



- ✓ Se puede enviar la clave a un servidor.
- ✓ Cuanto más “pública” sea la clave pública mejor.

Otras operaciones con PGP

PGP permite hacer otras operaciones interesantes:

- Dividir (split) una clave privada en varias subclaves, de forma que para deshacer una operación de cifrado o firmar un documento se requiera un umbral de estas subclaves dadas por diseño. Está basado en el esquema de Blakely-Shamir.
- Firmar las claves públicas de otros usuarios con distintos niveles de confianza.
- Revocar una clave, habilitar o deshabilitar una clave.
- Enviar, buscar y actualizar claves desde servidores.
- Cifrar con la opción sólo para tus ojos (JFYE), crear grupos, etc.

Le **recomiendo** que éstas y otras operaciones las realice a modo de ejercicio, instalando PGP en su máquina.

GPG (GNU Privacy Guard)

"GnuPG is a **complete and free replacement for PGP**. Because it does not use the patented IDEA algorithm, it can be used without any restrictions. GnuPG is a **RFC2440 (OpenPGP)** compliant application."

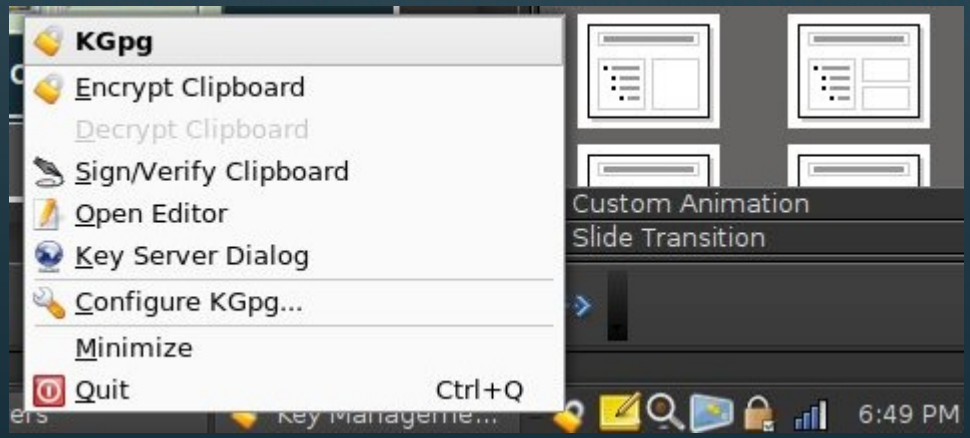
http://www.gnupg.org/

Version 1.0.0 has been released on September 7th, 1999. The current stable version is 1.4.9. GnuPG is Free Software."

Gnome / KDE

+

KGPG



Click Mouse Der.

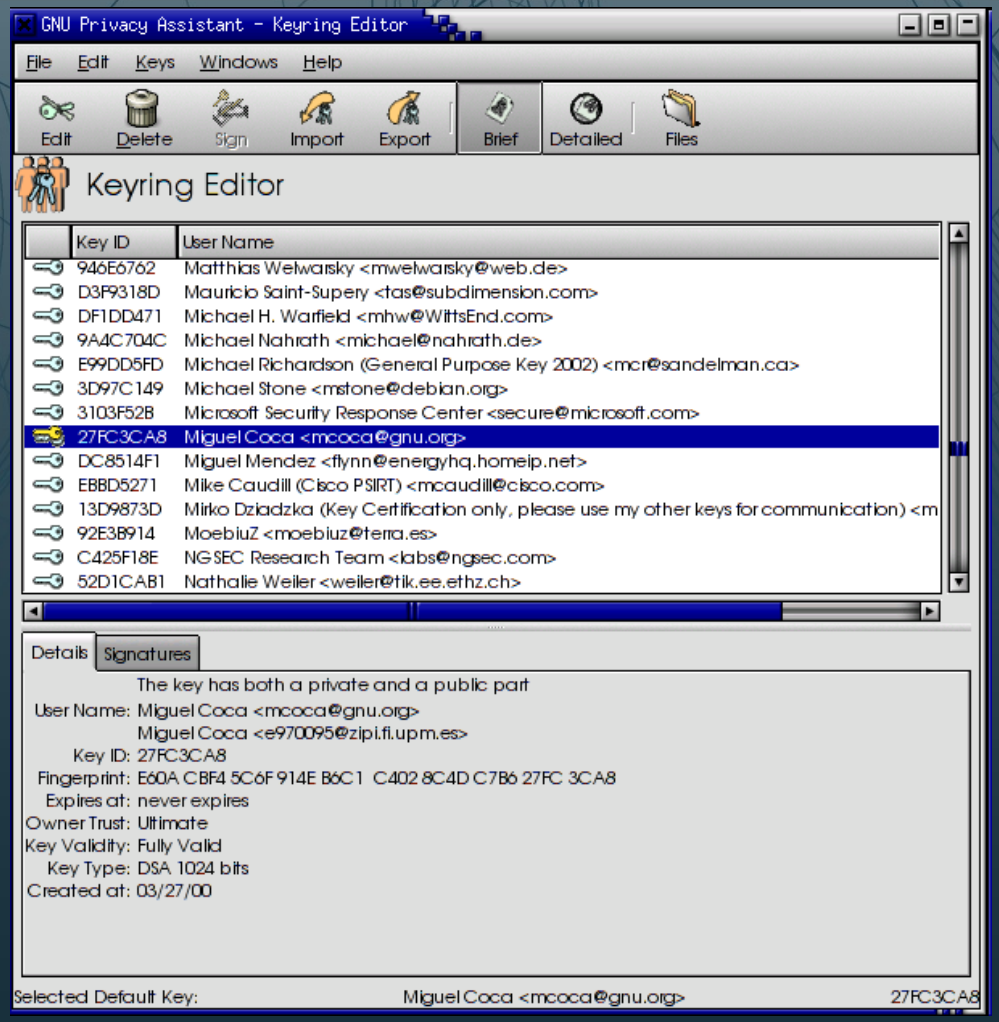
Features del GPG

- **Full replacement of PGP.**
- **Does not use any patented algorithms.**
- **GPLed, written from scratch.**
- Can be used as a filter program.
- Full OpenPGP implementation (see RFC2440 at RFC Editor).
- Better functionality than PGP and some security enhancements over PGP 2.
- **Decrypts and verifies PGP 5, 6, 7 and 8 messages.**
- **Supports ElGamal, DSA, RSA, AES, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 and TIGER.**
- Easy implementation of new algorithms using extension modules.
- The User ID is forced to be in a standard format.
- Supports key and signature expiration dates.
- **English, Danish, Dutch, Esperanto, Estonian, French, German, Japanese, Italian, Polish, Portuguese (Brazilian), Portuguese (Portuguese), Russian, Spanish, Swedish and Turkish language support.**
- Online help system.
- Optional anonymous message receivers.
- Integrated support for HKP keyservers (http://wwwkeys.pgp.net).
- Clears signed patch files which can still be processed by patch.
- and many more things....

