

UT02: – Técnicas De Cifrado.

Nombre: Ricardo Camacho Diaz-Roncero y Miguel Lillo Romero

Curso: 2º ASIR.

Índice

a) CIFRADO: Historia	2
- Realizar CISCO CCNA Security 2.0. Laboratorio: Explorando métodos de cifrado.....	2
b) CIFRADO: Funciones HASH--> Integridad y Autenticidad.....	3
- 1. Utiliza un programa en Windows o GNU/Linux para simular la integridad, utilizando MD5 y SHA1	3
2. Utiliza un programa en Windows o GNU/Linux para simular la autenticidad utilizando HMAC-MD5, HMAC-SHA1	6
c) CIFRADO: Simétrico y Asimétrico--> Confidencialidad, Integridad y Autenticidad.....	8
- 1. Utiliza un programa en Windows o GNU/Linux para simular la confidencialidad mediante “cifrado simétrico”	8
- 2. Utiliza un programa en Windows o GNU/Linux para simular la confidencialidad mediante “cifrado asimétrico”.....	10
- 3. Utiliza un programa en Windows o GNU/Linux para simular la autenticidad mediante “cifrado asimétrico”	17
4. Utiliza un programa en Windows o GNU/Linux para simular la “autenticidad”+“confidencialidad” mediante “cifrado asimétrico”	19
5. Utiliza un programa en Windows o GNU/Linux para simular una comunicación segura utilizando cifrados híbridos “autenticidad” + “confidencialidad”+“integridad”: asimétricos (clave pública) y simétricos (clave privada).....	21
Conclusión	25

Introducción

En esta práctica aprenderemos sobre el cifrado, para ello realizaremos técnicas de cifrado y veremos la historia que tienen.

a) CIFRADO: Historia.

- Realizar CISCO CCNA Security 2.0. Laboratorio: Explorando métodos de cifrado.

El cifrado Vigenère es un cifrado basado en diferentes series de caracteres o letras del cifrado César formando estos caracteres una tabla, llamada tabla de Vigenère, que se usa como clave. El cifrado de Vigenère es un cifrado de sustitución simple poli alfabético. El cifrado Vigenère se ha reinventado muchas veces.

Encriptación

Cipher Keyword	F	R	A	N	V	I	G	S	A	D
Encrypted Message	R	O	U	T	E	R	R	O	U	T
Decrypted Message	W	F	U	G	Z	Z	X	G	U	W

Desencriptación

Cipher Keyword	W	F	U	G	Z	Z	X	G	U	W
Encrypted Message	R	O	U	T	E	R	R	O	U	T
Decrypted Message	F	R	A	N	V	I	G	S	A	D

Página que recomienda Cisco para probar esto: [Vignere Cipher](#)

This page is for amusement only. Instructions are given below this form.

Input:

clear

Hola somos Miguel y Ricardo probando la practica.

Key:

clear

AWKOPU

Coding direction:

encode

decode

Output:

clear

Hkvo himkc Axauav m Gccwbrd jrklocxo hk dgucpsqp.

This page is for amusement only. Instructions are given below this form.

Input:

clear

Hkvo himkc Axauav m Gccwbrd jrklocxo hk dgucpsqp.

Key:

clear

AWKOPU

Coding direction:

encode

decode

Output:

clear

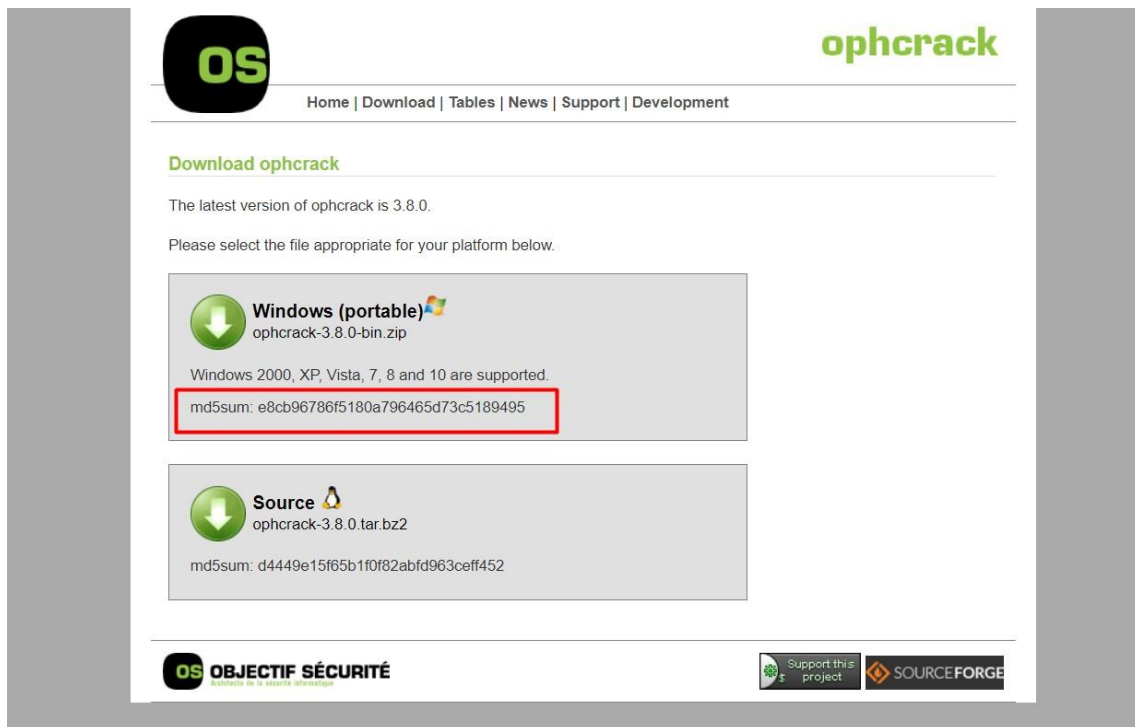
Hola somos Miguel y Ricardo probando la practica.

b) CIFRADO: Funciones HASH--> Integridad y Autenticidad.

- 1. Utiliza un programa en Windows o GNU/Linux para simular la integridad, utilizando MD5 y SHA1.

Para realizar esto usare PowerShell. PowerShell es una interfaz de consola (CLI) con posibilidad de escritura y unión de comandos por medio de instrucciones (script)). Esta interfaz de consola está diseñada para su uso por parte de administradores de sistemas, con el propósito de automatizar tareas o realizarlas de forma más controlada.

En mi caso lo usare comprobare la integridad de la versión portable de Ophcrack. Podemos ver el md5 desde su misma página web.



