UNPSJB

LICENCIATURA EN SISTEMAS OPGCPI

ADMINISTRACIÓN DE REDES Y SEGURIDAD

Trabajo Práctico 3

Criptografía

Cátedra
Lic. Bruno Damián Zappellini

Integrantes:
Luciano Serruya Aloisi

9 de noviembre de 2018

Índice

1. Conceptos básicos

Juan quiere mandarle un mensaje a Julio. A Julio no le importa asegurarse que el mensaje fue enviado por Juan, sin embargo Juan quiere estar seguro de que el mensaje no podra ser leído ni alterado por un tercero. Juan trabaja en una empresa en Argentina y Julio es empleado de una empresa ubicada en España

Para esta situación, un sistema de cifrado simétrico no cumpliría los requisitos solicitados; en el caso de que la clave compartida sea interceptada, la comunicación puede ser espiada por un tercero. Sin embargo, si se garantiza un canal seguro para transmitir la clave privada al inicio de la comunicación, sí se podría utilizar un cifrado simétrico. En caso de que esto último no sea posible, la alternativa sería usar un sistema de cifrado asimétrico, donde el emisor encripta el mensaje con la clave pública del receptor, y el receptor desencripta el mensaje recibido con su clave privada.

Adriana y Leandro quieren comunicarse en forma segura. Para ellos resulta facil conseguir un medio seguro para intercambiar informacion que luego necesiten para realizar esta comunicacion segura. En este caso lo que importa es que nadie pueda espiar los datos involucrados en dicha comunicación

Debido a que se cuenta con un canal seguro, se lo podría utilizar para compartir una clave, la cual se utilizará para cifrar los mensajes intercambiados de forma simétrica.

Analía usara el correo electrónico para enviar la aceptación de un contrato al estudio en el cual trabaja. Para la persona que lo reciba es importante tener la garantía de que el mismo fue enviado efectivamente por Analía

Lo adecuado para esta situación sería que el correo enviado por Analía sea firmado utlizando la clave privada de Analía.

1.1. Verdadero o falso

En los criptosistemas simétricos no puede garantizarse el no repudio porque ambas partes de la transacción conocen la clave utilizada

Falso. Si la clave compartida no fue transmitida por un canal seguro puedo haber sido

interceptada, logrando que un tercero envíe un mensaje encriptado.

Si únicamente me importara la eficiencia del método que uso para encriptar, debería optar por un algoritmo de cifrado asimétrico

Falso. Si se busca eficiencia, la encriptación asimétrica se debería evitar. La encriptación simétrica es mucho más rápida y no incrementa el tamaño del mensaje.

Con ambos tipos de criptosistemas necesito contar con un mecanismo seguro para transmitir la clave

Falso. Con los criptosistemas asimétricos, no es necesario que las claves públicas sean compartidas de forma segura (se pueden alojar en un servidor o un repositorio para que disintios usuarios la consigan).

2. *PKI*

Para poder verificar la firma de un correo eletrónico, el receptor necesita la **clave pública** del emisor.

Al enviar un mensaje firmado, el emisor genera un *resumen* del mensaje (con alguna función de *hashing*), y luego lo encripta con su **clave privada**, generando así la **firma digital**.

El receptor también genera el resumen del mensaje (con el mismo algoritmo), y desencripta la firma con la **clave pública** del emisor (el receptor debía contar con la clave pública del emisor o la debe poder conseguir ya sea a través de un *certificado* o de un repositorio). Luego compara el resumen que él generó, y el que recibió del emisor. Si son iguales, entonces la firma es considerada válida; si no son iguales, entonces significa que se utilizó una clave distinta para firmar el mensaje, o fue alterado.

Ahora bien, si el mensaje está encriptado, para poder abrirlo el receptor necesita su **clave privada**, ya que el mensaje debió ser cifrado con la **clave pública** del receptor. En caso de que el mensaje hubiese sido cifrado con la **clave privada** del emisor, cualquier tercero que tenga la **clave pública** del emisor podrá desencriptarlo.

