Cifrado asimétrico con GPG

Ismael Macareno Chouikh

2024-10-25

Índice

1. Instrucciones

Realizar un cifrado asimétrico con la aplicación `GPG`. Esta aplicación está basada en Linux y podremos realizar un cifrado simétrico como asimétrico.

Adjunto fichero PDF

Ejercicio:

- Investigar cuales son los parámetros más importantes de esta herramienta para el cifrado asimétrico (1 pto)
- Crea el par de clave pública y privada y borrarlas (pero al final de la práctica) (1 pto)
- Publicar en servidor público: `https://keyserver.ubuntu.com/` o `https://keys.openpgp.org/`
- Exportar la clave por 2 métodos (1 pto)
- Hacer copia de seguridad de la clave privada (1 pto)
- Crear certificado de revocación (1 pto)
- Crear un archivo de texto con un mensaje y realizar un cifrado asimétrico de dicho archivo (1 pto)
- Firma digital (1 pto)
- Abre el archivo con otro usuario o en otra máquina y descifrarlo (1 pto)
- Investiga que otras herramientas existen para realizar un cifrado asimétrico y realizar la misma prueba (1 pto)

2. Parámetros más importantes de `GPG`

- `-- version`: Este parámetro es bastante importante debido a que `gpg` tiene dos versiones
  - GnuPG 1.x
  - GnuPG 2.x
- `-- help`: Bastante importante para saber como funciona en caso de que no nos acordemos, etc.
- `--dump-options`: Imprime en pantalla una lista de las opciones disponibles
- `-s`: Firma un mensaje. Se puede combinar con las opciones `--encrypt` y a su vez con las opción `--symmetric` para encriptar un mensaje de manera simétrica.
- `-e`: Encripta datos para una o más de una clave pública. Está opción se puede combinar con la opción `-s`
- `-c`: Encripta de manera simétrica usando un parafraseado. Usa por defecto AES-128
- `-d` Desencripta el fichero que le proporcionemos
- `--list-public-keys`: Lista las claves públicas
- `--list-secret-keys`: Lista las claves privadas

Hay muchísimas otras opciones pero considero que estás son las más importantes.

3. Crear un par de claves

Para generar un par de claves lo único que hay que hacer es ejecutar el comando `gpg --gen-key`.

- Nos pedirá:
  - **tipo de clave**
  - **tamaño**
  - **validez**
  - **nuestro nombre**
  - **email**
  - **contraseña**
- **¡OJO!**, no parar de hacer algo, por ejemplo, mover el ratón

Real name: ismael Email address: ismael@ismael.com You selected this USER-ID: ïsmael <ismael@ismael.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? N Real name: ismael You selected this USER-ID: ïsmael <ismael@ismael.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? O We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy. We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy. gpg: revocation certificate stored as '*home/maka*.gnupg/openpgp-revocs.d/D30DEB9F5C0ABD842D58F3C2CDCAC2D071746359.rev' public and secret key created and signed.

pub ed25519 2024-10-18 [SC] [expires: 2027-10-18] D30DEB9F5C0ABD842D58F3C2CDCAC2D071746359 uid ismael <ismael@ismael.com>sub cv25519 2024-10-18 [E] [expires: 2027-10-18]

4. Publicarlas en un servidor público

Publico mis claves en https://keyserver.ubuntu.com/

Para publicar mi clave pública sigo los siguientes pasos:

1. Ejecutar el comando `gpg --list-public-keys` para saber cuáles son mis claves públicas

maka@makaSAD:~$ gpg –list-public-keys

home/maka.gnupg/pubring.kbx

---

pub rsa2048 2024-09-23 [SC] 3FD863EEAC3235E2266FDCDFE0BA8B5416AA7B9E uid [ultimate] Personales (Contraseña para archivos personales) sub rsa2048 2024-09-23 [E]

pub ed25519 2024-10-18 [SC] [expires: 2027-10-18] D30DEB9F5C0ABD842D58F3C2CDCAC2D071746359 uid [ultimate] ismael <ismael@ismael.com>sub cv25519 2024-10-18 [E] [expires: 2027-10-18]