Una vez guardado el programa simplemente abrimos PowerShell y ejecutamos el siguiente comando para sacar el hash del archivo:

CertUtil -hashfile "C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\ophcrack-3.8.0-bin.zip" MD5

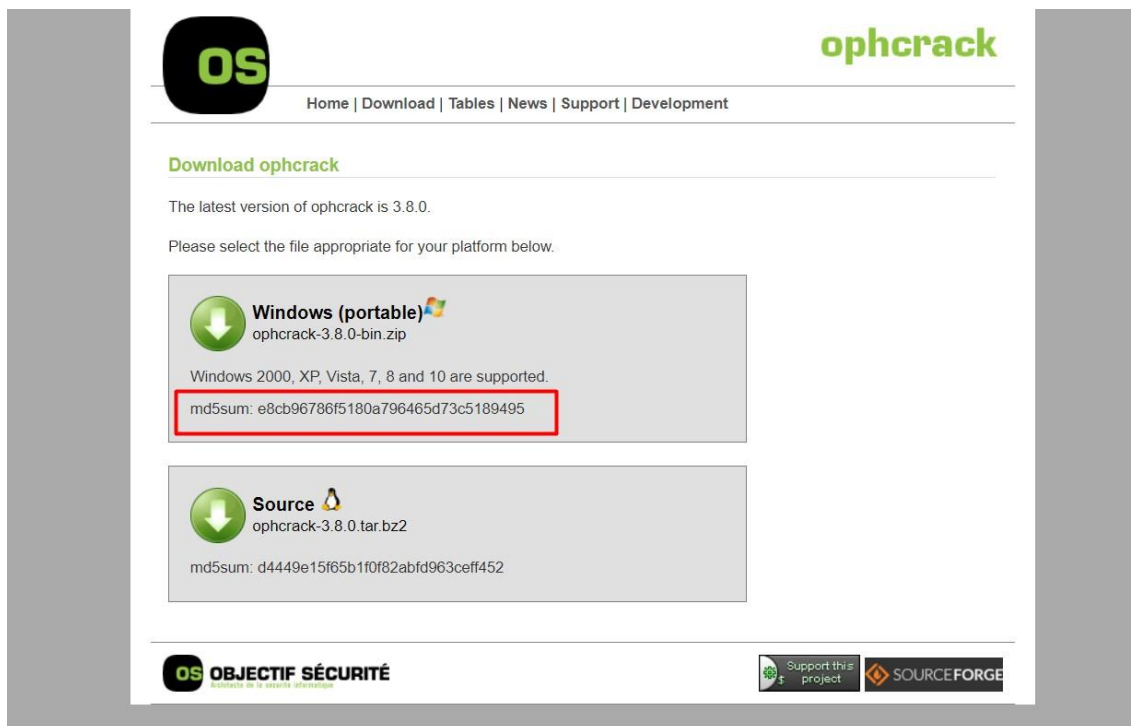
```
Windows PowerShell
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02> CertUtil -hashfile "C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\ophcrack-3.8.0-bin.zip" MD5
MD5 hash de c:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\ophcrack-3.8.0-bin.zip:
e8cb96786f5180a796465d73c5189495
CertUtil: -hashfile comando completado correctamente.
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02>
```

Vemos como nos genera un hash.

Podremos hacer lo mismo en SHA1.

```
Windows PowerShell
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02> certutil -hashfile "C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\ophcrack-3.8.0-bin.zip" SHA1
SHA1 hash de C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\ophcrack-3.8.0-bin.zip:
F7975e274bc4344080cab23408f78e9ad659e857
certutil: -hashfile comando completado correctamente.
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02>
```

Ahora simplemente tenemos que comprar los dos hashes (el de PowerShell con el que ofrece la página Ophcrack. Vemos que son iguales por lo cual la integridad es correcta.



```
Windows PowerShell
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02> certutil -hashfile "C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\ophcrack-3.8.0-bin.zip" MD5
MD5 hash de C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\ophcrack-3.8.0-bin.zip:
e8cb96786f5180a796465d73c5189495
certutil: -hashfile comando completado correctamente.
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02>
```

Para probar SHA1 al no encontrar programas que brinden en su página web el SHA1 para comprobar la integridad lo que haré es crear un archivo txt con algo escrito, generar su SHA1 y después modificarlo para que se vea cómo cambia.

Crearemos un documento de texto y generamos su SHA1.

TECNICAS DE CIFRADO

File Explorer window showing the folder structure: Este equipo > Escritorio > 2fp > Seguridad Informatica > RA02.

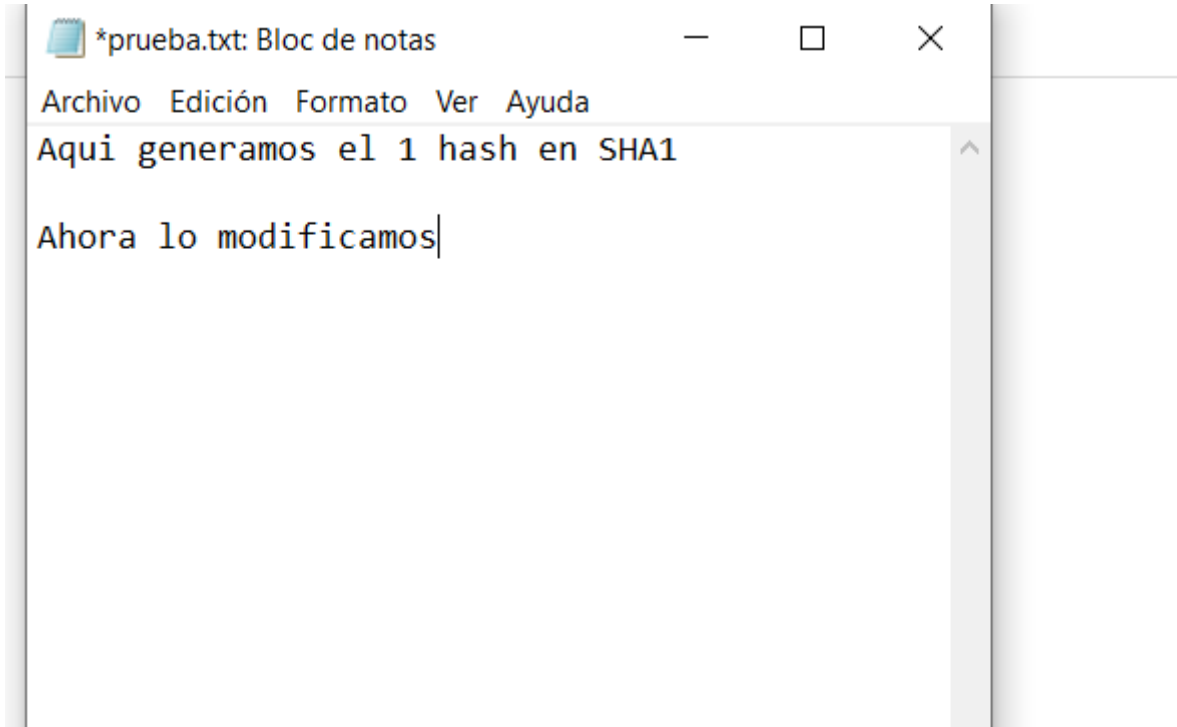
Nombre	Fecha de modificación	Tipo	Tamaño
Practicas	27/10/2023 11:56	Carpeta de archivos	
Teoria	20/10/2023 13:15	Carpeta de archivos	
ophcrack-3.8.0-bin.zip	27/10/2023 11:40	Archivo WinRAR ZIP	15.469 KB
prueba.txt.txt	27/10/2023 12:32	Documento de tex...	0 KB

Notepad window titled "prueba.txt.txt: Bloc de notas" containing the text: "Aqui generamos el 1 hash en SHA1".