2.1. Instalando un certificado en Firefox

- Algoritmo de firma utilizado: `PKCS #1 MD5 With RSA Encryption`
- Cantidad de bits de cifrado: `4096 bits`

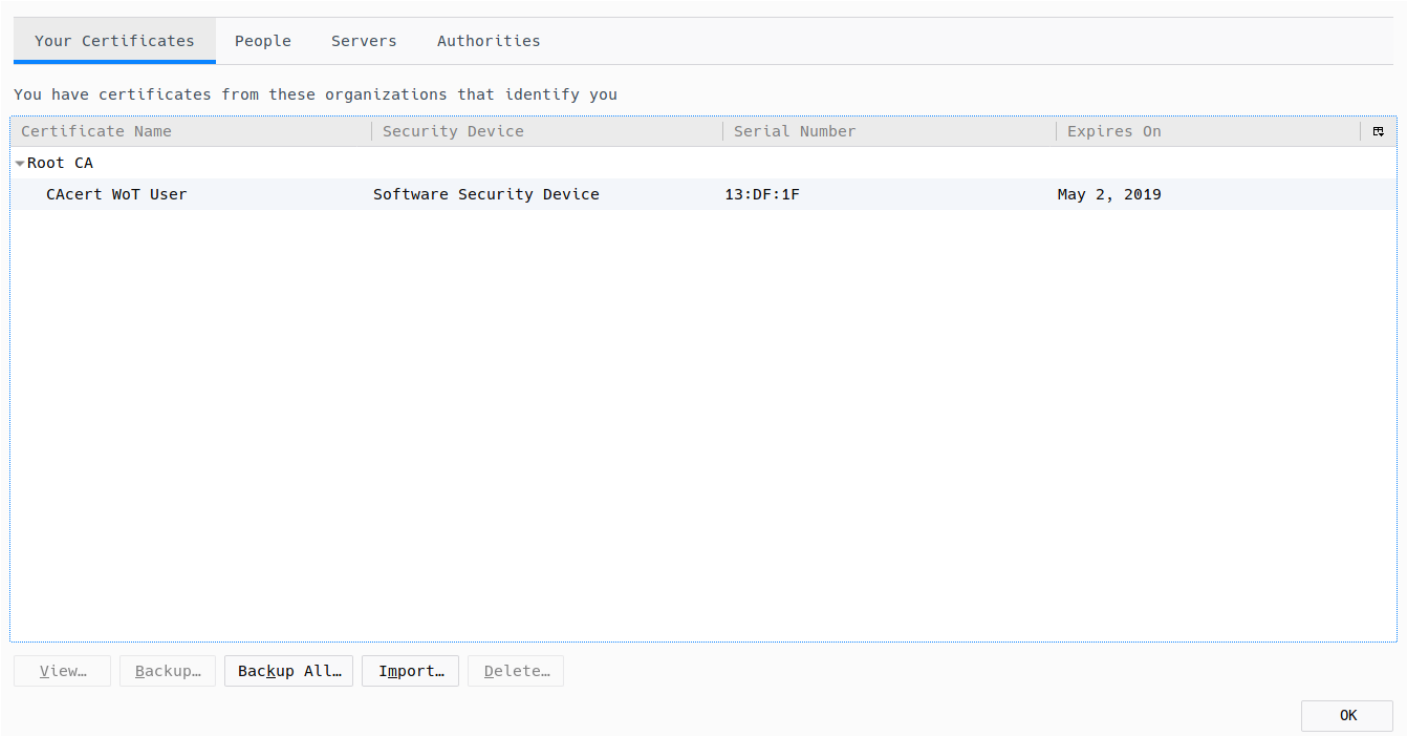
- Periodo de validez: desde el 30/03/2003 hasta el 29/03/2033
- Emisor:
  - E = support@cacert.org
  - CN = CA Cert Signing Authority
  - OU = http://www.cacert.org
  - O = Root CA

Certificado de CACert instalado en Firefox

Ingresando a CACert con HTTPS

2.2. Certificado personal

Después de generar el certificado personal para la dirección de correo electrónico *lucianoserruya@gmail.com* e instalarlo en Firefox, se puede ver el siguiente certificado instalado