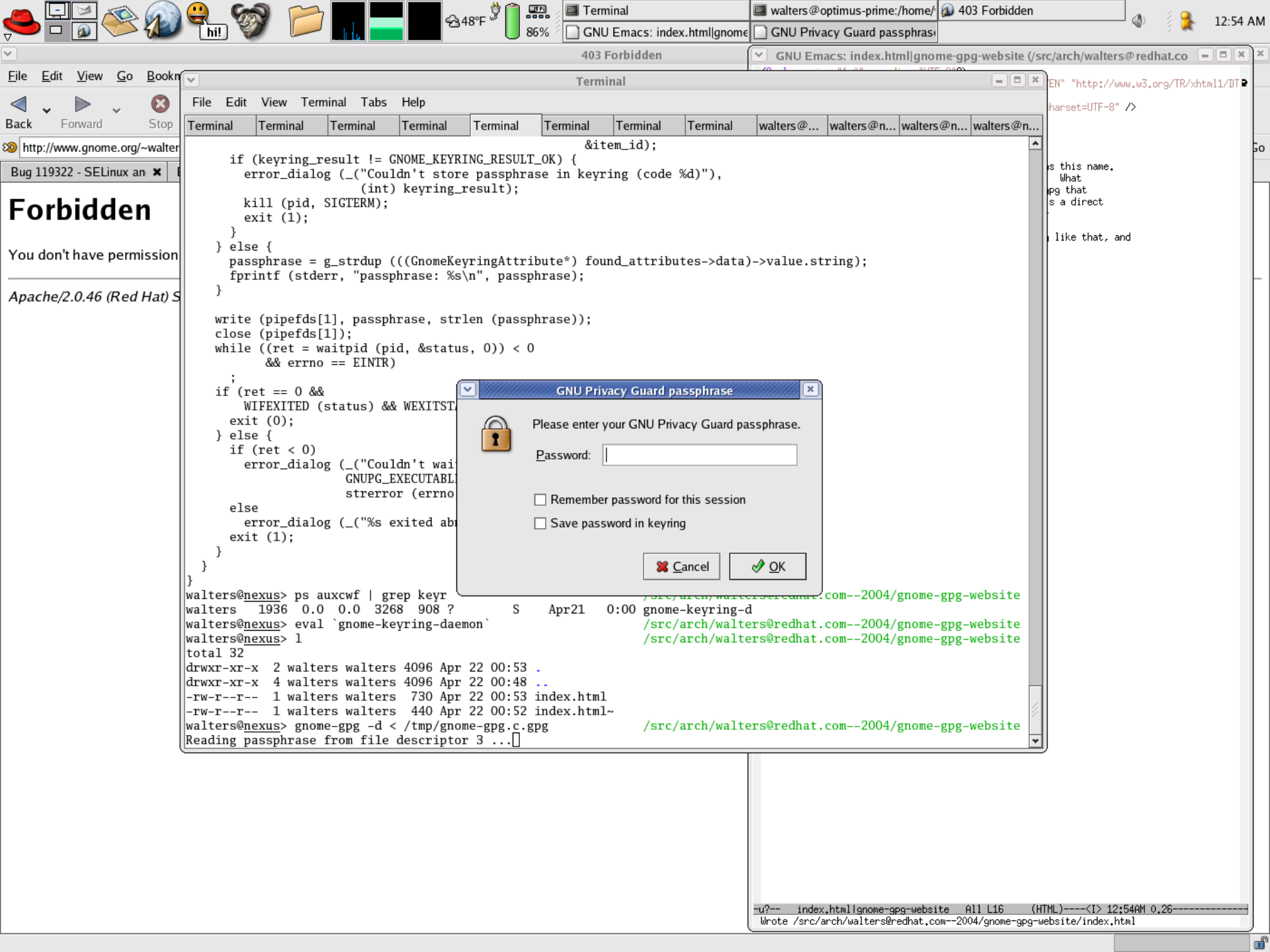
ADK?
NO

http://es.tldp.org/COMO-INSFLUG/es/mini/pdf/GPG-Mini-Como.pdf

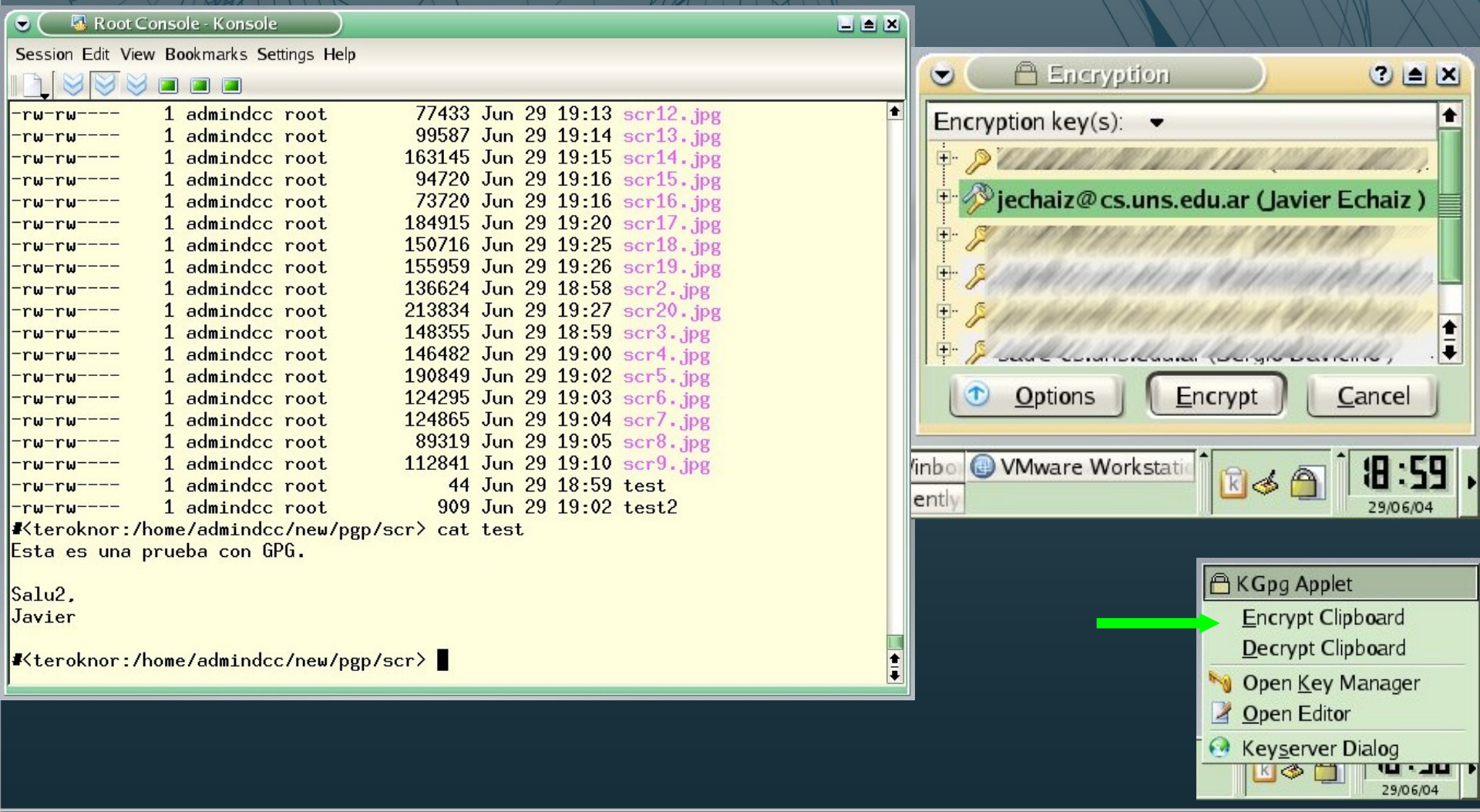
GPA, the Gnu Privacy Assistant



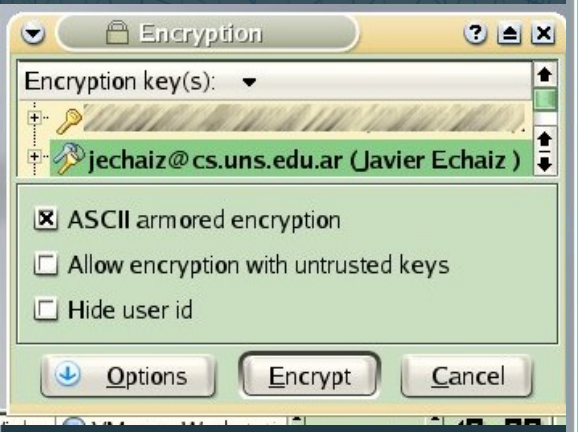
http://www.gnupg.org/related_software/gpa/index.en.html

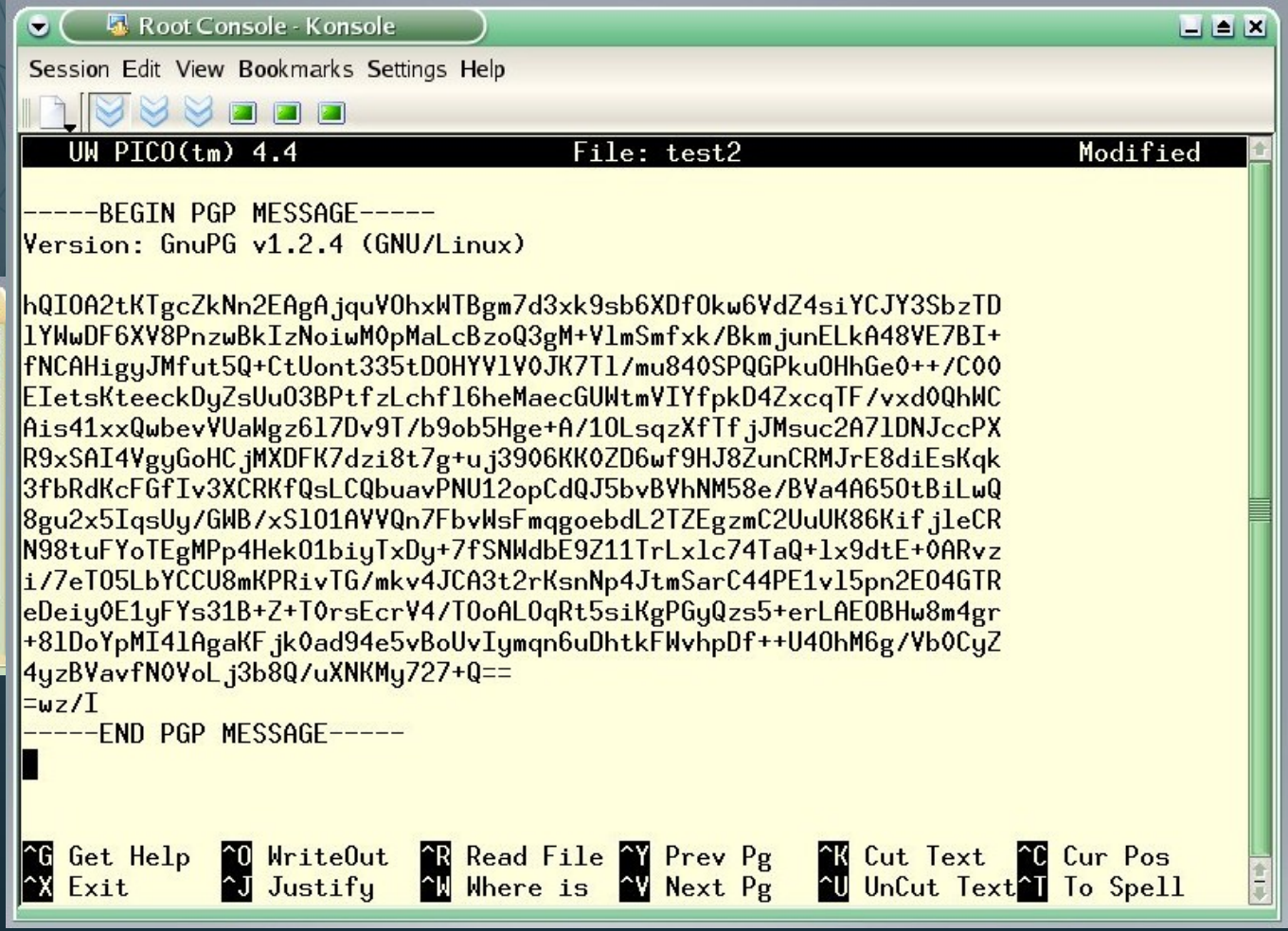

Terminal
walters@optimus-prime:/home/
403 Forbidden
GNU Emacs: index.html|gnome
GNU Privacy Guard passphras
48°F
86%
12:54 AM
403 Forbidden
GNU Emacs: index.html|gnome-gpg-website (/src/arch/walters@redhat.co
File Edit View Go Bookm
Back Forward Stop
http://www.gnome.org/~walter
Bug 119322 - SELinux an
Forbidden
You don't have permission
Apache/2.0.46 (Red Hat) S
Terminal
File Edit View Terminal Tabs Help
Terminal Terminal Terminal Terminal Terminal Terminal Terminal Terminal walters@... walters@n... walters@n... walters@n...
&item_id);
if (keyring_result != GNOME_KEYRING_RESULT_OK) {
error_dialog (_("Couldn't store passphrase in keyring (code %d)"),
(int) keyring_result);
kill (pid, SIGTERM);
exit (1);
} else {
passphrase = g_strdup (((GnomeKeyringAttribute*) found_attributes->data)->value.string);
fprintf (stderr, "passphrase: %s\n", passphrase);
write (pipefds[1], passphrase, strlen (passphrase));
close (pipefds[1]);
while ((ret = waitpid (pid, &status, 0)) < 0
&& errno == EINTR)
;
if (ret == 0 &&
WIFEXITED (status) && WEXITST
exit (0);
} else {
if (ret < 0)
error_dialog (_("Couldn't wai
GNUPG_EXECUTABL
strerror (errno
else
error_dialog (_("%s exited ab
exit (1);
}
}
}
GNU Privacy Guard passphrase
Please enter your GNU Privacy Guard passphrase.
Password:
Remember password for this session
Save password in keyring
Cancel
OK
walters@nexus> ps auxcwf | grep keyr
walters 1936 0.0 0.0 3268 908 ? S Apr21 0:00 gnome-keyring-d
walters@nexus> eval `gnome-keyring-daemon`
walters@nexus> l
total 32
drwxr-xr-x 2 walters walters 4096 Apr 22 00:53 .
drwxr-xr-x 4 walters walters 4096 Apr 22 00:48 ..
-rw-r--r-- 1 walters walters 730 Apr 22 00:53 index.html
-rw-r--r-- 1 walters walters 440 Apr 22 00:52 index.html~
walters@nexus> gnome-gpg -d < /tmp/gnome-gpg.c.gpg
Reading passphrase from file descriptor 3 ...
/src/arch/walters@redhat.com--2004/gnome-gpg-website
/src/arch/walters@redhat.com--2004/gnome-gpg-website
/src/arch/walters@redhat.com--2004/gnome-gpg-website
/src/arch/walters@redhat.com--2004/gnome-gpg-website
/src/arch/walters@redhat.com--2004/gnome-gpg-website
-u?-- index.html|gnome-gpg-website All L16 (HTML)----<I> 12:54AM 0.26
Wrote /src/arch/walters@redhat.com--2004/gnome-gpg-website/index.html