Windows PowerShell window showing the command execution:

```
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02> Certutil -hashfile "C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\prueba.txt" SHA1
SHA1 hash de C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\prueba.txt:
203e9e8cecb9855a19046026eb440b2440f772d
Certutil: -hashfile comando completado correctamente.
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02>
```

Si editamos el contenido vemos cómo cambia el SHA1.



```
Windows PowerShell
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02> CertUtil -hashfile "C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\prueba.txt" SHA1
SHA1 hash de C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\prueba.txt:
03e9e8cecbe9855a19046026eb440b2440f772d
CertUtil: -hashfile comando completado correctamente.
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02> CertUtil -hashfile "C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\prueba.txt" SHA1
SHA1 hash de C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02\prueba.txt:
fcb9d3919cd0a00a7a6ab0b59f4920e675baf33
CertUtil: -hashfile comando completado correctamente.
PS C:\Users\richa\Desktop\2fp\Seguridad Informatica\RA02>
```

2. Utiliza un programa en Windows o GNU/Linux para simular la autenticidad utilizando HMAC-MD5, HMAC-SHA1.

En términos criptográficos, es un tipo de código para autenticar mensajes en clave-hash (HMAC), es decir, es un diseño específico para generar un código de autenticación de mensaje (MAC) lo cual implica una función hash criptográfica junto con una llave criptográfica secreta.

También puede ser usada para verificar simultáneamente la autenticación de un mensaje y la integridad de los datos. El poder criptográfico del HMAC va a depender de la cantidad de caracteres de la clave secreta que se utilice.

Utilizaré dos páginas web conocidas para realizar esta tarea. La primera es la siguiente [\(Link\)](#). Introduciremos un mensaje con su clave y generaremos el HMAC-MD5.



This tool will allow you to test a variety of HMAC Hashing algorithms for a variety of purposes.

The String to be Encrypted

Hola somos Miguel y Ricardo

Secret Key

inves

Select Hash Algorithm to Use

md5

Generate

Clear



This tool will allow you to test a variety of HMAC Hashing algorithms for a variety of purposes.

The String to be Encrypted

Hola somos Miguel y Ricardo

Secret Key

inves

Select Hash Algorithm to Use

md5

Generate

Clear

Computed HMAC Result

9f8448ec80b34f483743d030ffa5baf6

Comprobaremos con otra página que el resultado es realmente correcto. [\(Link\)](#)

HMAC Generator

Select Algorithm

MD5

inves

Hola somos Miguel y Ricardo

 Generate HMAC

9f8448ec80b34f483743d030ffa5baf6

Podemos ver que los dos hmac-hashes son iguales por lo cual la autenticidad es correcta. Lo siguiente es hacer lo mismo, pero en HMAC-SHA1, lo realizaremos en las páginas web de antes.

The String to be Encrypted

Esto es una prueba con HMAC-SHA1

Secret Key

inves

Select Hash Algorithm to Use

sha1

Generate

Clear

Computed HMAC Result

01081a259f72cee565d2efa8bb6f4c51dd8864c9

Lo compararemos con la otra página web. Vemos que es igual que la anterior página por lo cual la autenticidad es correcta.

HMAC Generator

Select Algorithm

SHA1

inves

Esto es una prueba con HMAC-SHA1

 Generate HMAC



01081a259f72cee565d2efa8bb6f4c51dd8864c9

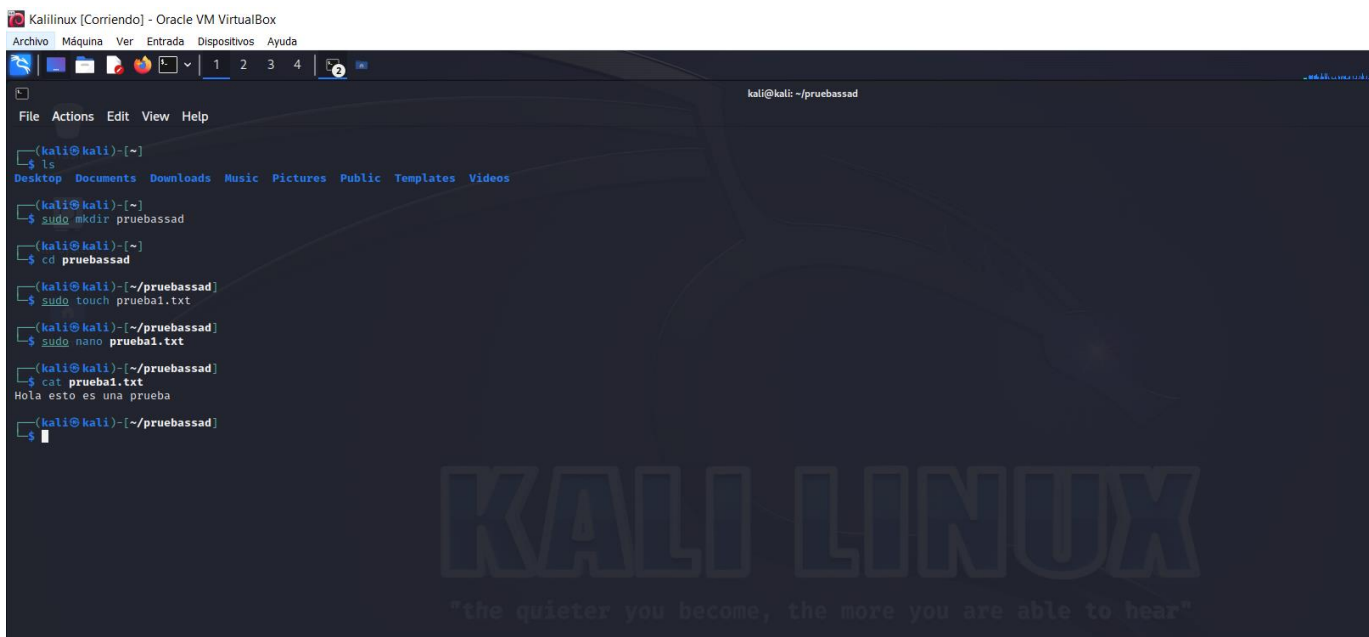
c) CIFRADO: Simétrico y Asimétrico--> Confidencialidad, Integridad y Autenticidad.

- 1. Utiliza un programa en Windows o GNU/Linux para simular la confidencialidad mediante “cifrado simétrico”.

La criptografía de clave simétrica, también llamada criptografía de clave secreta o criptografía de una clave, es un método criptográfico en el cual se usa una misma clave para cifrar y descifrar mensajes en el emisor y el receptor.