Certificado personal instalado en Firefox

2.3. Correos electrónicos firmados y encriptados

Para los siguientes experimentos que se muestran a continuación, se utilizaron dos cuentas personales: lucianoserruya@gmail.com *y* lucianoserruya@hotmail.com. *Para ambas cuentas se generaron los respectivos certificados en CACert y se instalaron correctamente en Thunderbird*

Una vez instalados los certificados de ambas cuentas en Thunderbird, se realizaron los siguientes experimentos:

- Enviar un correo firmado - el emisor necesita su clave privada para encriptar la firma, y el receptor necesita la clave pública del emisor para poder verificarla
- Enviar un correo encriptado - el emisor necesita la clave pública del receptor para encriptar el mensaje, y el receptor necesita su clave privada para poder desencriptarlo
- Enviar un correo firmado y encriptado - se tienen que cumplir las dos condiciones anteriores

A continuación se incluyen capturas de pantalla de los correos electrónicos enviados. Como el mismo cliente tiene tanto las claves privadas como las públicas de ambas cuentas, se pudieron realizar los experimentos sin incovenientes.

Thunderbird indica con un logo de un candado a los correos que hayan sido encriptados, y con un sobre a aquellos que fueron firmados (y que se pudo validar la firma).

Correo firmado

Correo encriptado

Correo firmado y encriptado

2.4. Análisis sobre una clave privada robada

La infraestructura de PKI brinda un elemento esencial una situación de este tipo. Este se llama ***Certificate Revocation List* (*CRL*)**, es un repositorio que mantiene la *CA* (*Certificate Authority*, ente encargado de emitir los certificados) con los certificados que dejaron de ser válido (que fueron revocados, los certificados expirados no entran en esta lista).

2.4.1. Cómo actuar

Tanto si la clave privada fue robada como si fue robada y además eliminada de la máquina del dueño, lo primero que se debe hacer es **revocar los certificados** que corresponden a esa clave privada. Para ello se debe agregar el número de identificación del certificado a la *CRL* de la *CA* correspondiente.

Cuando un cliente pida el certificado del servidor cuya clave privada fue comprometida, lo rechazará y pedirá uno nuevo al verificar que ese certificado se encuentra en la *CRL*.

2.4.2. Consecuencias con la información encriptada

Como la clave privada es la que usa utiliza para desencriptar mensajes, estando comprometida un tercero podría desencriptar los mensajes que vayan hacia el dueño original de la clave.

2.4.3. Consecuencias con la información firmada

Para firmar se utiliza la clave privada, por lo tanto alguien más podría firmar documentos haciéndose pasar por el dueño de la clave.

3. *PGP*

Al igual que en la sección anterior, para las pruebas de firmar y encriptar correos electrónicos con PGP, se utilizaron las mismas cuentas

3.1. Crear claves *PGP* con gpg

Para crear las claves PGP utilizando la aplicación de línea de comando gpg, se debe ejecutar el siguiente comando:

```
1 gpg --gen-key
```

A continuación se detallan los datos que son requeridos por la aplicación para crear el par de claves