GPG – Clipboard Encryption

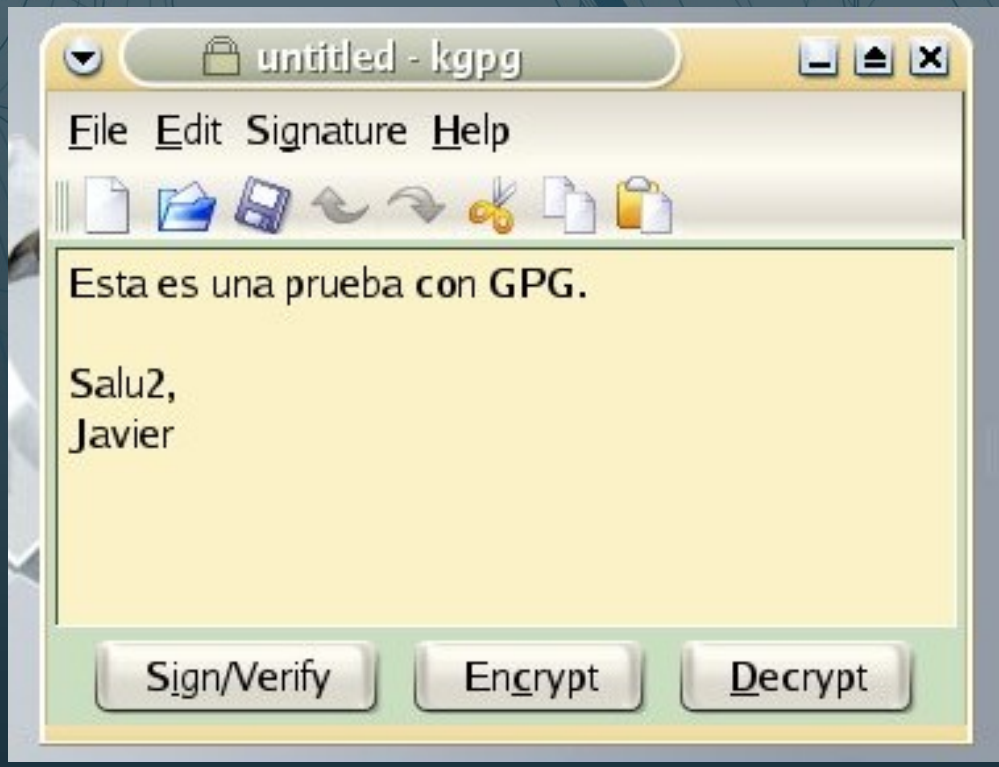


GPG – Clipboard Encryption

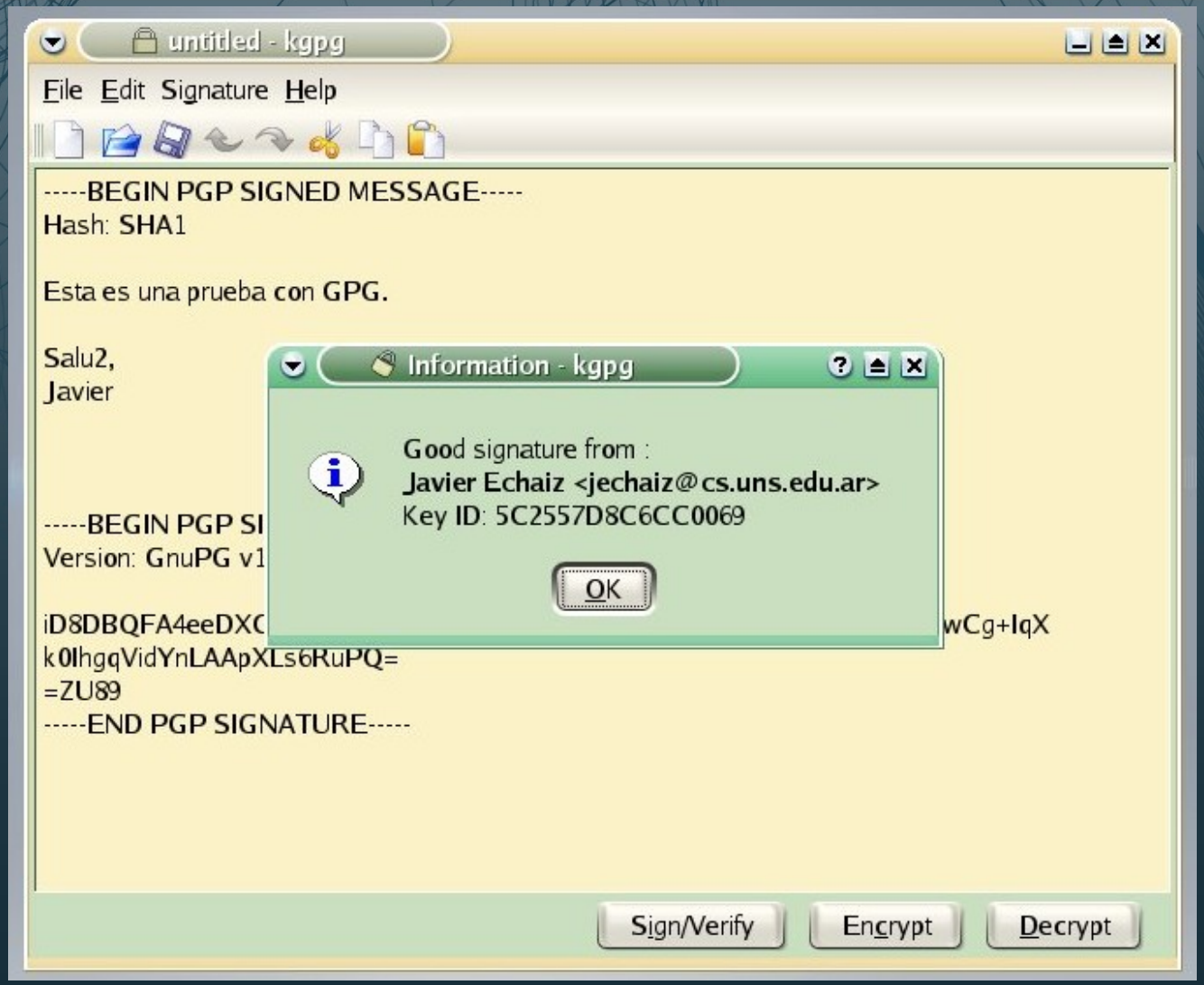




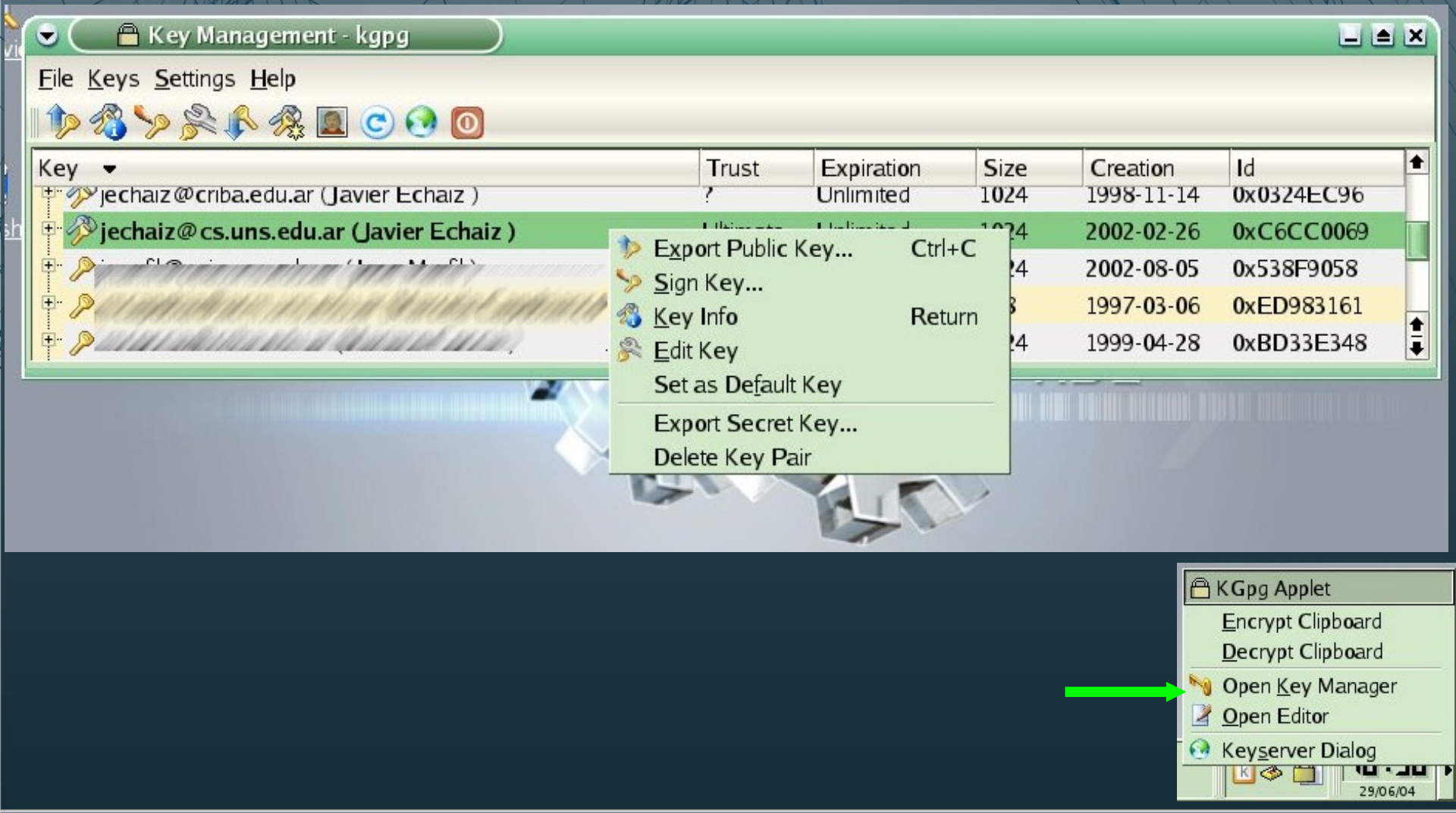
GPG – Clipboard Decryption



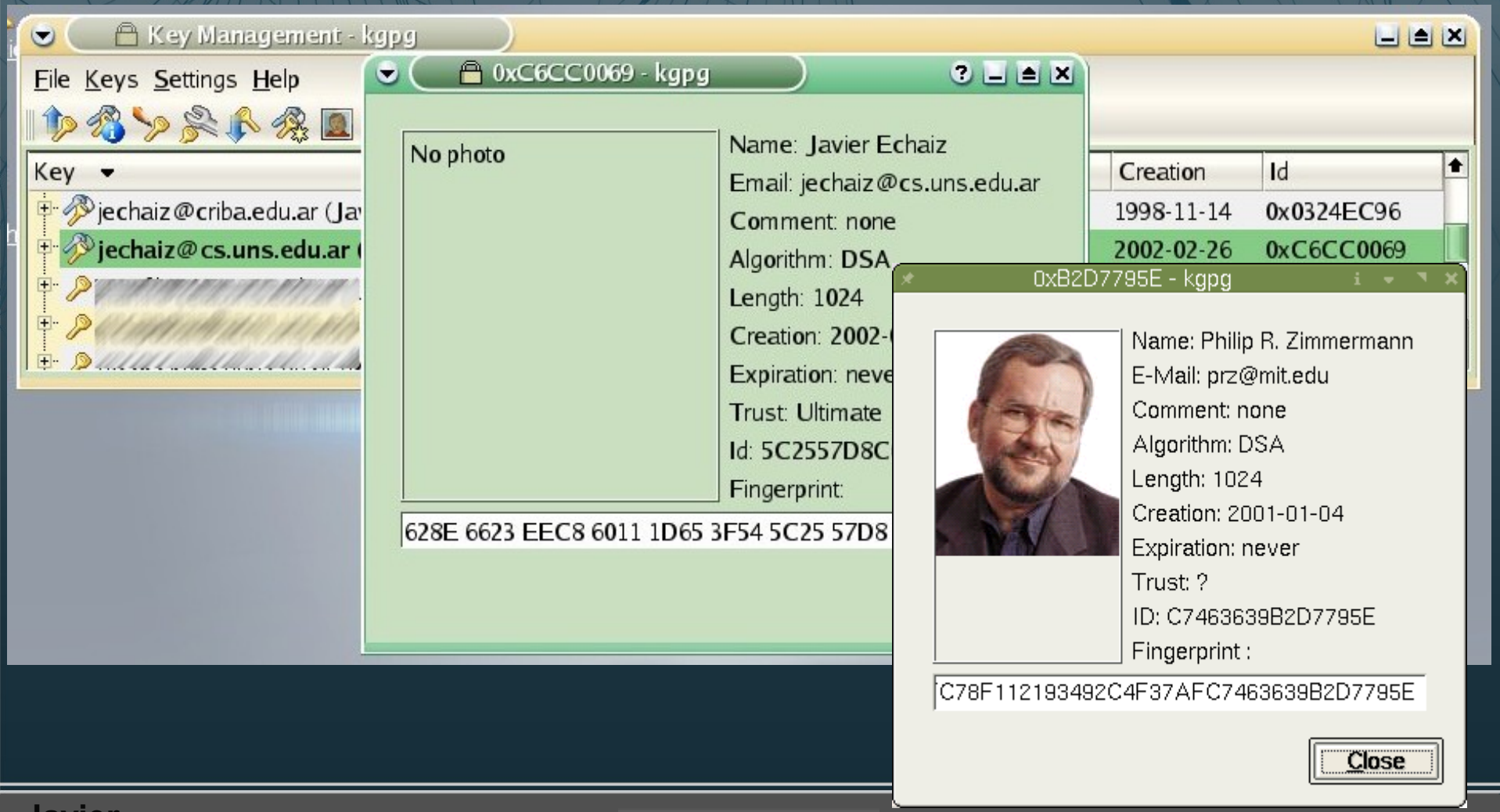
GPG – Clipboard Decryption



GPG – Key Management

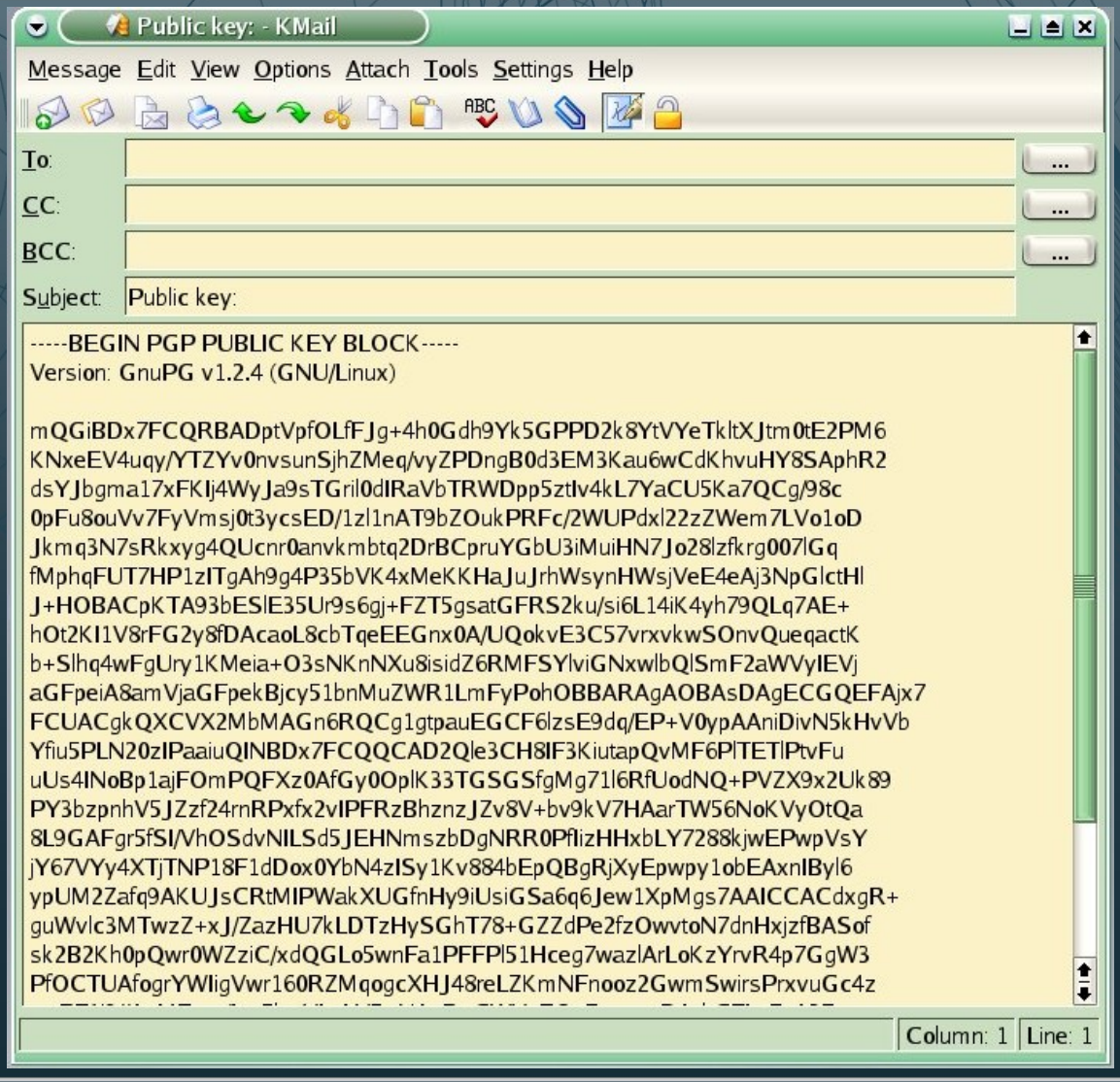


GPG – Key Management

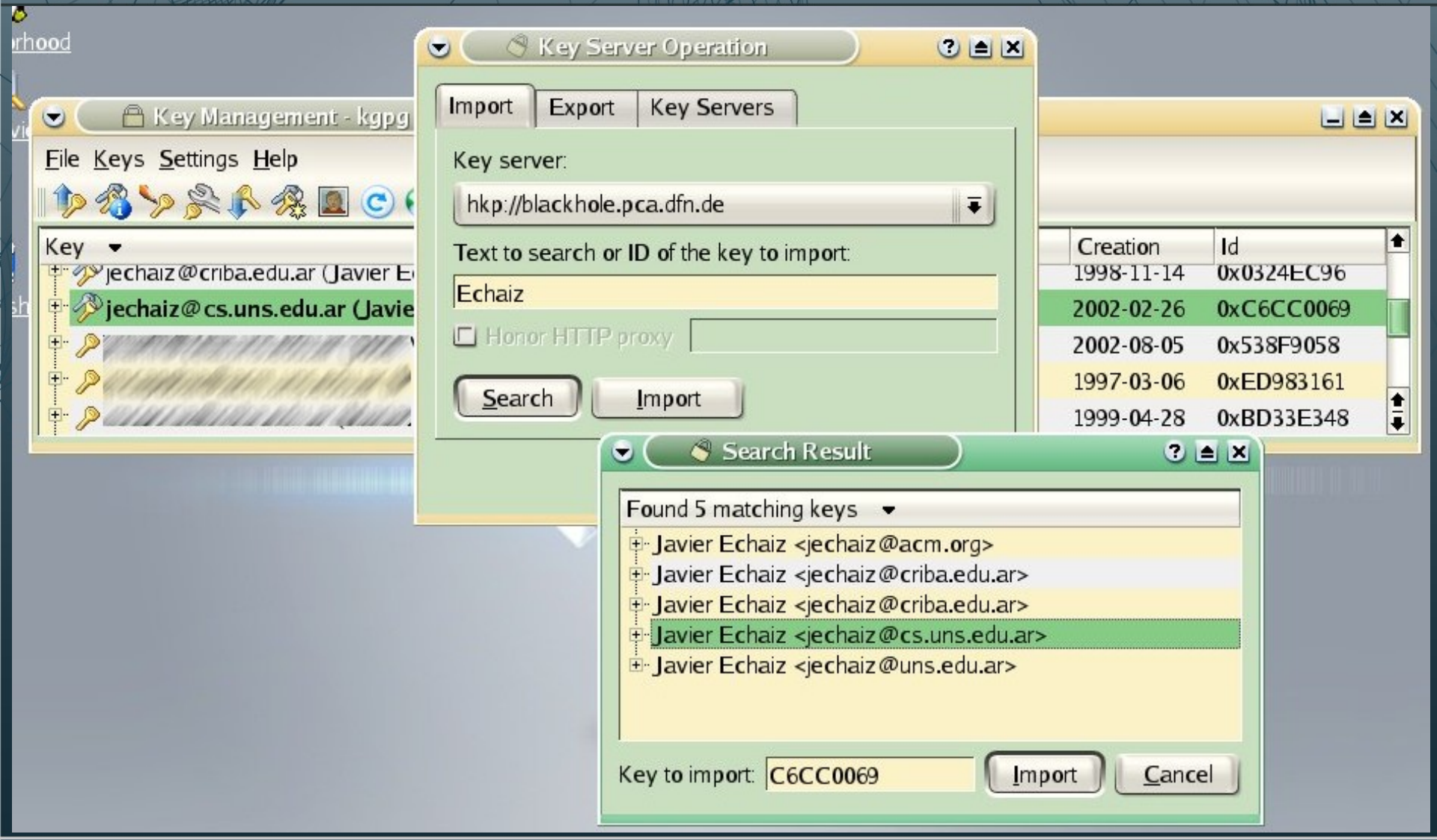


GPG – Key Management

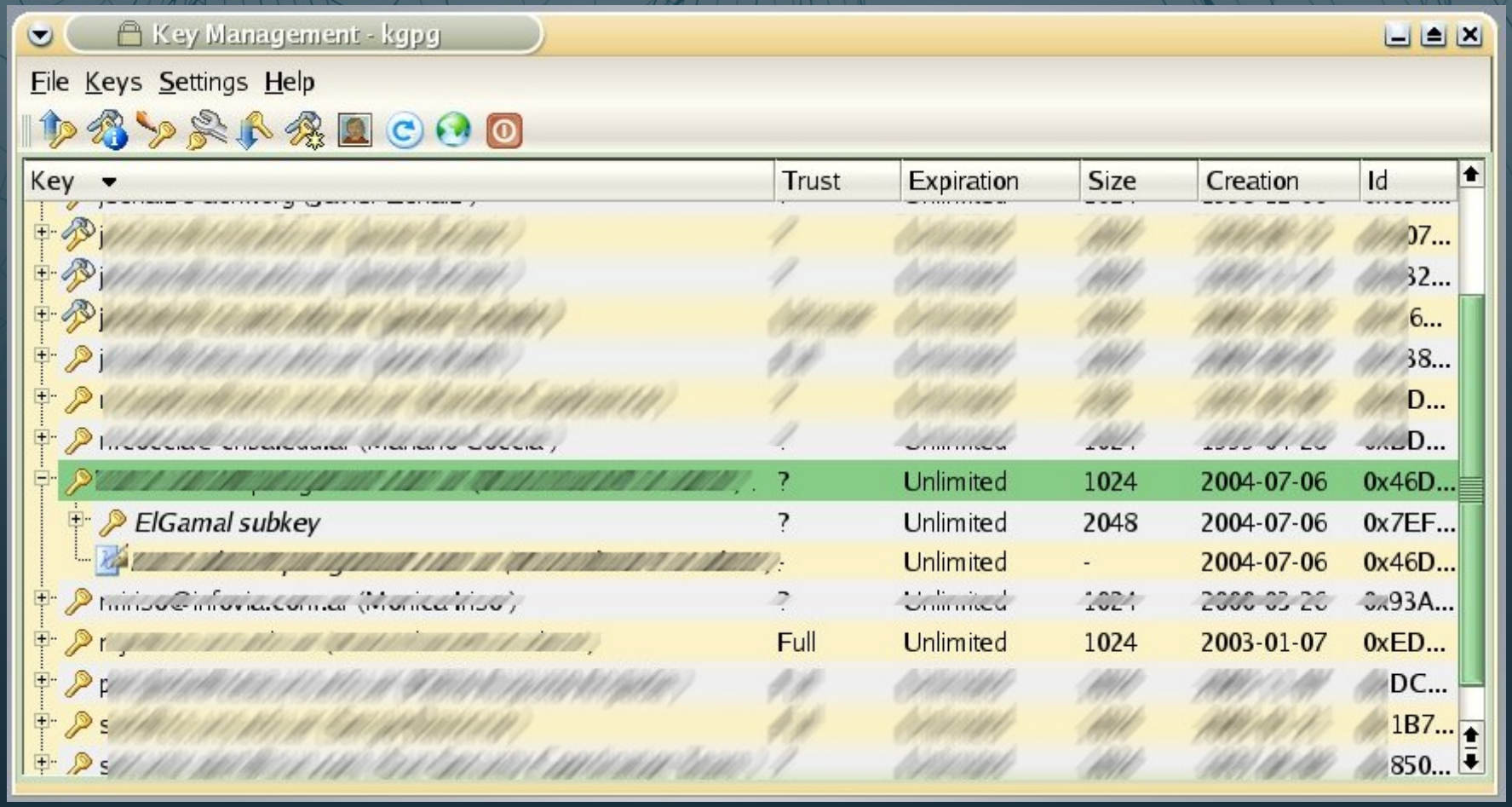
GPG – Key Management



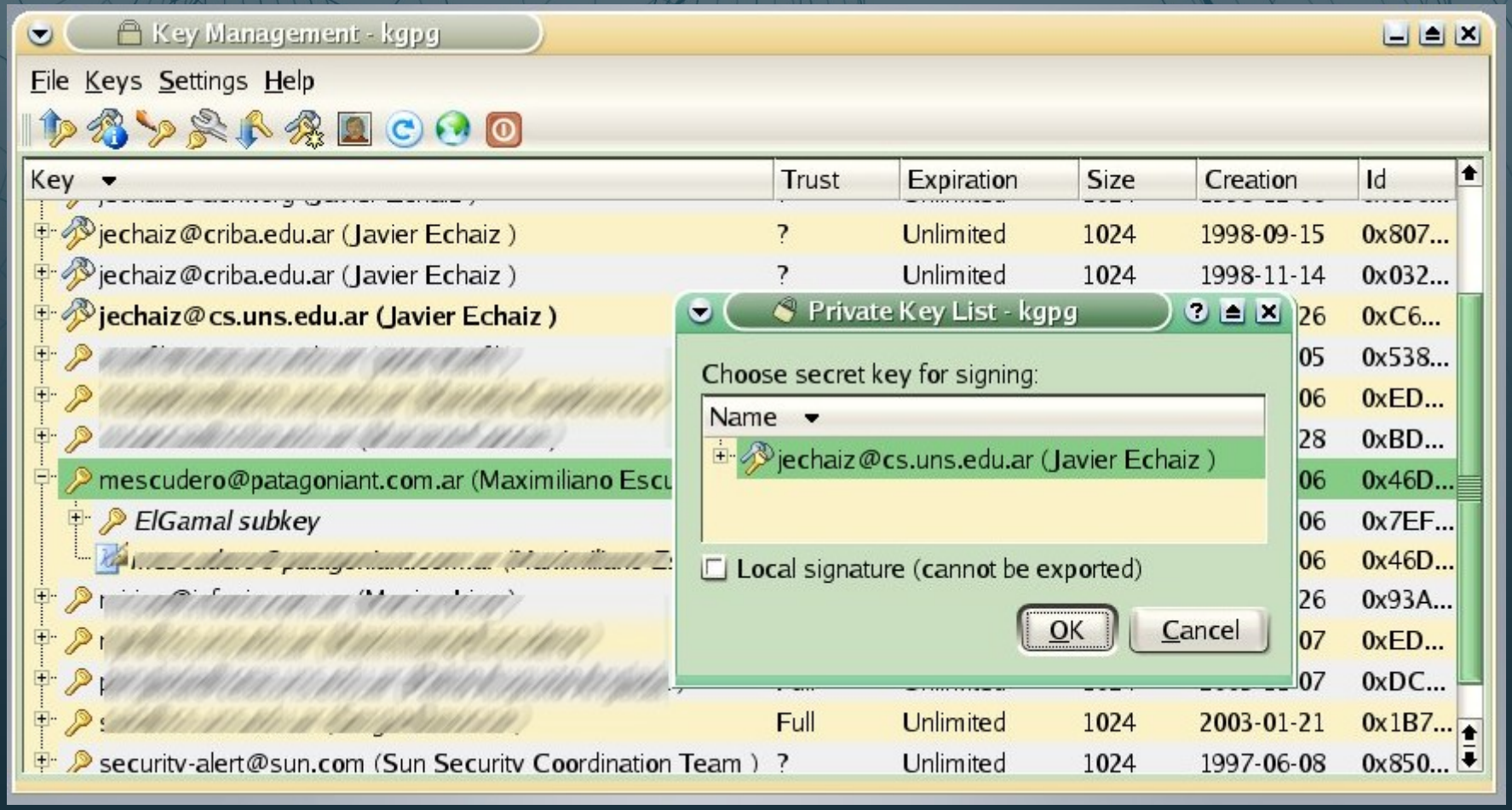
GPG – Key Management



GPG – Firma de Clave

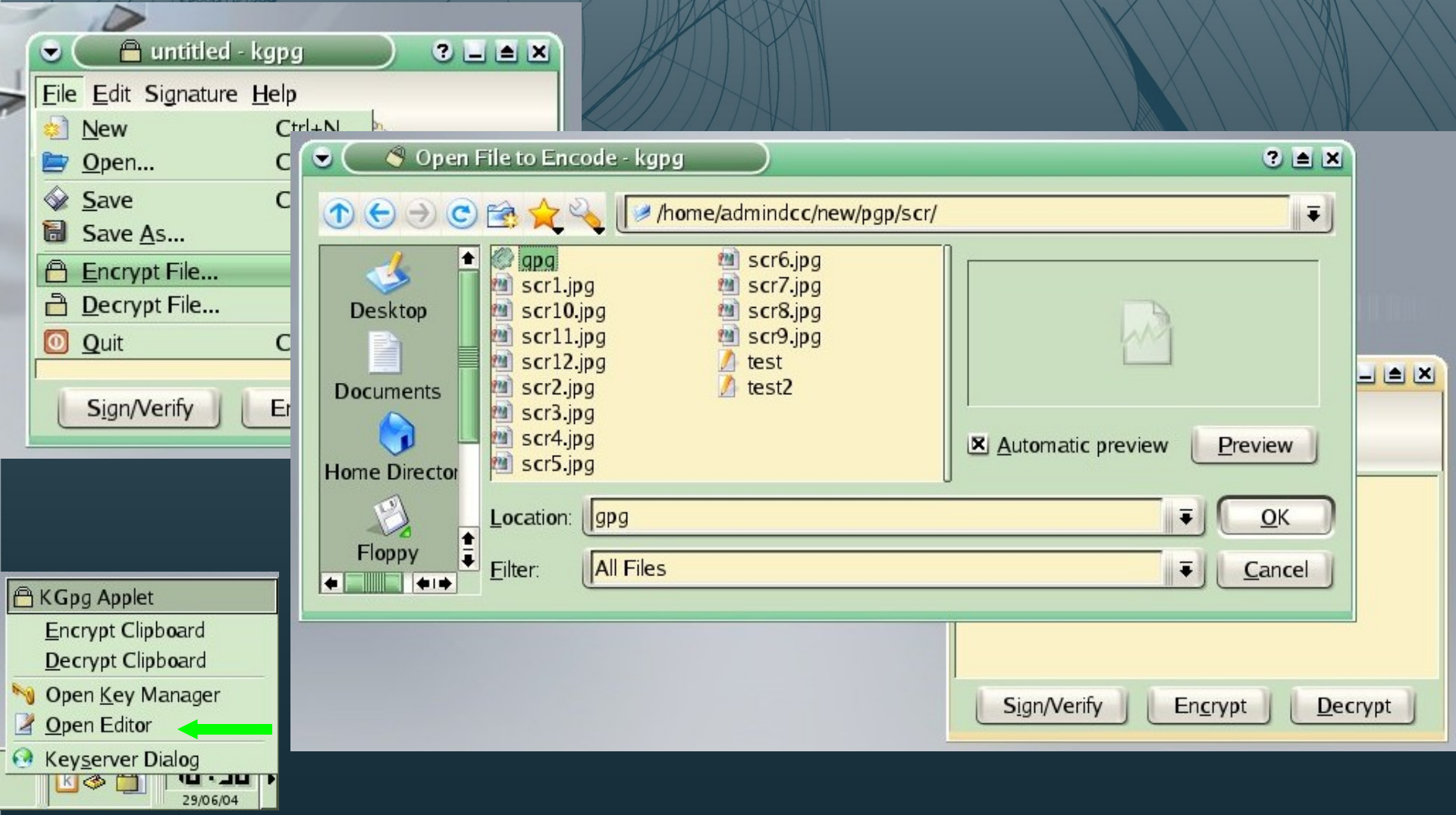


GPG – Firma de Clave

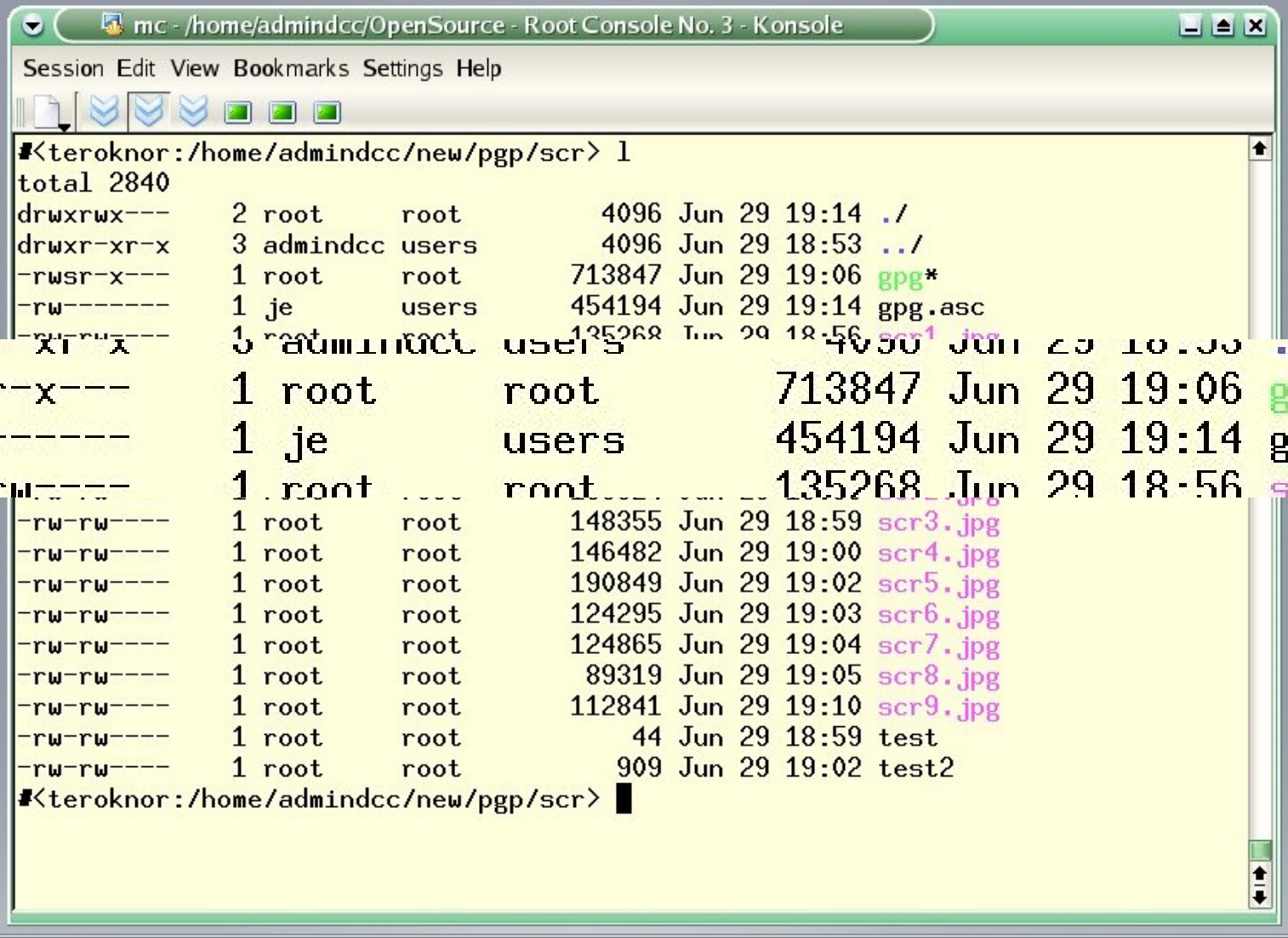