2. Luego de saber cuantas claves públicas tengo y saber cuál es la que quiero lo que hago es ejecutar el comando `gpg --export --armor ismael` para que me genere la clave publica que quiero

—BEGIN PGP PUBLIC KEY BLOCK—

mDMEZxIhchYJKwYBBAHaRw8BAQdAV+j8RU8Z9hzk27ybMDGUDMObfTEys0u8UTS3

V1daM7i0GmlzbWFlbCA8aXNtYWVsQGlzbWFlbC5jb20+iJkEExYKAEEWIQTTDeuf

XAq9hC1Y88LNysLQcXRjWQUCZxIhcgIbAwUJBaOagAULCQgHAgIiAgYVCgkICwIE

FgIDAQIeBwIXgAAKCRDNysLQcXRjWfOFAQCUrdcrKjS+0sxtBjpAu5x3SKl2mfRE

G+LkIPWmlZdXJQD/XhxHG6yCHOU9TxrPaZ5/HQbdkygli1KbyW0ykrrBNQ64OARn

EiFyEgorBgEEAZdVAQUBAQdAEtx+NLd3yNnf+4YNGcMqgwQ6MwE4kE7Jl9w3i30S

U0oDAQgHiH4EGBYKACYWIQTTDeufXAq9hC1Y88LNysLQcXRjWQUCZxIhcgIbDAUJ

BaOagAAKCRDNysLQcXRjWVaEAP4ys1MIB2jX+XjTLV6rFtbAAkAeZcOS97yT7Oeg

mW6RPwD9EJDXR7tLqTQ16pk2jaXYGZaJzLsx6Opw2Ws22dYxvQk=

=m2Hw

—END PGP PUBLIC KEY BLOCK—

3. Copiar la clave pública creada correspondiente a ismael@ismael.com que es ismael
4. Acceder a la página web https://keyserver.ubuntu.com/ y pegar la clave copiada

¡OJO!, hay que copiar toda la salida del comando `gpg --export --armor ismael` incluido el bloque.

5. Exportar la clave mediante 2 métodos distintos

En este caso lo que voy a hacer es exportar la clave pública tanto por CLI como por GUI.