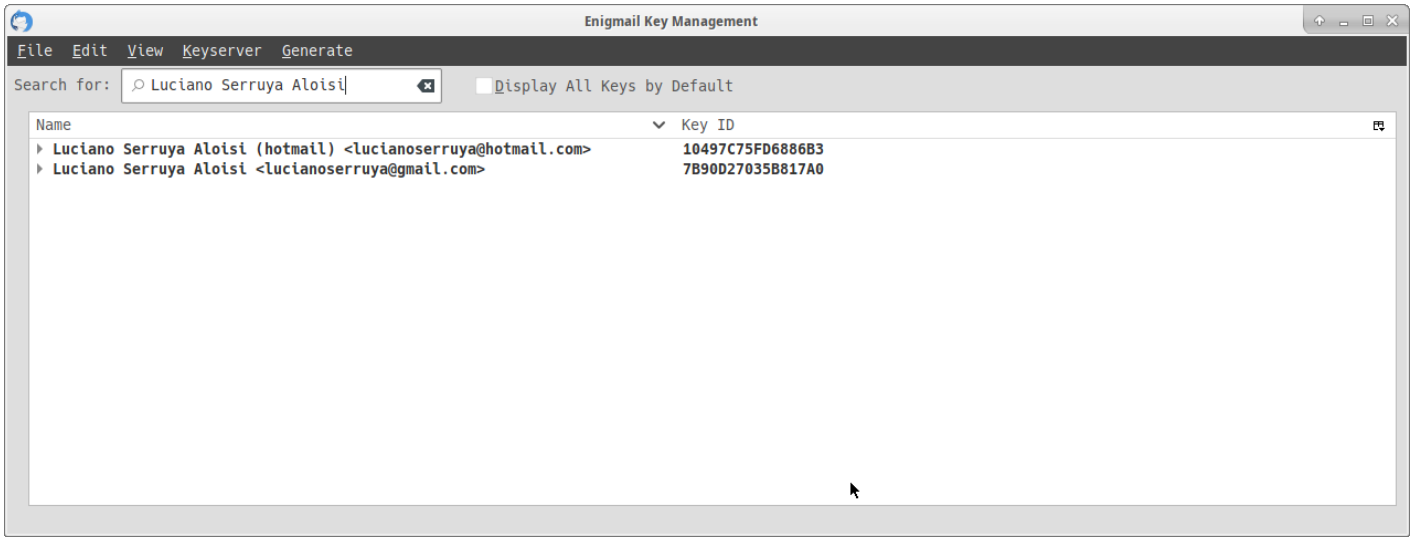
- Algoritmos de encriptación
- Tamaño (en bits) de la clave
- Tiempo máximo de validez de la clave
- Nombre real/nombre de usuario
- Dirección de correo electrónico
- Comentario

```
[luciano@arch-bangho:~/gitHub/ARyS/tp3/informe] master(+29/-0,1) bash ± gpg --list-keys "Luciano Serruya Aloisi"
pub   rsa2048 2018-11-08 [SC] [expires: 2019-11-08]
      5ADB663CBE6651EB85695A8E7B90D27035B817A0
uid           [ultimate] Luciano Serruya Aloisi <lucianoserruya@gmail.com>
sub   rsa2048 2018-11-08 [E] [expires: 2019-11-08]

pub   rsa2048 2018-11-08 [SC] [expires: 2019-11-08]
      ECF3F85B68047DF3B2EBF28310497C75FD6886B3
uid           [ultimate] Luciano Serruya Aloisi (hotmail) <lucianoserruya@hotmail.com>
sub   rsa2048 2018-11-08 [E] [expires: 2019-11-08]
```