GPG – Firma de Clave

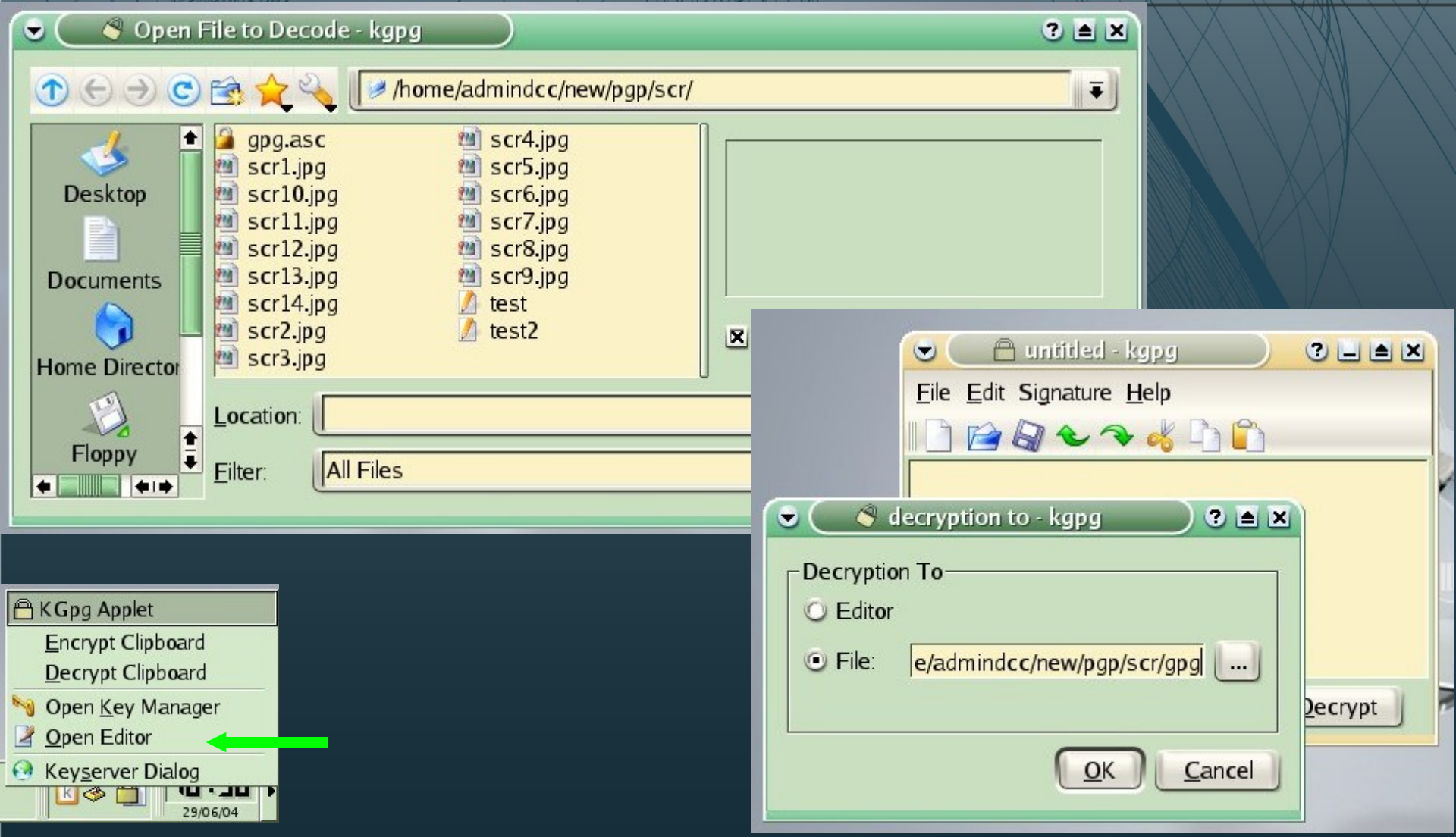
GPG – File Encryption



GPG – File Encryption



GPG – File Decryption



GPG – File Decryption

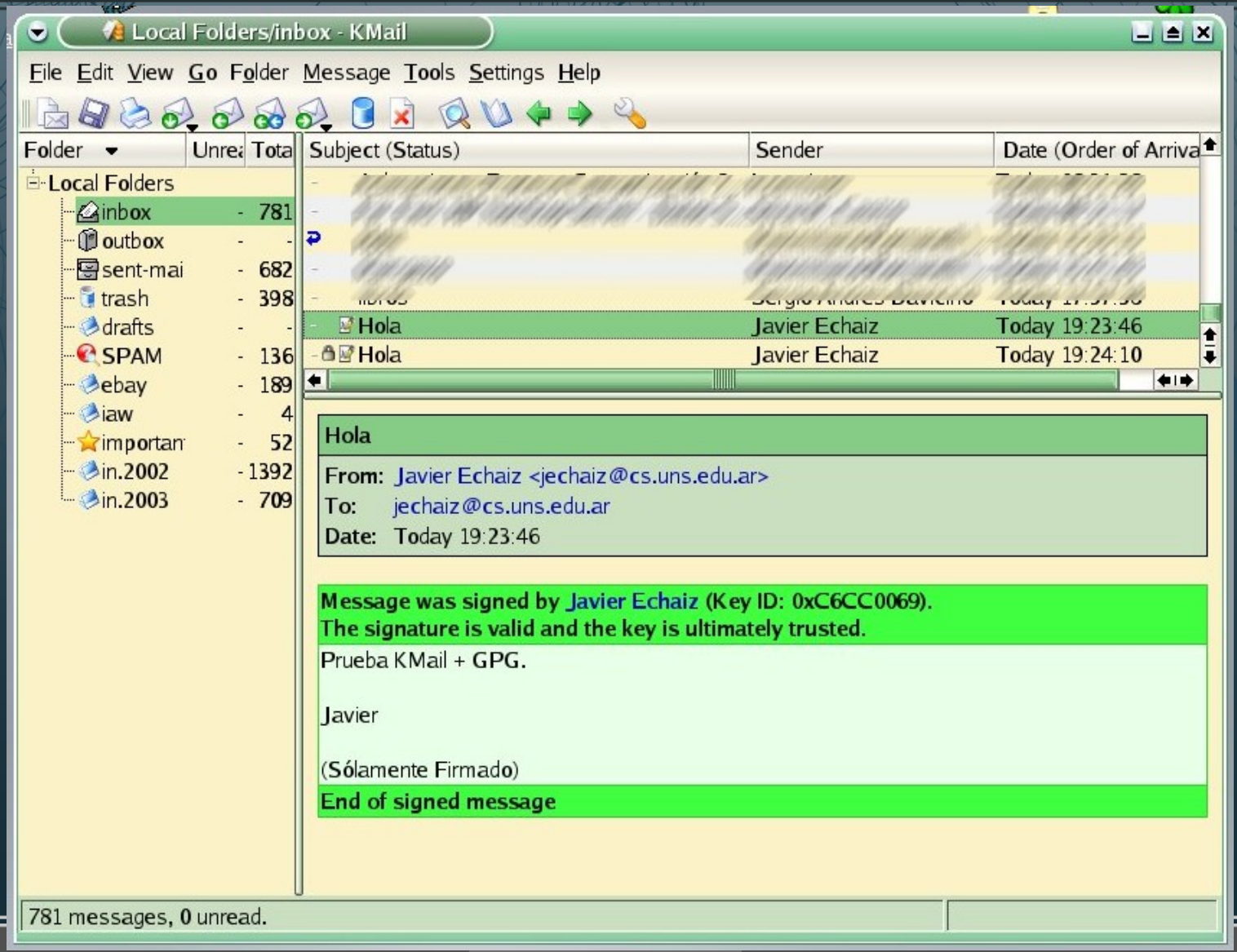
mc - /home/admindcc/OpenSource - Root Console No. 3 - Konsole

Session Edit View Bookmarks Settings Help

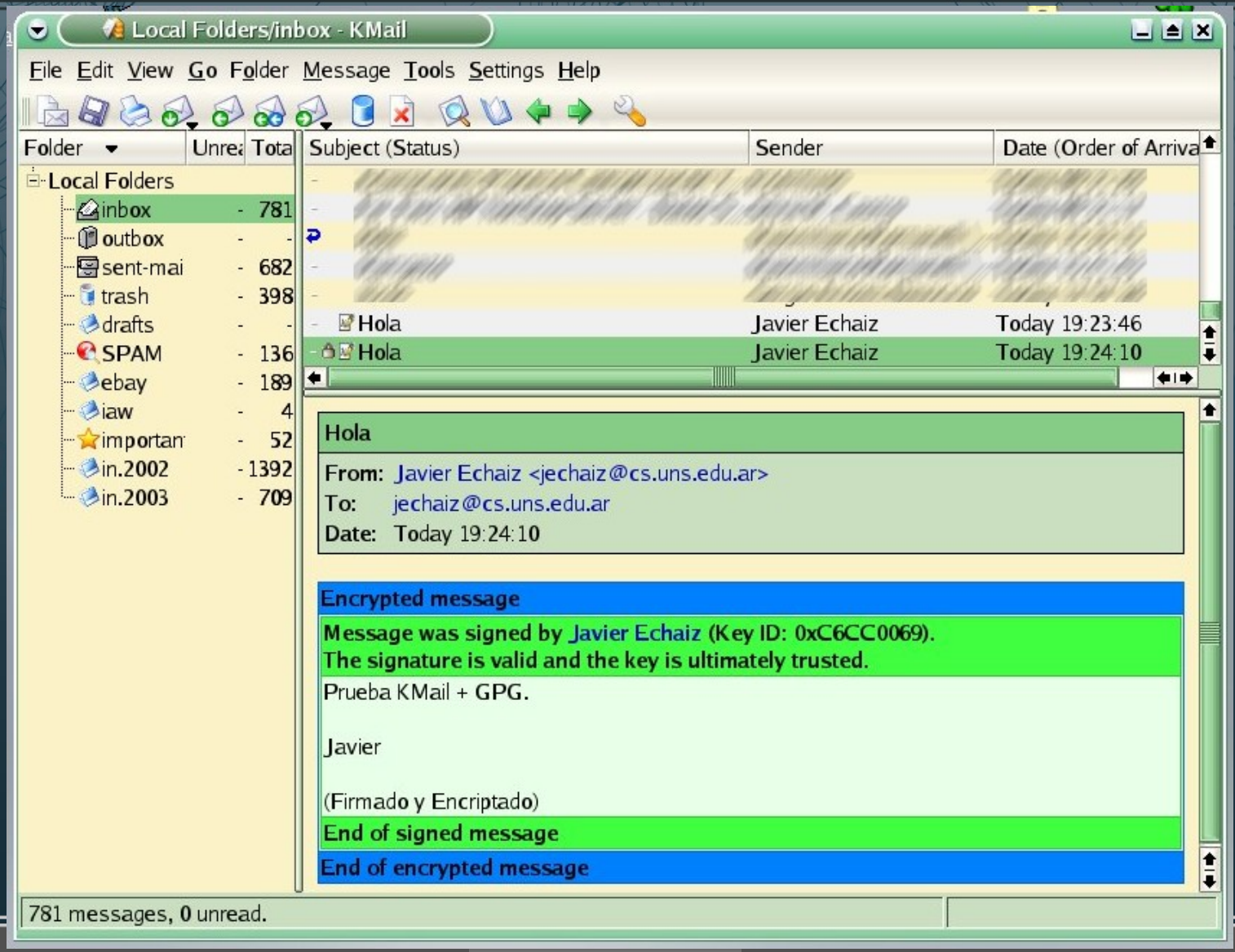
```
#<teroknor:/home/admindcc/new/pgp/scr> ls -la
total 3180
drwxrwx---    2 root     root         4096 Jun 29 19:18 ./
drwxr-xr-x    3 admindcc users        4096 Jun 29 18:53 ../
-rw-------    1 je       users      713847 Jun 29 19:18 gpg
-rw-------    1 je       users      454194 Jun 29 19:14 gpg.asc
-rw-rw----    1 root     root       135268 Jun 29 18:56 scr1.jpg