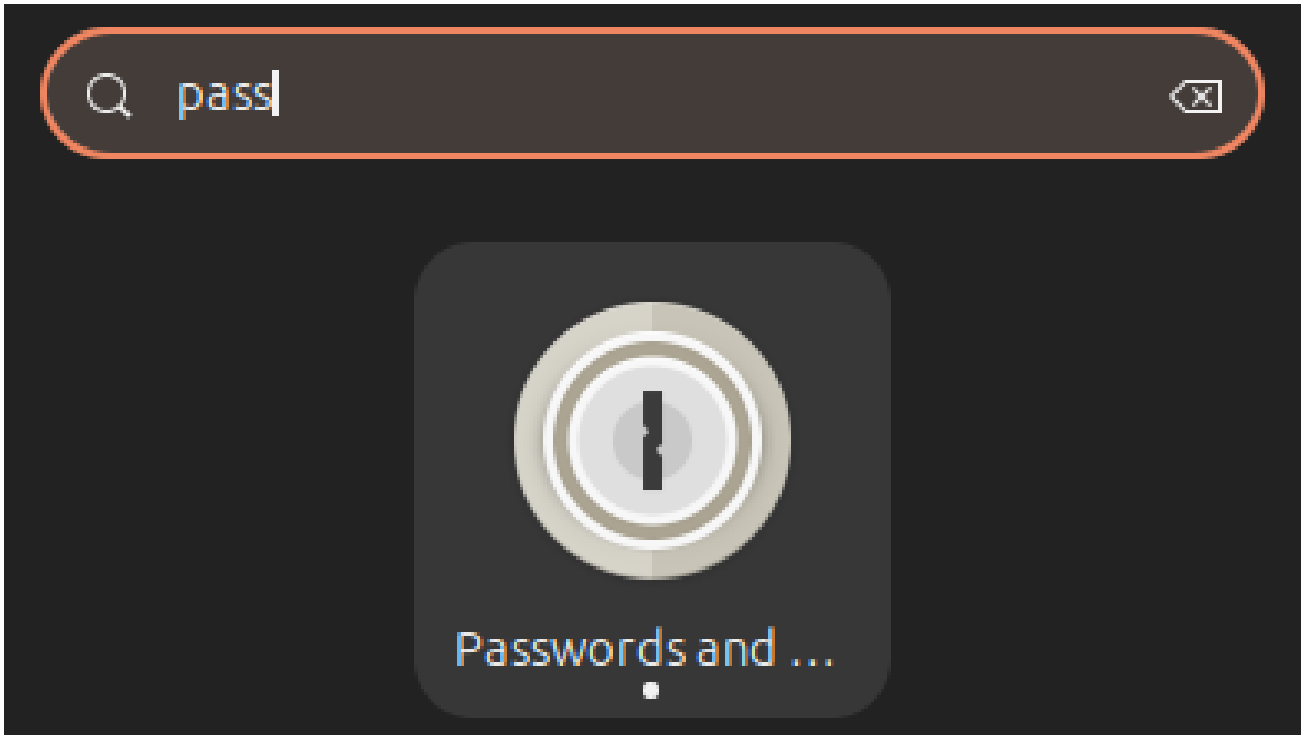
5.1. Mediante CLI

Para exportar la clave pública mediante CLI lo que único que hay que hacer es ejecutar el siguiente comando:

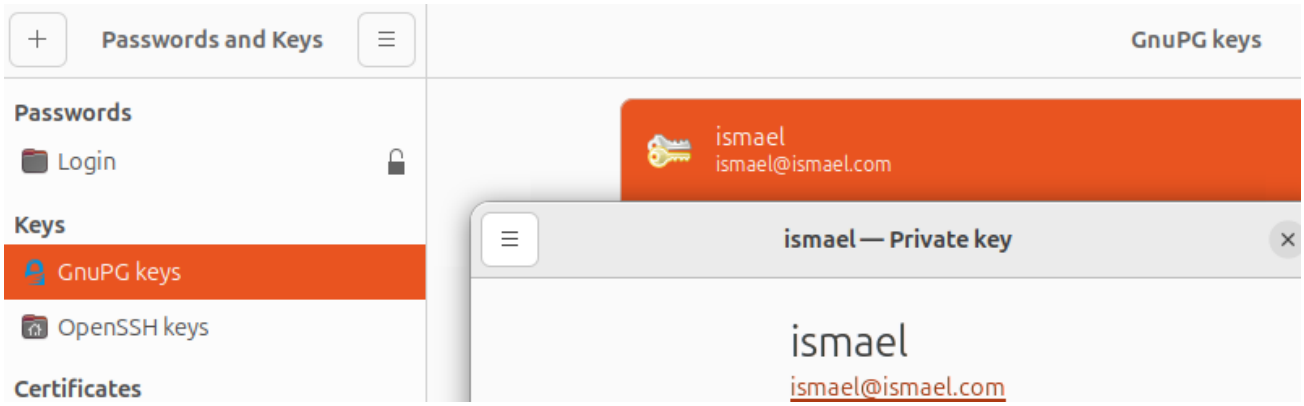
```
gpg --export --armor ismael >claveCli.asc
```

5.2. Mediante GUI

Para esto lo que haremos será buscar en el entorno gráfico de nuestro sistema (en mi caso un Ubuntu 24.04) la aplicación de contraseñas y claves