Claves públicas PGP

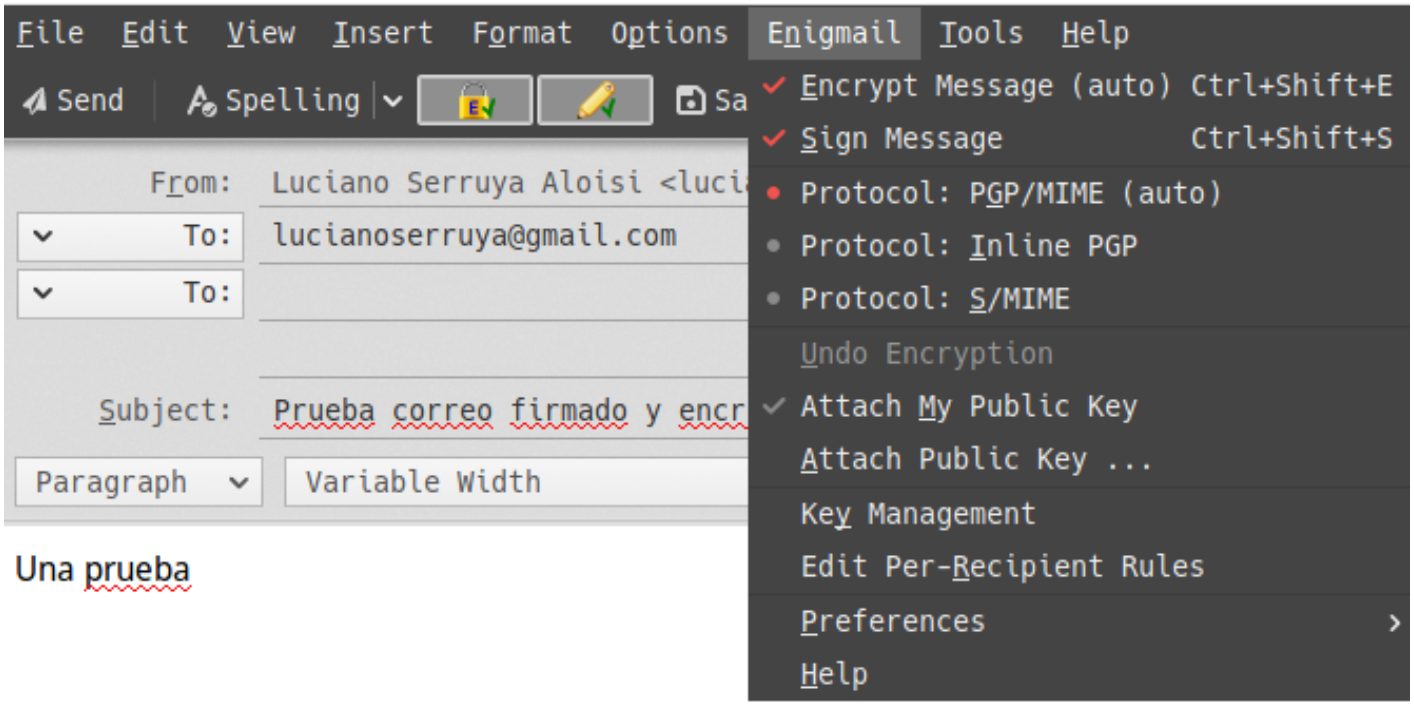
Una vez creado el par de claves, a través del complento de Thunderbird, *Enigmail*, se pueden cargar al cliente de correo para poder encriptar y firmar los correos con estas claves



Claves públicas cargadas en Thunderbird

3.1.1. Prueba de correo firmado y encriptado

A continuación se muestran capturas de un correo electrónico firmado y encriptado con PGP, enviado desde la cuenta *lucianoserruya@hotmail.com* a la cuenta *lucianoserruya@gmail.com*.