-rw-rw----    1 root     root       113457 Jun 29 19:09 scr10.jpg
-rw-rw----    1 root     root       134552 Jun 29 19:11 scr11.jpg
-rw-rw----    1 root     root        77433 Jun 29 19:13 scr12.jpg
-rw-rw----    1 root     root        99587 Jun 29 19:14 scr13.jpg
-rw-rw----    1 root     root       163145 Jun 29 19:15 scr14.jpg
-rw-rw----    1 root     root        94720 Jun 29 19:16 scr15.jpg
-rw-rw----    1 root     root        73720 Jun 29 19:16 scr16.jpg
-rw-rw----    1 root     root       136624 Jun 29 18:58 scr2.jpg
-rw-rw----    1 root     root       148355 Jun 29 18:59 scr3.jpg
-rw-rw----    1 root     root       146482 Jun 29 19:00 scr4.jpg
-rw-rw----    1 root     root       190849 Jun 29 19:02 scr5.jpg
-rw-rw----    1 root     root       124295 Jun 29 19:03 scr6.jpg
-rw-rw----    1 root     root       124865 Jun 29 19:04 scr7.jpg
-rw-rw----    1 root     root        89319 Jun 29 19:05 scr8.jpg
-rw-rw----    1 root     root       112841 Jun 29 19:10 scr9.jpg
-rw-rw----    1 root     root           44 Jun 29 18:59 test
-rw-rw----    1 root     root          909 Jun 29 19:02 test2
#<teroknor:/home/admindcc/new/pgp/scr> ls -la =gpg
-rwsr-xr-x    1 root     root       713847 Jun  9 09:40 /usr/local/bin/gpg*
```

Archivo desencriptado a partir de gpg.asc

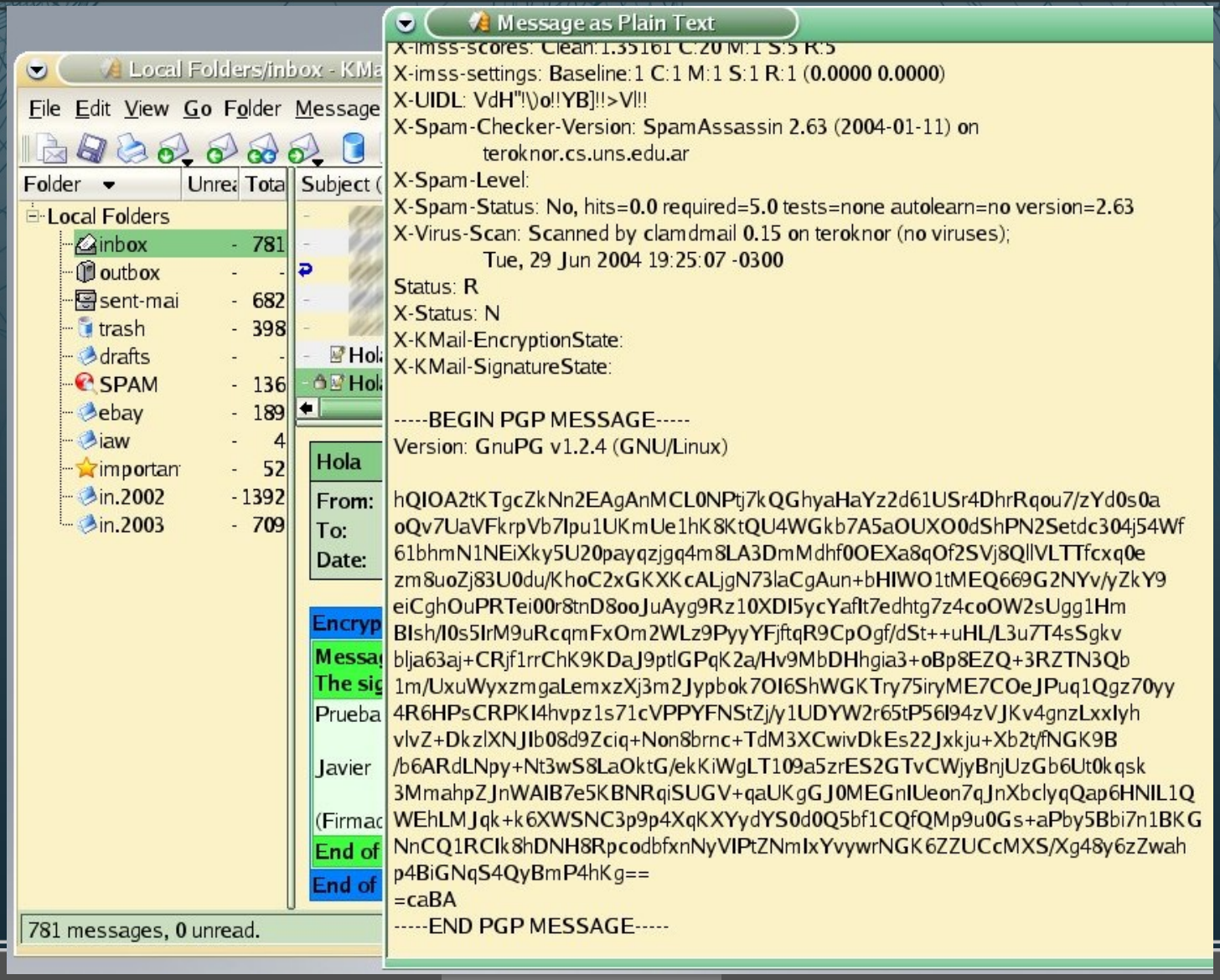
GPG – Integración con KMail (1)