Utilizaremos para llevar acabo esta práctica Linux con Kalilinux El programa que usare es pgp.

Primero crearemos un documento y lo cifraremos con una clave simétrica.



```
Kalilinux [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

kali@kali: ~/pruebassad

File Actions Edit View Help

kali@kali:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos

kali@kali:~$ sudo mkdir pruebassad

kali@kali:~$ cd pruebassad

kali@kali:~/pruebassad$ sudo touch prueba1.txt

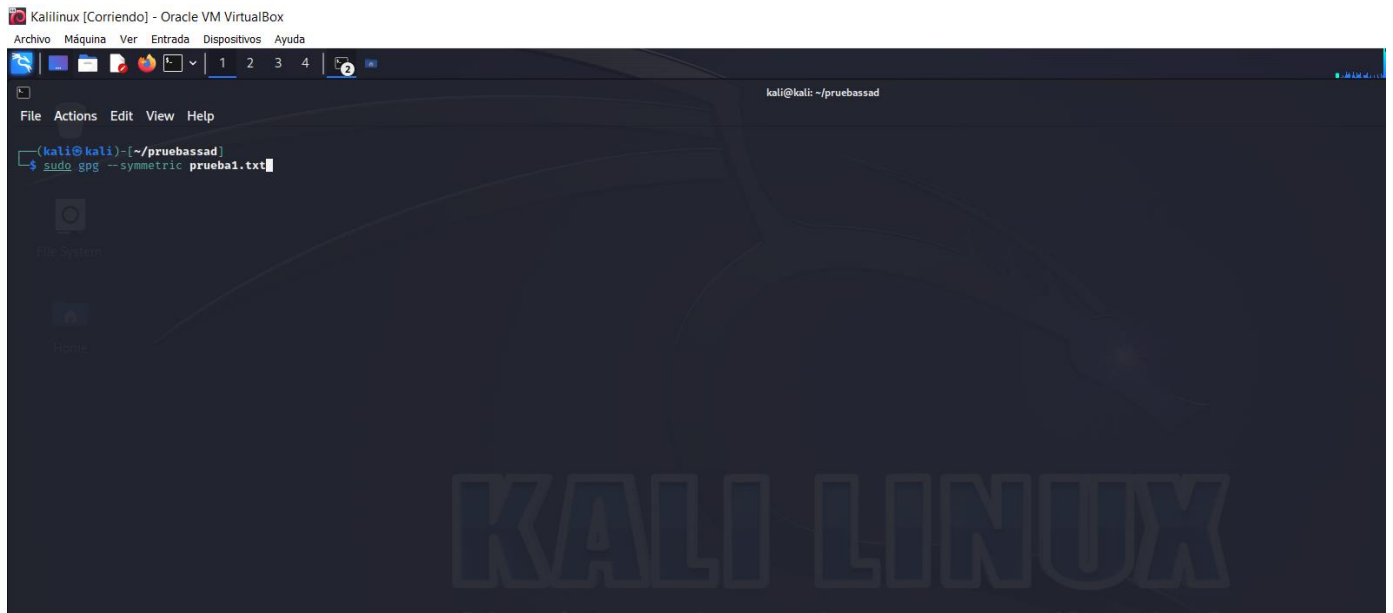
kali@kali:~/pruebassad$ sudo nano prueba1.txt

kali@kali:~/pruebassad$ cat prueba1.txt
Hola esto es una prueba

kali@kali:~/pruebassad$
```

KALI LINUX
"the quieter you become, the more you are able to hear"

Ahora lo cifraremos de manera simétrica con gpg.



Ponemos una clave.

Vemos como nos generó el mismo archivo, pero con extensión. gpg

```

kali@kali: ~/pruebassad
File Actions Edit View Help

(kali@kali)-[~/pruebassad]
$ sudo gpg --symmetric prueba1.txt

(kali@kali)-[~/pruebassad]
$ ls
prueba1.txt  prueba1.txt.gpg

(kali@kali)-[~/pruebassad]
$

```

Ahora simplemente enviaremos el archivo cifrado con clave simétrica al /home del segundo usuario que tenemos.

```

kali@kali: ~/pruebassad
File Actions Edit View Help

(kali@kali)-[~/pruebassad]
$ sudo adduser ricardo2
info: Adding user `ricardo2' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `ricardo2' (1001) ...
info: Adding new user `ricardo2' (1001) with group `ricardo2 (1001)' ...
info: Creating home directory `/home/ricardo2' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ricardo2
Enter the new value, or press ENTER for the default
  Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
info: Adding new user `ricardo2' to supplemental / extra groups `users' ...
info: Adding user `ricardo2' to group `users' ...

(kali@kali)-[~/pruebassad]
$ ls
prueba1.txt  prueba1.txt.gpg

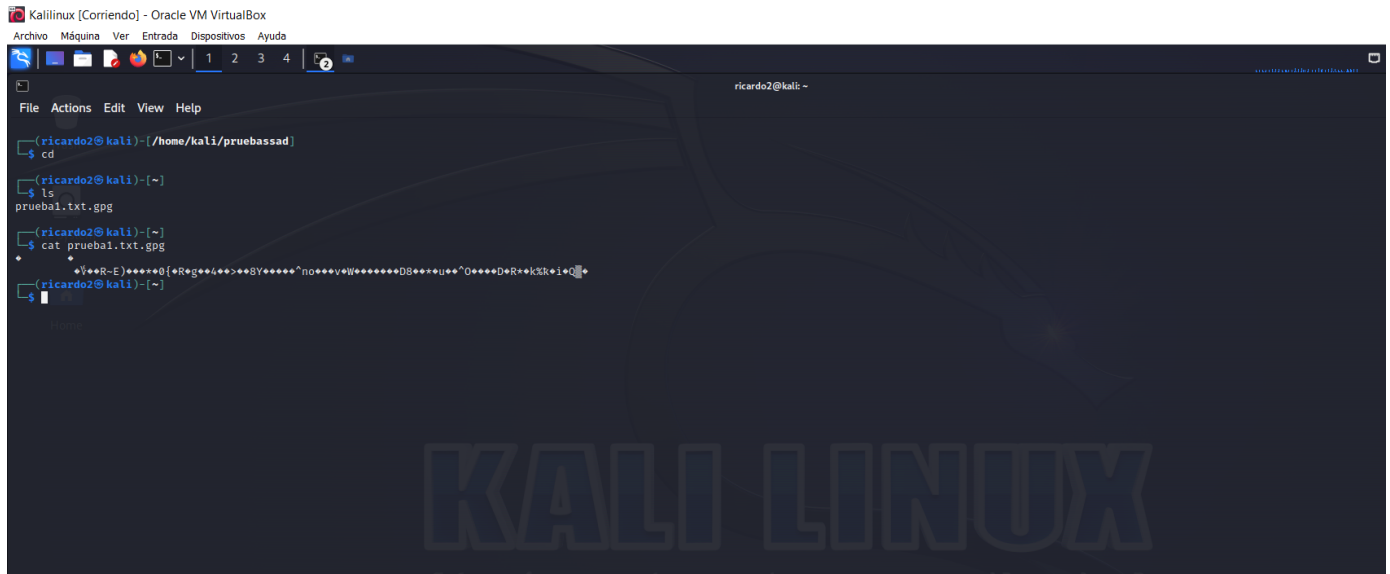
(kali@kali)-[~/pruebassad]
$ cp prueba1.txt.gpg /home/ricardo2
cp: cannot stat '/home/ricardo2/prueba1.txt.gpg': Permission denied

(kali@kali)-[~/pruebassad]
$ sudo cp prueba1.txt.gpg /home/ricardo2

(kali@kali)-[~/pruebassad]
$

```

Iniciaremos sesión con el segundo usuario e intentaremos ver el contenido con un simple cat. Vemos como lógicamente no deja al estar cifrado.