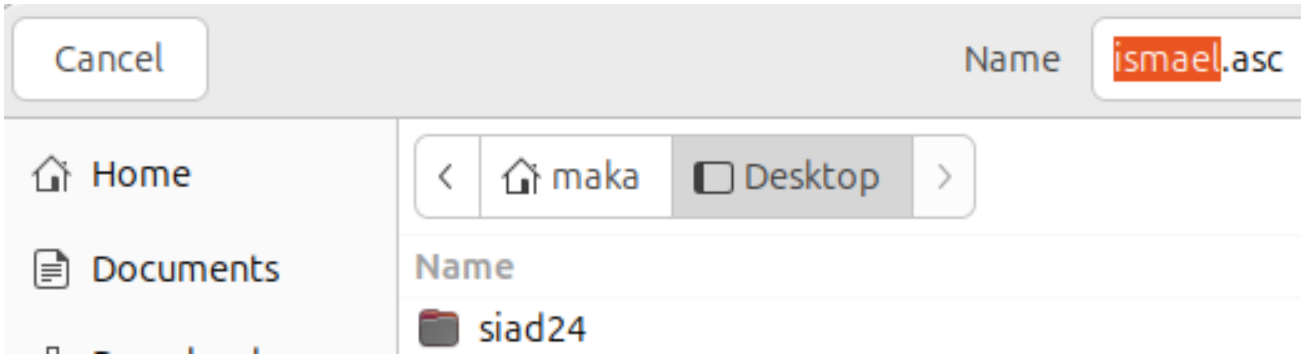


Luego de abrir la aplicación lo que tendremos que hacer será seleccionar la clave de la cuál queremos disponer de la clave pública y hacer clic en la misma.



Cuando se nos haya abierto la ventana que se puede apreciar en la imagen de arriba lo que tendremos que hacer será:

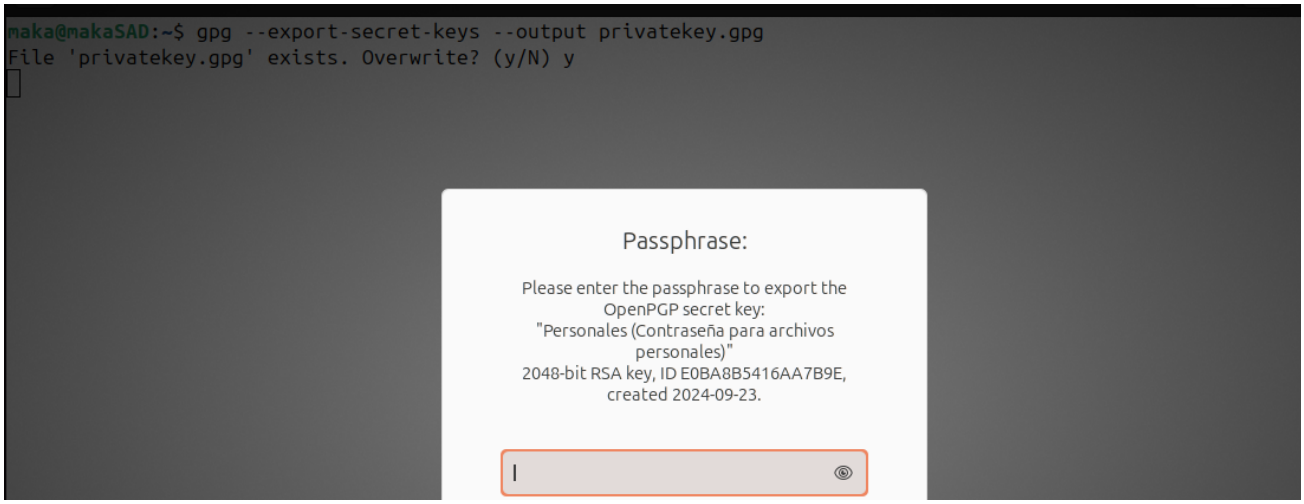
- hacer clic en el icono que aparece arriba a la izquierda
- exportar clave pública



6. Hacer una copia de seguridad de la clave privada

Para hacer la copia de seguridad de la clave privada lo único que hago es ejecutar el comando `gpg --export-secret-keys --output privateKey.gpg`

¡OJO!, se tiene que hacer mediante un entorno gráfico, si se hace mediante SSH no funciona.



7. Crear certificado de revocación

Un certificado de revocación GPG es un fichero que permite cancelar una llave pública GPG en caso de que la llave privada esté comprometida, perdida o sea inaccesible.