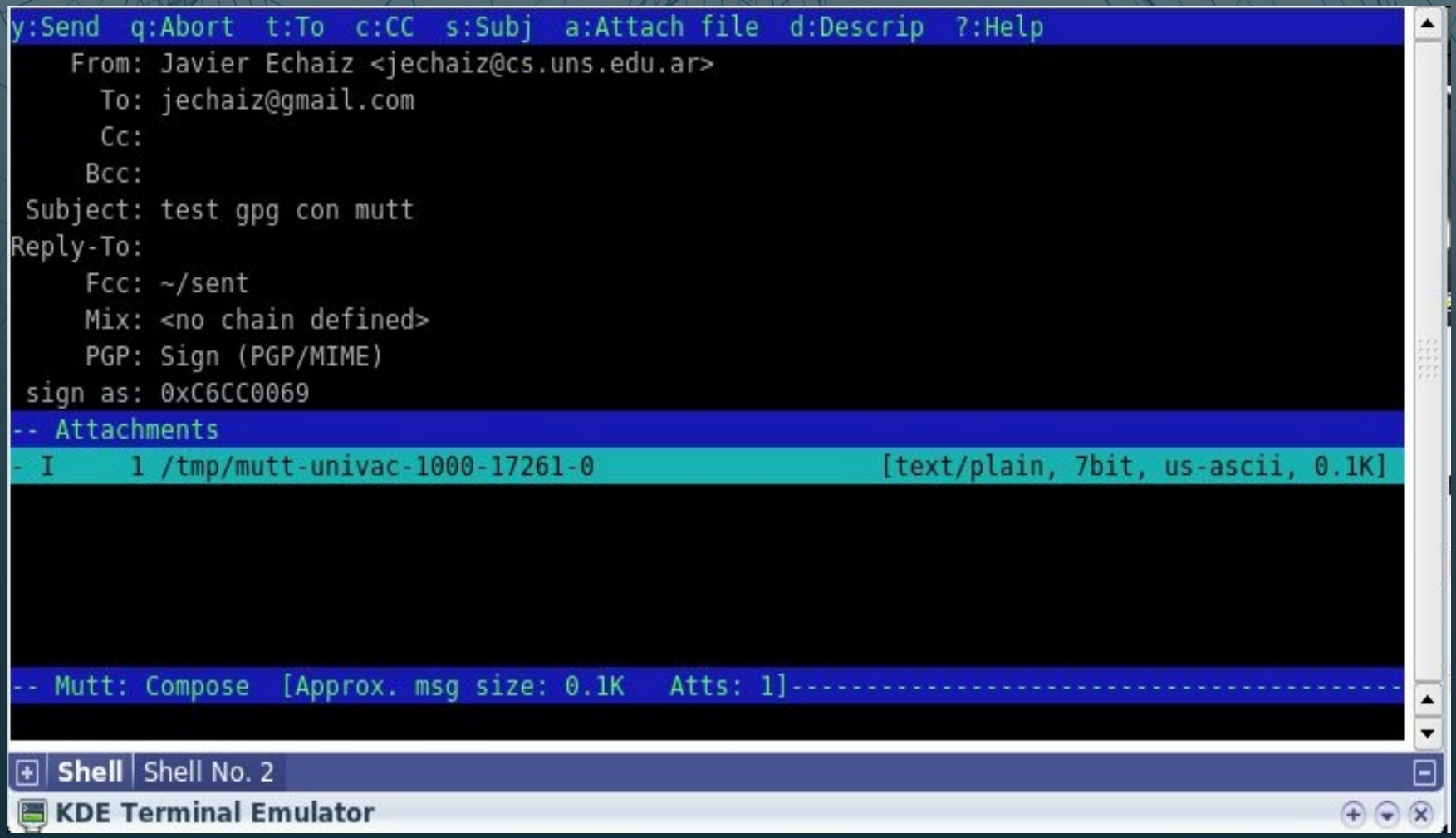
GPG – Integración con KMail (2)



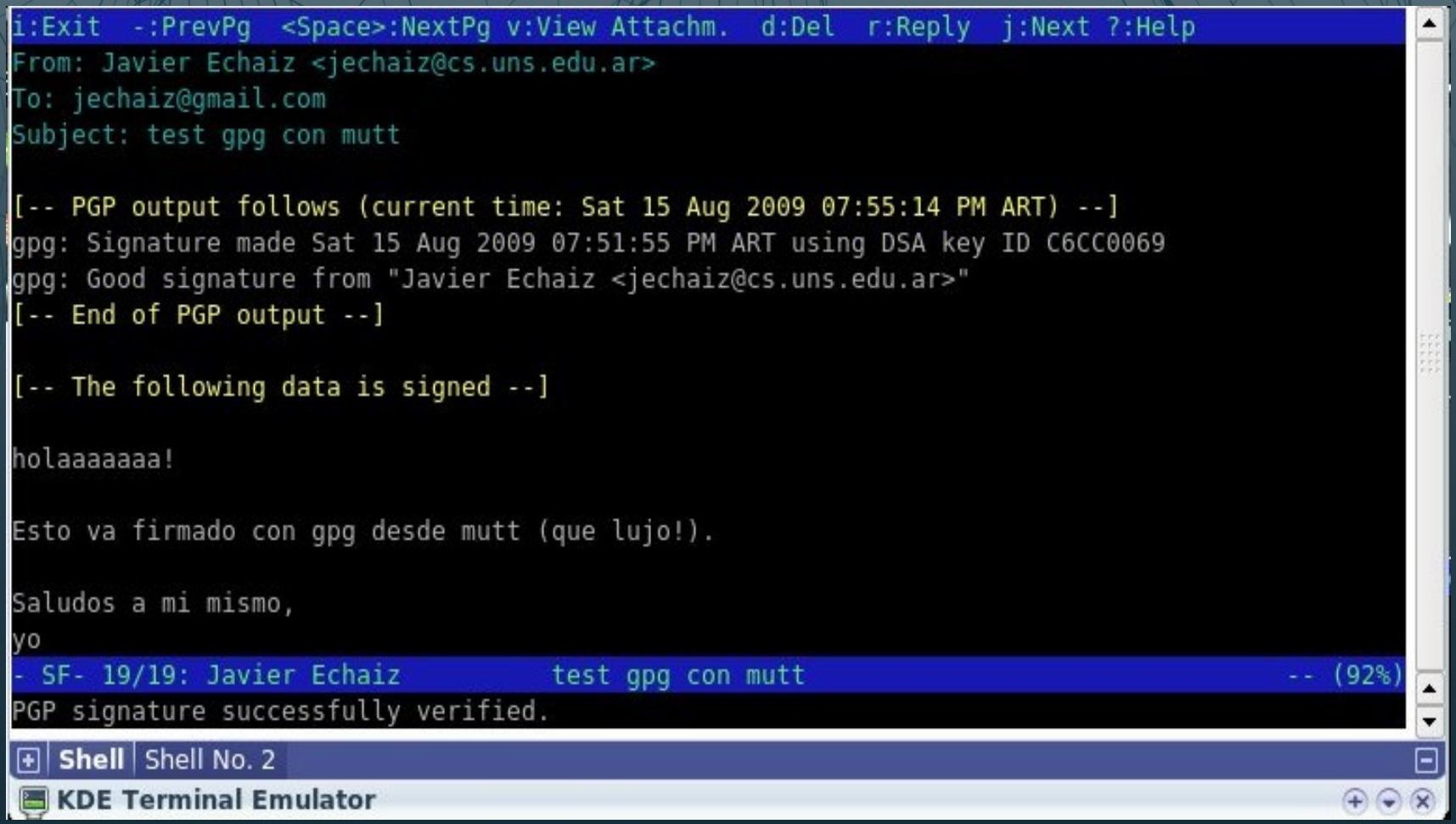
GPG – Integración con KMail (3)



GPG – Integración con mutt (1)



GPG – Integración con mutt (2)



GPG: cómandos + útiles

- gpg --sign [archivo]
- gpg --clearsign [archivo]
- gpg --detach-sign [archivo]
- gpg [-u Remitente] -r Destinatario [--armor] --sign --encrypt [archivo]
- gpg --verify [archivo]
- Otros comandos útiles:
- --fingerprint --list-sigs --list-keys –import –export –list-secret-keys (usar por separado!)
- --edit-key UID (por ejemplo Command> sign)

Más???


Interfaces de Usuario - GnuPG.org - Chromium
Javier Echaiz - My PGP ...
Interfaces de Usuario -...
http://www.gnupg.org/related_software/frontends.es.html
GnuPG
Deutsch · English · Español · Français · Italiano
Réplicas (Mirrors)
Contenidos
Interfaces de Usuario
Interfaces gráficas (GUIs)
Programas de correo-e
Programas de charla en tiempo real (chat)
Interfaces relacionados con la red
Interfaces para programar guiones (scripts)
Para plataformas *nix
Para plataformas Windows
Para plataformas Mac
INTERFACES DE USUARIO
Parece que la página original es más reciente que esta traducción.
Aquí hay algunas listas de programas con soporte para GnuPG.
Para su comodidad, las interfaces de usuario han sido agrupadas en categorías homogéneas. Una interfaz puede pertenecer a una o más categorías a la vez. Cada lista está ordenada alfabéticamente.
Interfaces gráficas (GUIs)
Programas de correo-e
Programas de charla en tiempo real (chat)
Interfaces relacionados con la red
Interfaces para programar guiones (scripts)
Para plataformas *nix
Para plataformas Windows
Para plataformas Mac
Si se siente con ganas de mejorar estas listas, por favor, rellene un formulario de Informe de Fallo .

Todavía Más???

- Puedo firmar binarios?
  - Elfsign (desactualizado) o bsign (reemplaza TripWire / Afick).
  - bsign --sign /tmp/bin/ls
  - bsign -V /tmp/bin/ls
  - **Nota: cuidado, modifica ejecutables!!!!**
- Puedo firmar PDFs?
  - Sinadura (www.sinadura.net)

S/MIME

- Secure/Multipurpose Internet Mail Extension.
- Será probablemente el estándar de la industria.
- PGP estándar para seguridad en e-mails personales.

S/MIME, Circa 1998

- RSA Data Security promueve el estándar “S/MIME”.
- Identificación basada en Certificados.
- S/MIME incorporado en:
  - M$ Outlook Express.
  - M$ Outlook.
  - Lotus Notes.
  - … y por supuesto gpg! ;)

CN: Phil Zimmermanl
DN: prz@mit.edu

CN: Javier Echaiz
DN:je@cs.uns.edu.ar

Headers en MIME

http://mailformat.dan.info/headers/mime.html

- **MIME-Version:** debe ser “1.0”
  - RFC 2045, RFC 2046.
- **Content-Type:** (application/word).

  multipart/signed
  multipart/encrypted
  } Relacionados con Seguridad

- **Content-Transfer-Encoding:** indica como se codificó el mensaje (radix-64)
- **Content-ID:** string que identifica unívocamente el tipo de contenido.
- **Content Description:** necesario cuando el contenido no es legible (e.g., mpeg).

Funcionalidad de S/MIME

- **Enveloped Data:** se encripta el contenido y las claves de sesión para uno o más receptores.
- **Signed Data:** el MD (Message Digest) se encripta con la clave privada del emisor. Se convierte a radix-64.
- **Clear-Signed Data:** se firma pero no se encripta el mensaje.
- **Signed and Enveloped Data:** varias formas de encriptar y firmar.

Algoritmos en S/MIME

- **Message Digesting:** SHA-1 y MD5
- **Digital Signatures:** DSS
- **Secret-Key Encryption:** Triple-DES, RC2/40 (exportable).
- **Public-Private Key Encryption:** RSA con claves de entre 512 y 1024 bit y Diffie-Hellman para claves de sesión.

Key-Management en S/MIME

- S/MIME usa Certificados de Clave Pública X.509 versión 3 firmado por una CA (Certification Authority).
- Funciones:
  - **Key Generation** - Diffie-Hellman, DSS y pares de claves RSA.
  - **Registration** - las claves públicas deben registrarse en una CA para recibir un certificado de clave pública X.509.
  - **Certificate Storage** - Local (como en una aplicación browser) para diferentes servicios.
  - **Signed and Enveloped Data** – Varias formas de encriptar y firmar.

Certificados UIT-T X.509

- La idea básica es tener datos acerca de una persona u organización en un formato estándar.
- Alguna organización confirma que la información es verdadera.
- Esa organización firma el certificado, igual que un escribano público.
- Tenemos entonces una confirmación por parte de un tercero (en el cual confiamos) acerca de la veracidad de la clave.
- Podemos obtener certificados X.509 de distintos lugares, uno es Verisign.

Ejemplo de CA

- **Ejemplo: Verisign** http://www.verisign.com
  - **Class-1:** la dirección de e-mail del comprador se verifica enviándole información “vital”.
  - **Class-2:** también se verifica la dirección postal.
  - **Class-3:** el comprador debe apersonarse, o enviar documentos firmados por un escribano.