The screenshot shows a Kali Linux terminal window titled 'Kali Linux [Corriendo] - Oracle VM VirtualBox'. The terminal displays the following commands and output:

```
ricardo2@kali: ~/home/kali/pruebassad
$ cd
$ ls
prueba1.txt.gpg
$ cat prueba1.txt.gpg
+Y+R-E)+++++@{+R+g++/+++>+*+SY+++++^no+++v+W+++++DB+++++u++^O++++D+R+*k3k+i*Q
```

The background of the terminal window features a large, faint 'KALI LINUX' logo.

Simplemente lo descifraremos con gpg, nos pedirá la contraseña que pusimos al cifrar el archivo.

```
$ sudo gpg --decrypt prueba1.txt.gpg
```

Vemos como nos descifra el archivo y podemos ver su contenido.



The screenshot shows a Kali Linux terminal window with the following content:

```
Kalilinux [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

(ricardo2@kali)-[/home/kali/pruebassad]
$ sudo gpg --decrypt pruebal.txt.gpg
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
Hola esto es una prueba

(ricardo2@kali)-[/home/kali/pruebassad]
$
```

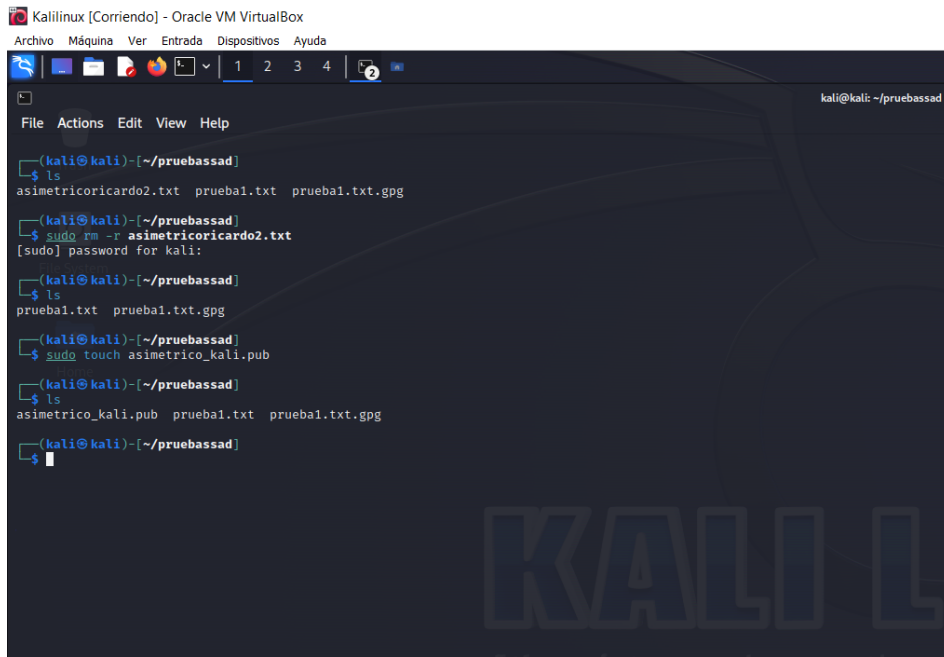
The terminal background features the Kali Linux logo and the slogan "the quieter you become, the more you are able to hear".

- 2. Utiliza un programa en Windows o GNU/Linux para simular la confidencialidad mediante “cifrado asimétrico”.

La criptografía asimétrica, también llamada criptografía de clave pública o criptografía de dos claves, es el método criptográfico que usa un par de claves para el envío de mensajes. Las dos claves pertenecen a la misma persona que recibirá el mensaje.

Para realizar esta práctica, igual que antes, utilizaremos gpg en Kalilinux.

Crearemos para ello un nuevo archivo con la cuenta ricardo2.



```
KaliLinux [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

kali@kali: ~/pruebassad

File  Actions  Edit  View  Help

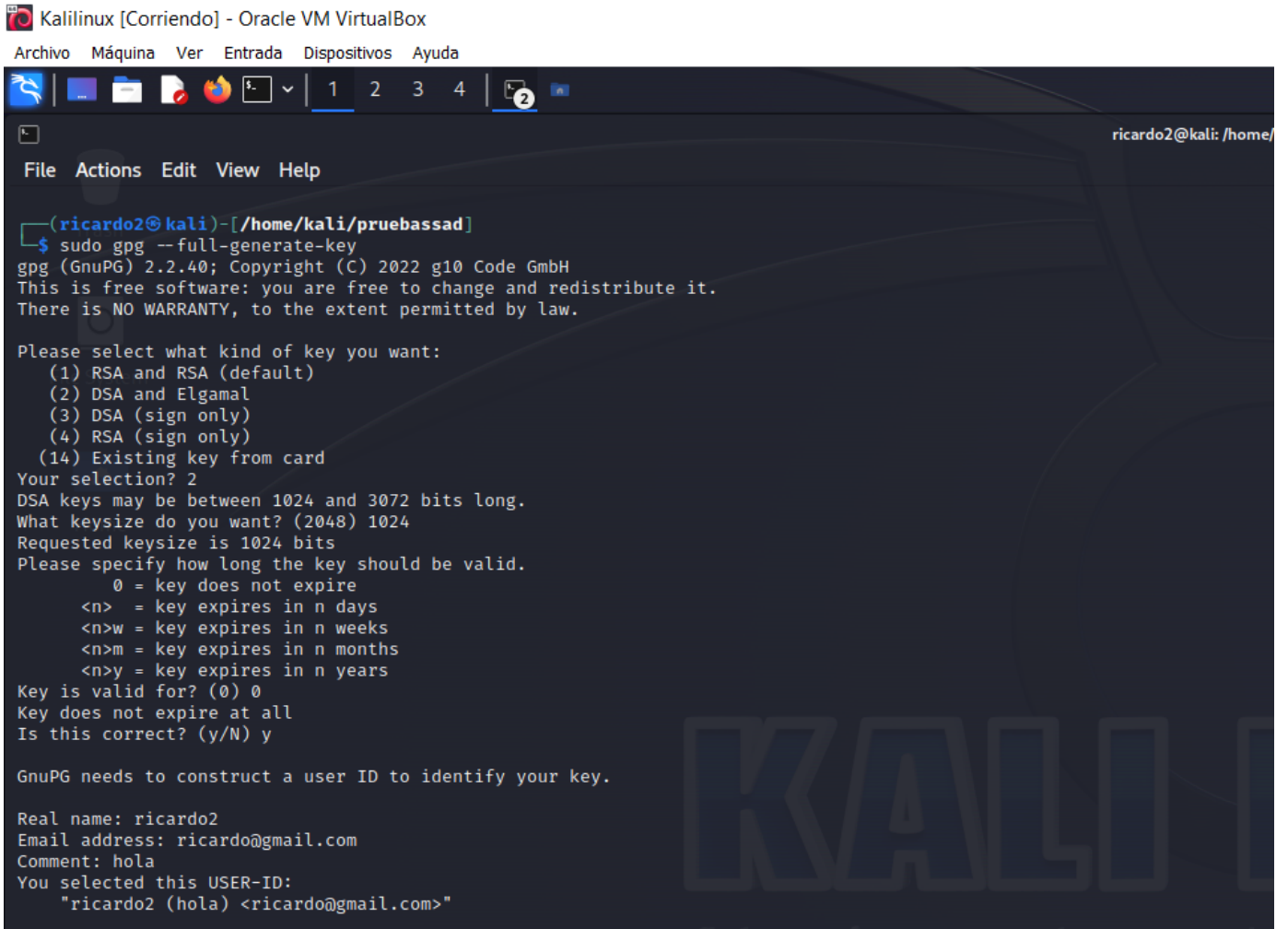
(kali@kali)~/pruebassad
$ ls
asimetricoricardo2.txt  prueba1.txt  prueba1.txt.gpg
(kali@kali)~/pruebassad
$ sudo rm -r asimetricoricardo2.txt
[sudo] password for kali:
(kali@kali)~/pruebassad
$ ls
prueba1.txt  prueba1.txt.gpg
(kali@kali)~/pruebassad
$ sudo touch asimetrico_kali.pub
(kali@kali)~/pruebassad
$ ls
asimetrico_kali.pub  prueba1.txt  prueba1.txt.gpg
(kali@kali)~/pruebassad
$
```