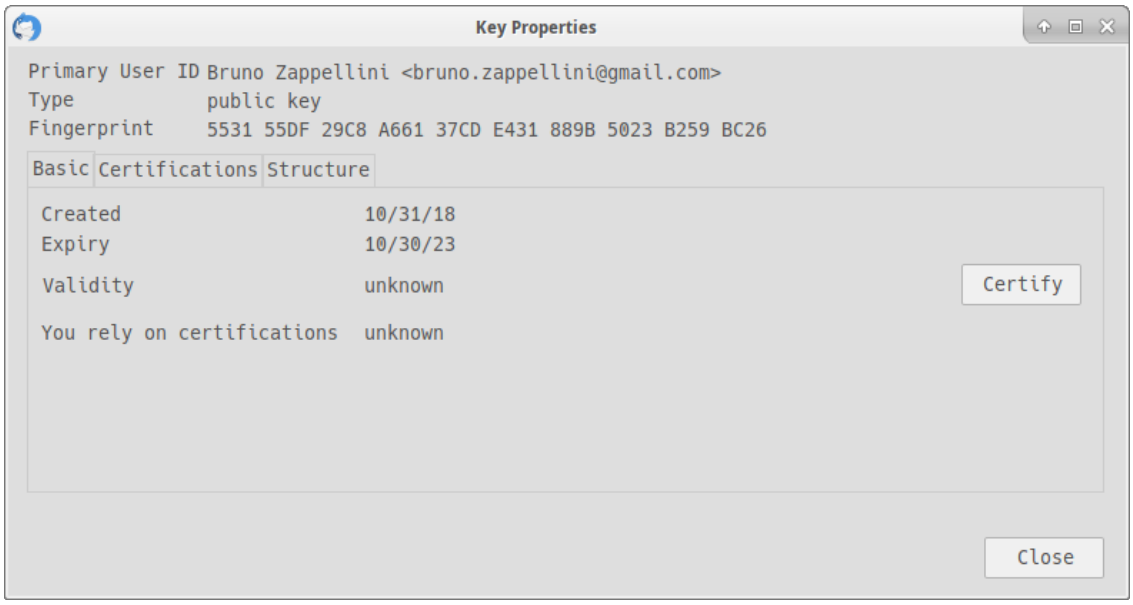


Enviado correo firmado y encriptado

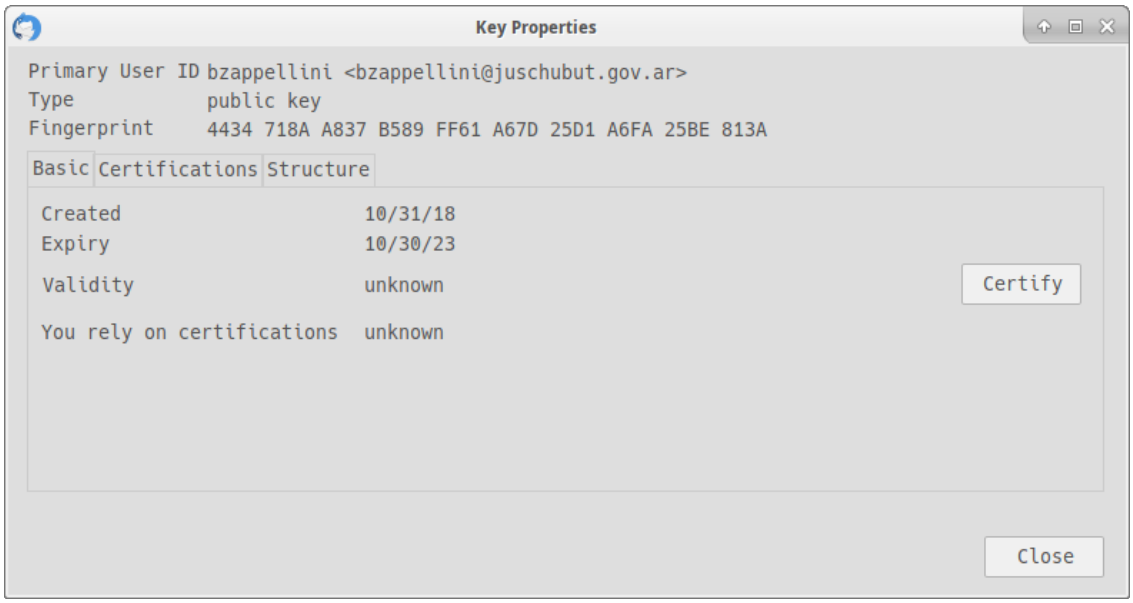
Correo recibido correctamente

3.2. Relaciones de confianza

Una vez importadas las claves de las cuentas *bruno.zappellini@gmail.com* y *bzappellini@juschubut.gov.ar*, el estado de validez de las claves es el siguiente:

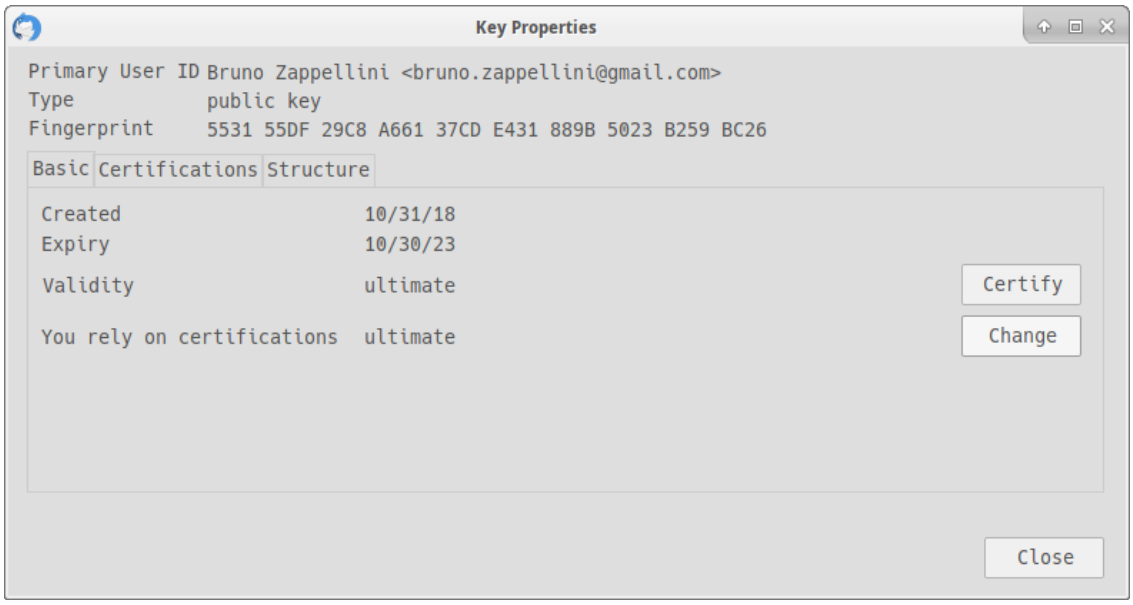


Clave para *bruno.zappellini@gmail.com*

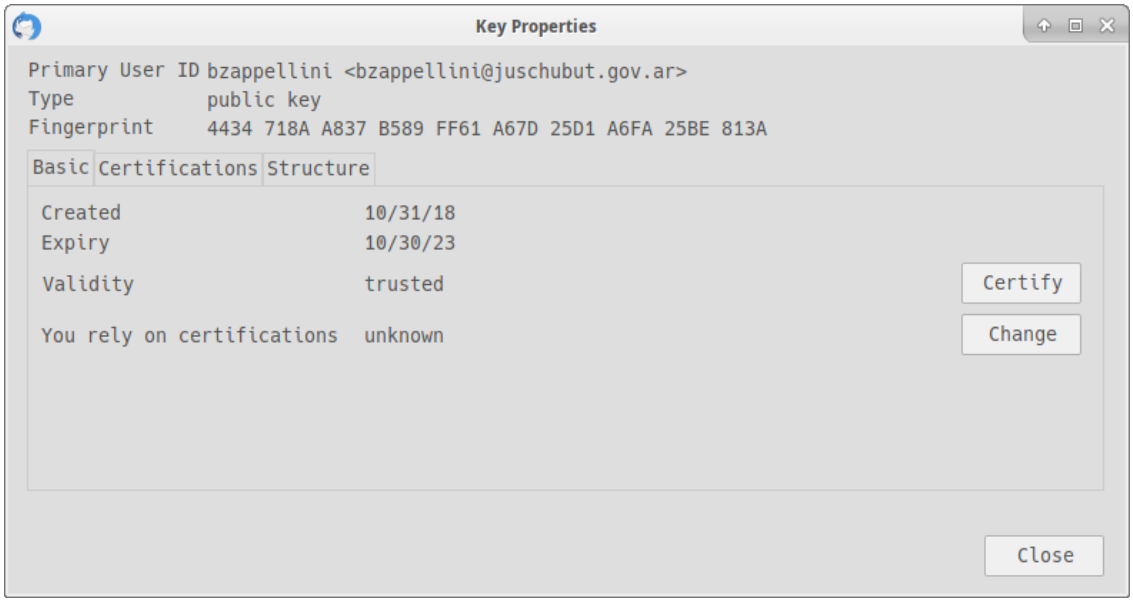


Clave para *bzappellini@juschubut.gov.ar*

Ahora bien, como la segunda clave está firmada por la primera, si se certifica que la primera es de quién dice ser (con la opción `Certify` en Thunderbird), la segunda automáticamente también estaría certificada. Este compartamiento se debe al ***modelo de confianza*** de PGP.