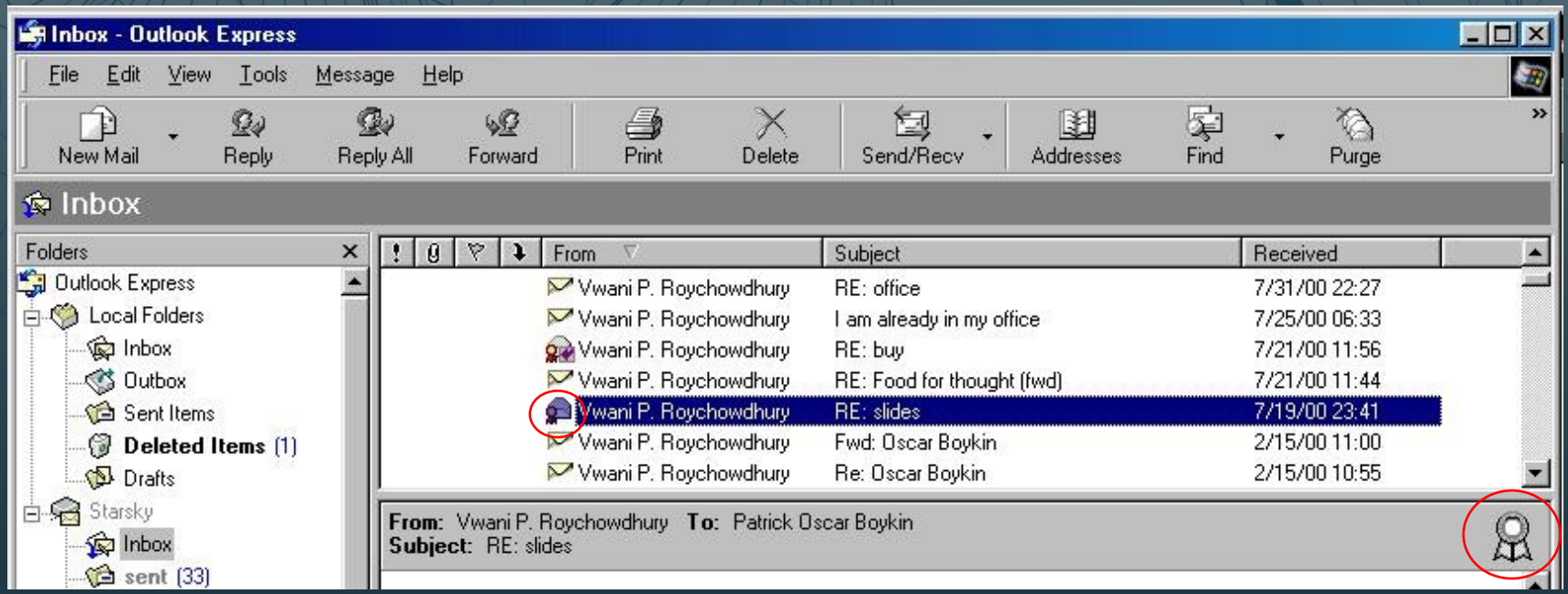
Verificación de Certificados

- Dodo un certificado comprobamos su validez de la siguiente forma:

1. Comprobamos su caducidad.
2. Comprobamos qué CA lo expidió.
3. Comprobamos el certificado de la CA, así como todos los certificados de CA's superiores que haya, si se da el caso, certificado a la CA anterior.
4. Comprobamos que el certificado no fue manipulado, comprobando la firma digital de la CA.
5. Comprobamos CRL de la CA para verificar que no ha sido revocado.

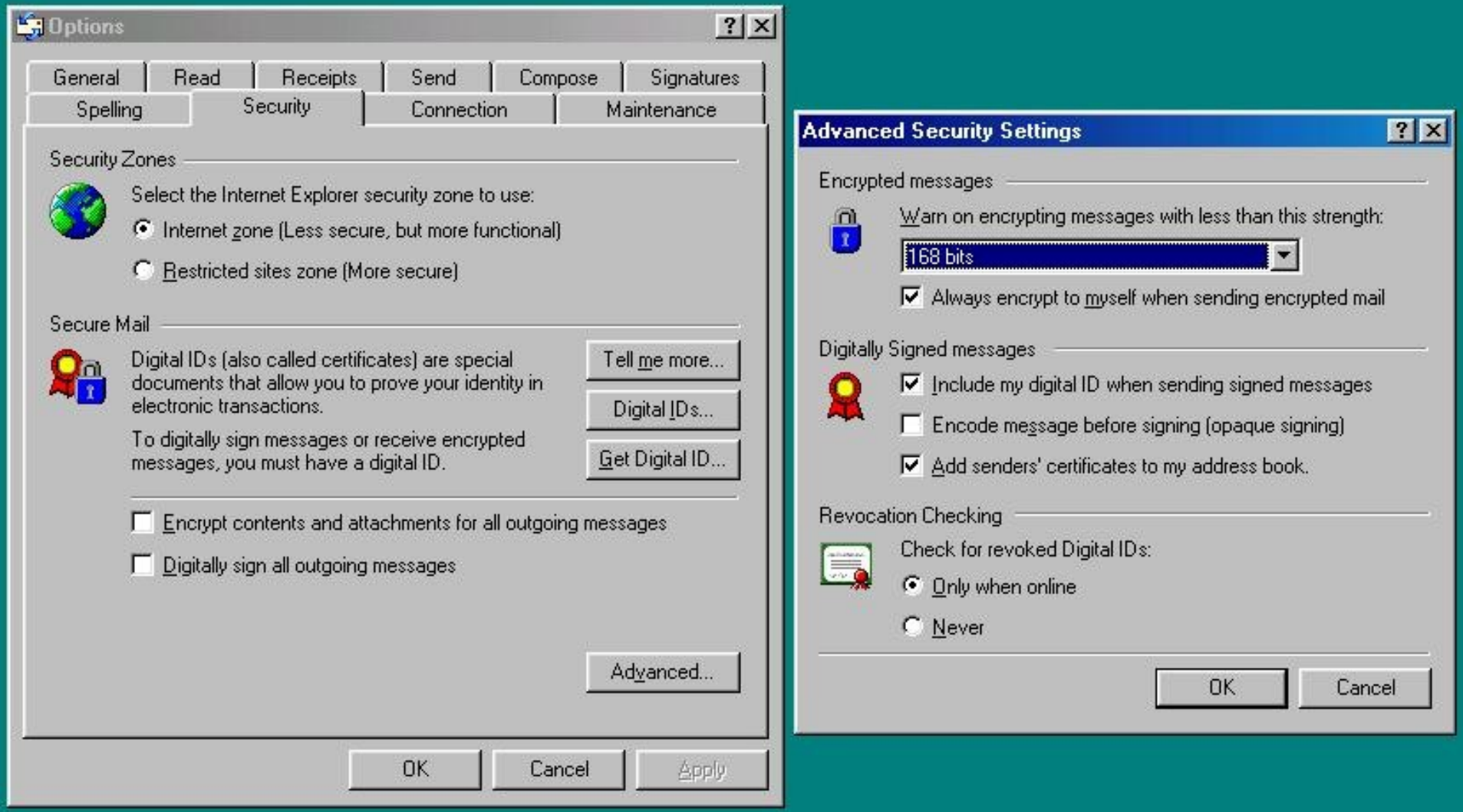
Después de todo esto podemos establecer una conexión "segura" y "autenticada".

Firma con S/MIME (M$)



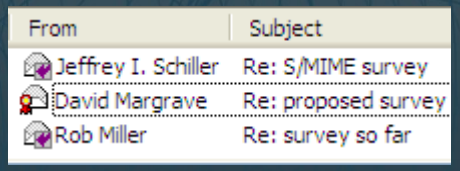
Nótese el “ribbon” que denota que el mensaje está firmado digitalmente.

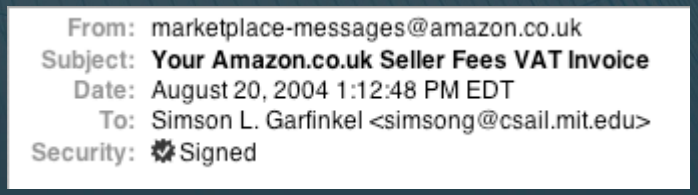
Opciones S/MIME en (M$) Outlook



Firmas Digitales HOY

- El soporte de S/MIME es casi universal.
- Funciona bárbaro si la Certificate Authority es conocida:





- Horrible si la CA es desconocida.
- Problema: Los usuarios no pueden crear sus propios certificados; tienen que *obtenerlos*.

PGP vs. S/MIME

- PGP: Web of Trust.
  - El certificado puede ser firmado por muchos.
  - Grados de confianza.
- S/MIME: No define modelo de confianza.
  - Implementa casi cualquier modelo.
  - Certificados X.509.
  - Los certificados no pueden firmarse más de una vez.

¿Cuál elijo?

S/MIME en GPG?

» Claro que sí!

DESCRIPTION

gpgsm is a tool similar to gpg to provide digital encryption and signing servicesd on X.509 certificates and the CMS protocol. It is mainly used as a backend for S/MIME mail processing. gpgsm includes a full features certificate management and complies with all rules defined for the German Sphinx project.

```
$univac:/tmp> gpgsm --version
gpgsm (GnuPG) 2.0.7
Copyright (C) 2007 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: ~/.gnupg
Supported algorithms:
Cipher: 3DES, AES, AES192, AES256, SERPENT128, SERPENT192, SERPENT256
Pubkey: RSA
Hash: MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512
$univac:/tmp>
```

Conclusiones

- La Seguridad es un elemento fundamental.
- El servicio de e-mails es uno de los servicios más importantes y más usados.
- También puede agregar seguridad pgp a chat, PDF's, php, etc.
  - La tecnología necesaria EXISTE.

Gracias...

Puede bajar estas slides

http://cs.uns.edu.ar/~jechaiz/pgp.pdf

jechaiz@cs.uns.edu.ar

Backup Slides

Estándar PKCS

- PKCS: Public-Key Cryptography Standards. Conjunto de especificaciones técnicas desarrolladas por Netscape, RSA y otros desarrolladores de informática con el objeto de uniformizar las técnicas de criptografía pública.
- Publicación de la primera versión 1.0 en el año 1991.
- PKCS forma parte de distintos estándares de hecho como ANSI X9, PKIX, SET, S/MIME y SSL.
- A la fecha existen 14 documentos con títulos genéricos que van desde PKCS #1 a PKCS #15. Los puede bajar desde el servidor http://www.rsasecurity.com/rsalabs/node.asp?id=2124.
- Mantendremos los títulos originales en su versión en inglés de RSA Security Inc. Public-Key Cryptography Standards PCKS.

Documentos del estándar PKCS

- PKCS #1: RSA Cryptography Standard
- PKCS #2: Incluido ahora en PKCS #1
- PKCS #3: Diffie-Hellman Key Agreement Standard
- PKCS #4: Incluido ahora en PKCS #1
- PKCS #5: Password-Based Cryptography Standard
- PKCS #6: Extended-Certificate Syntax Standard
- PKCS #7: Cryptographic Message Syntax Standard
- PKCS #8: Private-Key Information Syntax Standard
- PKCS #9: Selected Attribute Types
- PKCS #10: Certification Request Syntax Standard
- PKCS #11: Cryptographic Token Interface Standard
- PKCS #12: Personal Information Exchange Syntax Standard
- PKCS #13: Elliptic Curve Cryptography Standard
- PKCS #15: Cryptographic Token Information Format Standard

Links acerca de PGP

- OpenPGP
  - http://www.openpgp.org/
- How PGP works?
  - http://www.pgpi.org/doc/pgpintro/
- Tom McCune's PGP Questions & Answers
  - http://www.mccune.cc/PGPpage2.htm
- PGP - Getting Started
  - http://net.gurus.com/pgp/
- RSA
  - http://www.rsasecurity.com/
- Why OpenPGP's PKI is better than an X.509 PKI
  - http://www.openpgp.org/technical/whybetter.shtml

Trust Models

- Direct Trust
  - I trust your cert because you gave it to me
  - Very secure trust model (do you trust yourself)
  - Scales least well
  - Used in OpenPGP, S/MIME, IPsec, TLS/SSL, etc.

Trust Models

- Hierarchical Trust
  - I trust your cert because its issuer has a cert issued by someone … whom I trust
  - Least secure trust model
    - Damage spreads through tree
    - Recovery is difficult

Trust Models

- Hierarchical Trust (continued)
  - Best scaling, mimics organizations
  - Used in OpenPGP, S/MIME, IPsec, TLS/SSL, etc.

Trust Models

- Cumulative Trust (a.k.a. Web of Trust)
  - I trust your cert because some collection of people whom I trust issued certifications
  - Potentially more secure than direct trust
  - Scales almost as well as HT for intra-organization

Trust Models

- Cumulative Trust
  - Handles inter-organization problems
    - Company A issues only to full-time employees
    - Company B issues to contractors and temps
    - A and B's management issue edict for cross certification
  - Addresses "two id" problem
    - How do you know John Smith(1) is John Smith(2)?

So What?

- X.509 is everywhere
  - OpenPGP is small (code and data)
    - Zerucha imp. is 5000 lines of C (sans crypto)
  - Suitable for embedded & end-user applications
    - Used by banks, etc. transparently
  - It's Flexible and Small!
  - It actually works

Why Bother?

- S/MIME will take over
  - PGP has years of deployment
    - 90%? Traffic is some PGP.
  - PGP is only strong crypto
    - S/MIME 3 is much better
    - Outside the US, there is distrust
    - Can you see the source?
  - Cisco: Secure email is PGP's to lose

Myths

- It's email only
  - It's for any "object"
- It requires the web of trust
  - Can use any trust model
  - Businesses use PGP with hierarchies today
- It's proprietary
  - IETF Standard

Present Into The Future

- Ultimately, data formats are less important than you'd think
- On desktops, size matters less
  - But small systems will be with us always
- Description of the OpenPGP philosophy
  - PGP implemented in X.509
  - Certification Process

OpenPGP Philosophy

- Everyone is *potentially* a CA
  - This is going to happen whether you like or not.
- Everyone has different policies
  - Wait until you do inter-business PKI
- One size will not fit all
  - Validity is in the eye of the beholder
  - Trust comes from below

Potential PGP/X.509 merger

- Ideas of PGP
- Syntax of X.509
- Disclaimer
  - This doesn't exist
  - It's all still experimental