KALI LINUX

Ahora generaremos las claves necesarias (dos, una publica y otra privada) para cada usuario.

Primero crearemos las claves del primer usuario.

`$ gpg --full-generate-key`



```

Kalilinux [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

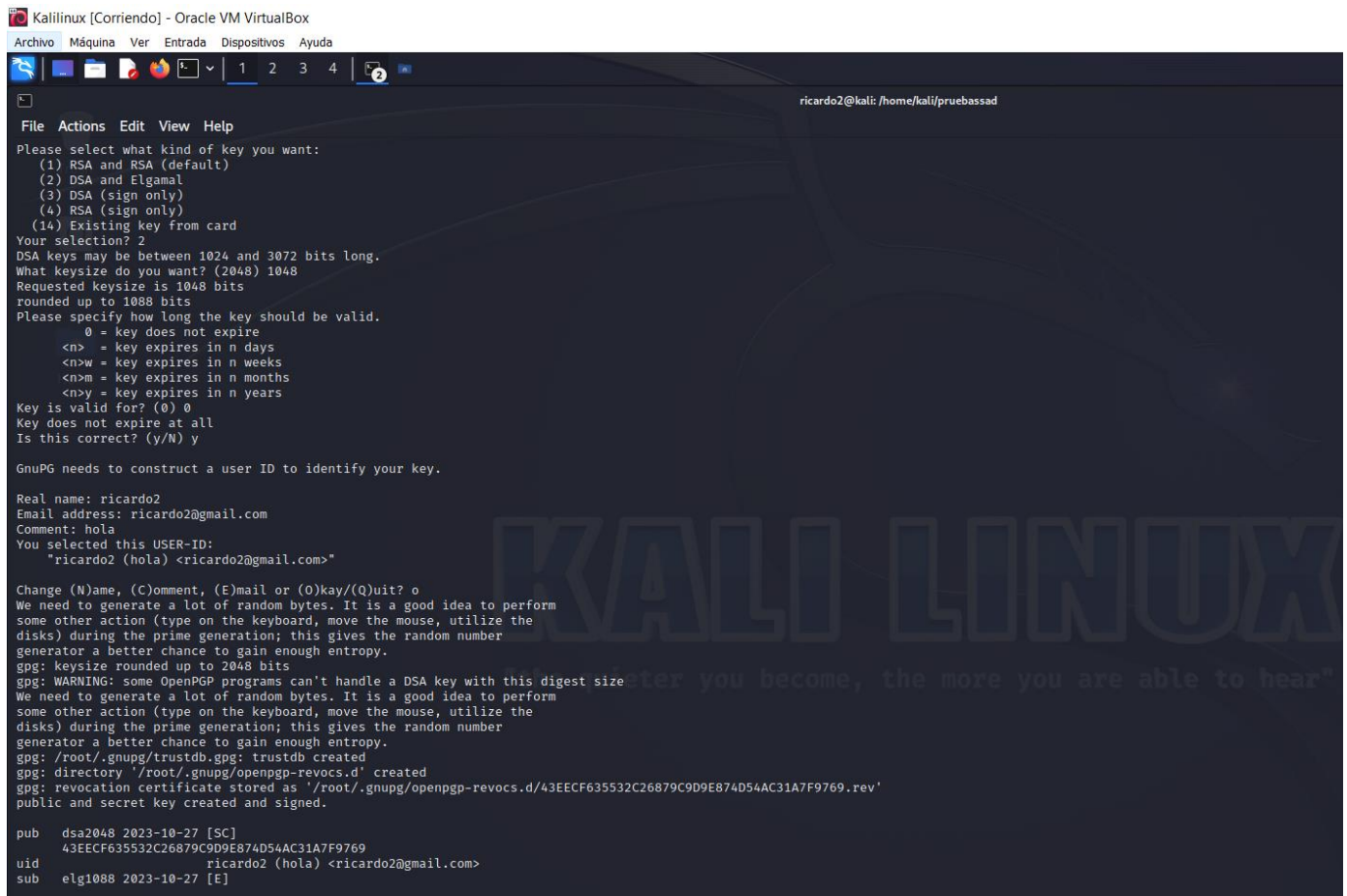
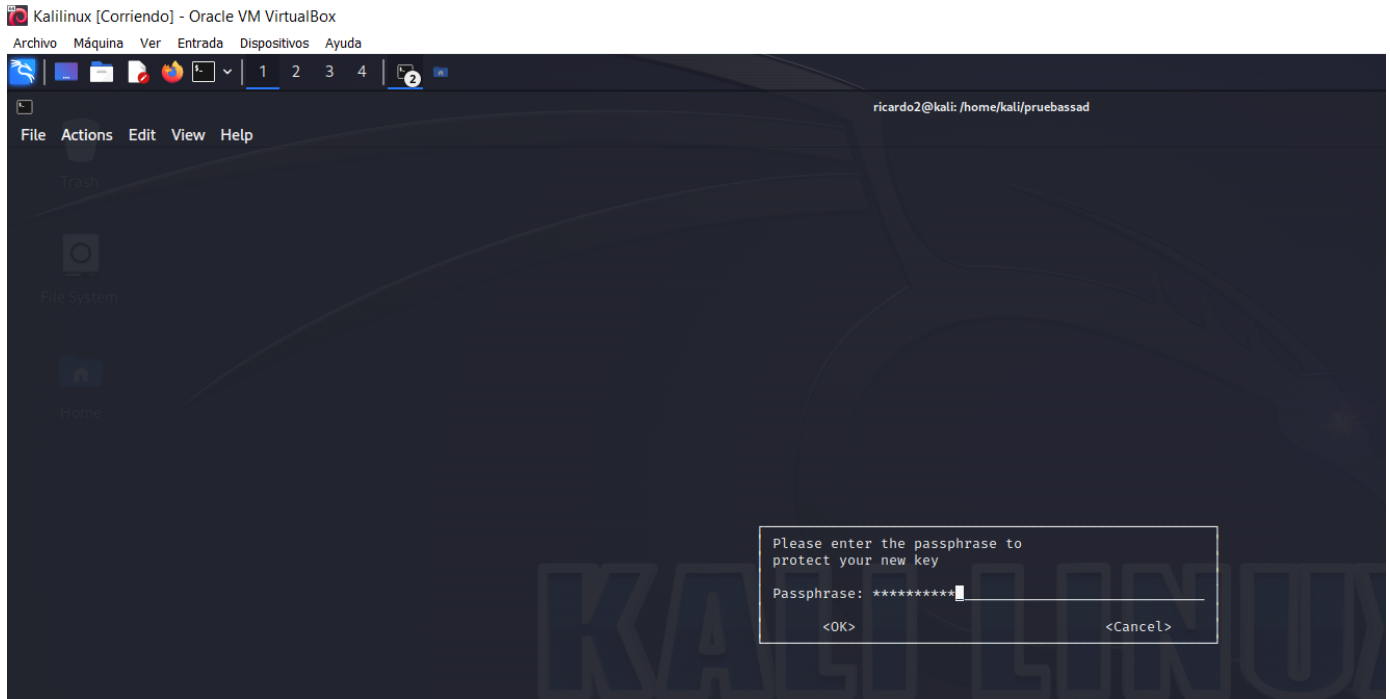
(ricardo2@kali) - [/home/kali/pruebassad]
$ sudo gpg --full-generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (14) Existing key from card
Your selection? 2
DSA keys may be between 1024 and 3072 bits long.
What keysize do you want? (2048) 1024
Requested keysize is 1024 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

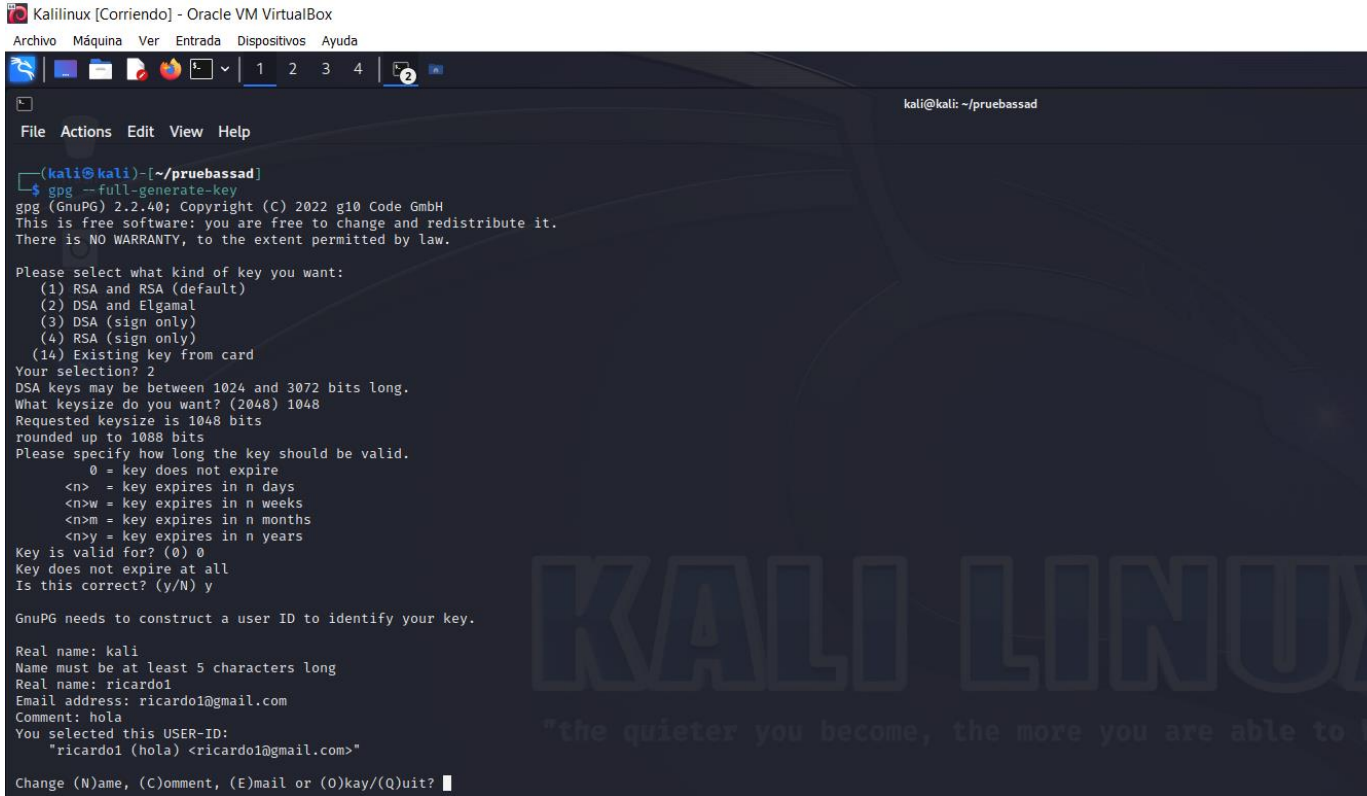
GnuPG needs to construct a user ID to identify your key.

Real name: ricardo2
Email address: ricardo@gmail.com
Comment: hola
You selected this USER-ID:
  "ricardo2 (hola) <ricardo@gmail.com>"
  
```

Le ponemos una contraseña.



Ahora, vamos al otro usuario (kali) y generaremos su clave igual que hicimos con ricardo2.



```

KaliLinux [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
kali@kali: ~/pruebassad

File  Actions  Edit  View  Help

(kali@kali)-[~/pruebassad]
$ gpg --full-generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (14) Existing key from card
Your selection? 2
DSA keys may be between 1024 and 3072 bits long.
What keysize do you want? (2048) 1048
Requested keysize is 1048 bits
rounded up to 1088 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: kali
Name must be at least 5 characters long
Real name: ricardo1
Email address: ricardo1@gmail.com
Comment: hola
You selected this USER-ID:
    "ricardo1 (hola) <ricardo1@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?