Para crear un certificado de revocación tendremos que realizar los siguientes pasos:

1. Identificar la llave pública mediante el comando `gpg -k`
2. Generar el certificado de revocación mediante el comando `gpg -a -o revoke.asc --gen-revoke IDCLAVEPUBLICA`
3. En el proceso de crear el certificado de revocación se nos pedirá que indiquemos el motivo

```
maka@makaSAD:~$ gpg -a -o revoke.asc --gen-revoke ismael

sec  ed25519/CDCAC2D071746359 2024-10-18 ismael <ismael@ismael.com>

Create a revocation certificate for this key? (y/N) y
Please select the reason for the revocation:
  0 = No reason specified
  1 = Key has been compromised
  2 = Key is superseded
  3 = Key is no longer used
  Q = Cancel
(Probably you want to select 1 here)
Your decision? 1
Enter an optional description; end it with an empty line:
> Para la practica de SAD
>
Reason for revocation: Key has been compromised
Para la practica de SAD
Is this okay? (y/N) y
Revocation certificate created.

Please move it to a medium which you can hide away; if Mallory gets
access to this certificate he can use it to make your key unusable.
It is smart to print this certificate and store it away, just in case
your media become unreadable.  But have some caution:  The print system of
your machine might store the data and make it available to others!
```

¡OJO!, se necesita un entorno gráfico debido a que nos va a aparecer una ventana solicitando la contraseña de la clave

8. Cifrado asimétrico de fichero con contenido

Para realizar este apartado tendremos que seguir los siguientes pasos:

1. Generar un nuevo par de claves para el usuario al que se supone que le daremos el fichero cifrado. En mi caso usare el usuario fary de mi MV Ubuntu 24.04 como destinatario y al usuario "makaçomo remitente. Para generar el par de claves usaremos el comando `gpg --gen-key`
2. Ejecutaremos el comando `gpg -e -u REMITENTEr "DESTINATARIO"NOMBREFICHERO`

```
Real name: fary
Email address: fary@fary.com
You selected this USER-ID:
    "fary <fary@fary.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: revocation certificate stored as '/home/maka/.gnupg/openpgp-revocs.d/25A33DD1CD6B7A622FBA9DBD0FE73B522C87A849.re
public and secret key created and signed.

pub   ed25519 2024-10-20 [SC] [expires: 2027-10-20]
      25A33DD1CD6B7A622FBA9DBD0FE73B522C87A849
uid                      fary <fary@fary.com>
sub   cv25519 2024-10-20 [E] [expires: 2027-10-20]

maka@makaSAD:~$ gpg -e -u "ismael" -r "fary" ficheroAsimetrico.txt
```

¡OJO!, se tiene que realizar mediante el entorno gráfico no por conexión SSH porque aparecerá una ventana para introducir la contraseña a la hora de generar el par de claves.

9. Firma digital

Para este apartado tendremos que completar los siguientes pasos:

1. Generar un par de claves (en mi caso uso las que ya tengo)
2. Ejecutar el comando `gpg --detach-sign NOMBREARCHIVO`
3. Verificar la firma mediante el comando `gpg --verify NOMBREARCHIVO.sig`

```
maka@makaSAD:~$ gpg --detach-sign ficheroAsimetrico.txt
maka@makaSAD:~$ gpg --ver
--verbose         --verify          --verify-files    --verify-options  --version
maka@makaSAD:~$ gpg --verify ficheroAsimetrico.txt.
ficheroAsimetrico.txt.gpg  ficheroAsimetrico.txt.sig
maka@makaSAD:~$ gpg --verify ficheroAsimetrico.txt.sig
gpg: assuming signed data in 'ficheroAsimetrico.txt'
gpg: Signature made Sun 20 Oct 2024 08:28:25 PM CEST
gpg:                using RSA key 3FD863EEAC3235E2266FDCDFE0BA8B5416AA7B9E
gpg: Good signature from "Personales (Contraseña para archivos personales)" [ultimate]
```

10. Abrir fichero cifrado asimétrico con otro usuario y descifrarlo

Para realizar este apartado lo que hare será usar al otro usuario que tengo creado en mi MV Ubuntu 24.04 llamado `fary` ya que en el apartado 8 cuando cifre el fichero puse como destinatario `fary`.