Clave para *bruno.zappellini@gmail.com* certificada - *se le concedieron los máximos niveles de confianza*



La clave para *bzappellini@juschubut.gov.ar* es validada automáticamente por el *modelo de confianza*

3.2.1. Cambiando confianza de las claves

Al recibir un correo de la cuenta *bruno.zappellini@gmail.com* teniendo la clave PGP sin verificar, el cliente Thunderbird indica que se pudo verificar la firma digital, pero muestra una salida distinta

Clave PGP sin verificar

Ahora bien, si verificamos que la clave es de su dueño, certificándolo con nuestra clave privada, Thunderbird muestra la firma de otra manera

Clave PGP verificada con nuestra clave privada

3.3. Mecanismo de firma y encriptación con PGP

PGP al ser un sistema de encriptación **asimétrico**, utiliza la **clave privada** para firmar los documentos, y la **clave pública del receptor** para encriptarlos.

3.4. Situaciones de confianza de una clave

Un cliente de correo (como Thunderbird, por ejemplo) puede confiar en una clave PGP (que no sea nuestra) por alguna de las siguientes razones:

- El usuario explícitamente decidió confiar en dicha clave (firmando la misma) - relación de confianza **directa**
- Se puede establecer una relación de confianza **indirecta**
    - El dueño de alguna clave de la cual confiamos plenamente (confianza *FULL*) firmó la clave
    - La clave fue firmada por *X* claves en las que confiamos marginalmente (confianza *MARGINAL*)

3.5. Análisis sobre una clave privada robada

PGP permite **revocar** claves a través de **certificados de revocación de claves**, ya sea porque se cambió de clave o porque la clave privada fue comprometida.

Cuando una clave es revocada, PGP debe contactar a los servidores en los cuales esté alojada la clave pública para actualizar su estado. De esta forma, cuando otros usuarios quieran actualizar la clave pública desde el servidor, estos se enterarán que está revocada.

Ahora bien, para generar el certificado de revocación se necesita tanto la **clave privada**, como su **contraseña**.

3.5.1. Cómo actuar

Si la clave privada fue comprometida **pero todavía está presente en la máquina del dueño**, se puede generar un certificado de revocación para revocarla.

Si la clave privada fue comprometida y eliminada de la máquina del dueño, **no hay nada que se pueda hacer**; se necesitan las dos partes (la clave privada y su contraseña) para generar el certificado de revocación.

3.5.2. Consecuencias con la información encriptada

Aquel que posea la clave privada podrá desencriptar la información que iba encriptada para el dueño original.

3.5.3. Consecuencias con la información firmada

El usuario que tenga la clave privada podrá firmar correos haciéndose pasar por el dueño original.

4. Criptografía simétrica y esteganografía

4.1. Encriptando archivos con `gpg`

La aplicación `gpg` permite generar pares de claves asimétricas PGP y también permite encriptar **simétricamente** un archivo; según las páginas `man`, el algoritmo por defecto para encriptación simétrica es **AES-128** (se puede cambiar utilizando el parámetro `-cipher-algo`).

Se puede crear un nuevo archivo y encriptarlo con `gpg` de la siguiente manera

```
1 echo "Un mensaje secreto" > secreto.txt
2 gpg -c secreto.txt
```

La aplicación levantará una ventana para ingresar una contraseña, y luego confirmarla; con dicha clave se encriptará el archivo de forma simétrica.

```
[luciano@arch-bangho:~] bash $ ls -l secreto.txt*
-rw-r--r-- 1 luciano luciano 19 Nov  9 20:14 secreto.txt
-rw-r--r-- 1 luciano luciano 95 Nov  9 20:16 secreto.txt.gpg
```

Archivo encriptado y no encriptado - el archivo encriptado es considerablemente mayor

Para desencriptarlo, sencillamente se debe ejecutar el comando

```
1 gpg -d secreto.txt.gpg > secreto_desencriptado.txt
```

Ingresar la contraseña con la que se encriptó el mensaje.