1. Acceder al usuario `maka` y enviar el fichero al usuario `fary`, en este caso yo no voy a enviar el fichero, se lo pasaré al usuario `fary` através de `cp` y modificare el usuario y grupo
   - `cp ficheroAsimetrico.txt.gpg /home/fary/Desktop`
   - `sudo chown fary:fary /home/fary/Desktop/ficheroAsimetrico.txt.gpg`
2. Exportar las claves públicas y privadas del usuario que cifro el fichero y pasársela al usuario que va a descifrar el fichero, en mi caso la clave pública del usuario `maka` aunque las voy a exportar todas para no complicarme
   - `gpg --export-secret-keys --output secretitos.gpg`
   - `cp secretitos.gpg /home/fary/Desktop/`
   - `sudo chown fary:fary /home/fary/Desktop/*`
3. Acceder de manera gráfica al usuario `fary` para descifrar el fichero
   - `gpg --import secretitos.gpg`
   - `gpg --decrypt -o aver.txt ficheroAsimetrico.txt.gpg`

```
fary@makaSAD:~/Desktop$ gpg --decrypt -o aver.txt ficheroAsimetrico.txt.gpg
gpg: encrypted with cv25519 key, ID B6A130121A9C62B8, created 2024-10-20
      "fary <fary@fary.com>"
fary@makaSAD:~/Desktop$ cat aver.txt
para la practica de sad
```

¡OJO!, todo se tiene que realizar cambiando de usuario gráficamente.

11. Otras herramientas existentes

11.1. OPENSSL

En mi caso solo voy a probar esta herrmienta CLI.

`OPENSSL` Consiste en un robusto paquete de herramientas de administración y bibliotecas relacionadas con la criptografía, que suministran funciones criptográficas a otros paquetes como OpenSSH y navegadores web (para acceso seguro a sitios HTTPS).

11.1.1. Generación de claves públicas y privadas

1. Generar la clave privada
   - `openssl genrsa -out privada.pem 2048`
2. Generar la clave pública a partir de la privada
   - `openssl rsa -pubout -in privada.pem -out publica.pem`

```
fary@makaSAD:~$ openssl genrsa -out privada.pem 2048
fary@makaSAD:~$ open
open             opensnoop.bt     openvpn
opensnoop-bpfcc  openssl          openvt
fary@makaSAD:~$ openssl rsa -pubout -in privada.pem -out publica.pem
writing RSA key
fary@makaSAD:~$ ls -la | grep prm
fary@makaSAD:~$ ls -la | grep pem
-rw-------  1 fary fary 1708 Oct 25 01:51 privada.pem
-rw-rw-r--  1 fary fary  451 Oct 25 01:51 publica.pem
```

11.1.2. Cifrado de Fichero

1. Crear un fichero con su respectivo contenido
   - `echo "Probando openssl para asimetrico» open.txt`
2. Cifrar el fichero
   - `openssl enc -aes-256-cbc -in open.txt -out opencif.cipher -pass pass:clave`

```
fary@makaSAD:~$ echo "Probando openssl para asimetrico" > open.txt
fary@makaSAD:~$ openssl enc -aes-256-cbc -in open.txt -out opencif.cipher -pass pass:clave
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

¡AVISO!, aunque nos de ese pequeño error que se puede apreciar en la imagen de arriba el fichero `.cipher` se genera igualmente.

11.1.3. Descifrar el fichero con otro usuario

1. Pasar la clave privada y el fichero del usuario al otro
2. Modificar el dueño del ambos ficheros con `chown`
3. Ejecutar el siguiente comando

- `openssl enc -aes-256-cbc -d -pass file:privada.pem -in opencif.cipher cifrado -out resultado`

12. Valoración Personal

Práctica demasiado larga a mi parecer, hay partes que son más sencillas que otras cosa que es bastante normal pero por ejemplo me sobra el último punto de investigar otra herramienta.

Por lo demás, al menos me he entretenido un tiempo para hacerla, algo es algo....