```

Le ponemos una contraseña.

Ya la tendremos generada.

KaliLinux [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

1 2 3 4 5

kali@kali: ~/pruebassad

File Actions Edit View Help

Please select what kind of key you want:

- (1) RSA and RSA (default)
- (2) DSA and Elgamal
- (3) DSA (sign only)
- (4) RSA (sign only)
- (14) Existing key from card

Your selection? 2

DSA keys may be between 1024 and 3072 bits long.

What keysize do you want? (2048) 1048

Requested keysize is 1048 bits

rounded up to 1088 bits

Please specify how long the key should be valid.

- 0 = key does not expire
- <n> = key expires in n days
- <n>w = key expires in n weeks
- <n>m = key expires in n months
- <n>y = key expires in n years

Key is valid for? (0) 0

Key does not expire at all

Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: ricardo1

Email address: ricardo1@gmail.com

Comment: hola

You selected this USER-ID:

"ricardo1 (hola) <ricardo1@gmail.com>"

Change (N)ame, (C)omment, (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.

gpg: keysize rounded up to 2048 bits

gpg: WARNING: some OpenPGP programs can't handle a DSA key with this digest size

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: /home/kali/.gnupg/trustdb.gpg: trustdb created

gpg: directory '/home/kali/.gnupg/openpgp-revocs.d' created

gpg: revocation certificate stored as '/home/kali/.gnupg/openpgp-revocs.d/3BA2EF470A07E1055D09A9C961B20FCC3588B2B5.rev'

public and secret key created and signed.

pub dsa2048 2023-10-27 [SC]

3BA2EF470A07E1055D09A9C961B20FCC3588B2B5

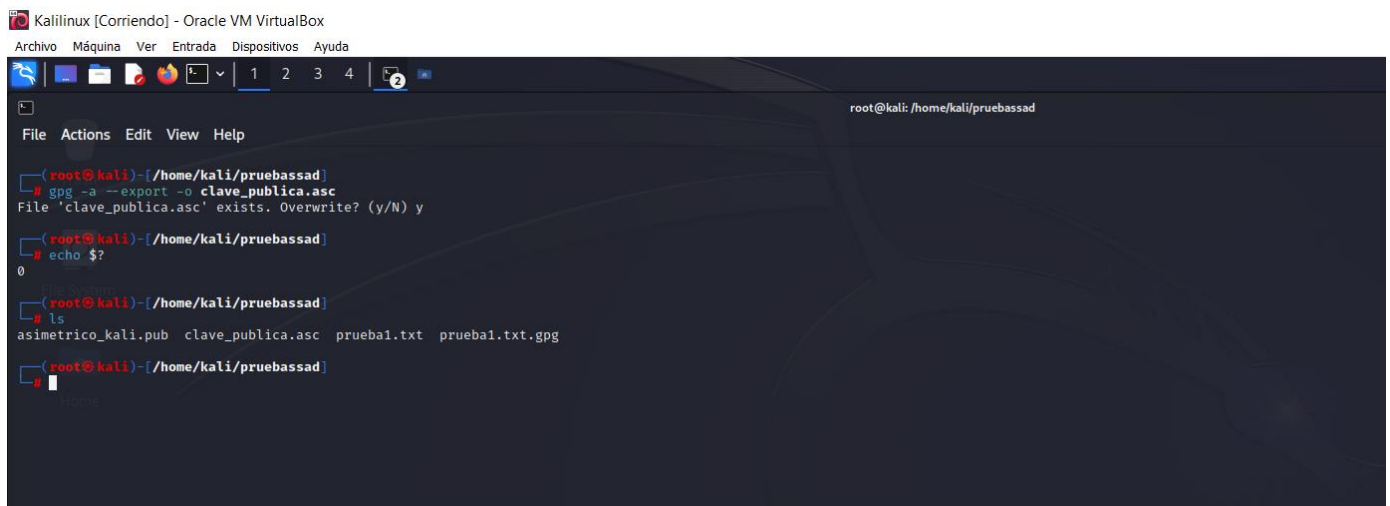
uid ricardo1 (hola) <ricardo1@gmail.com>

sub elg1088 2023-10-27 [E]

Ahora simplemente exportaremos las claves publicas de los dos usuarios.

kali:

```
$ gpg -a --export -o clavepublica.asc
```

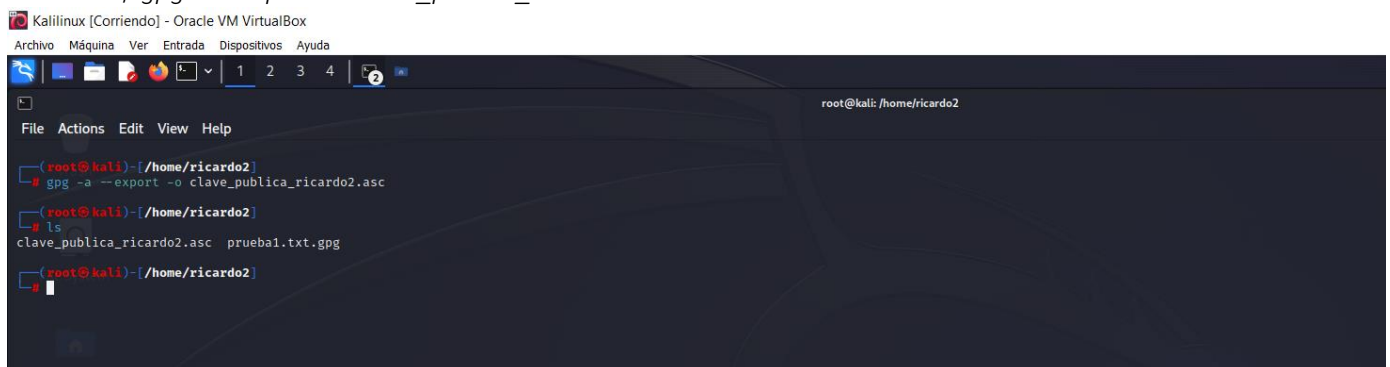


The screenshot shows a terminal window titled 'Kali Linux [Corriendo] - Oracle VM VirtualBox'. The user is logged in as root at kali. The terminal shows the following commands and output:

```
(root@kali)-[/home/kali/pruebassad]
$ gpg -a --export -o clave_publica.asc
File 'clave_publica.asc' exists. Overwrite? (y/N) y
(root@kali)-[/home/kali/pruebassad]
$ echo $?
0
(root@kali)-[/home/kali/pruebassad]
$ ls
asimetrico_kali.pub  clave_publica.asc  prueba1.txt  prueba1.txt.gpg
(root@kali)-[/home/kali/pruebassad]
```

Ricardo2:

```
$ gpg -a --export -o clave_publica_ricardo2.asc
```

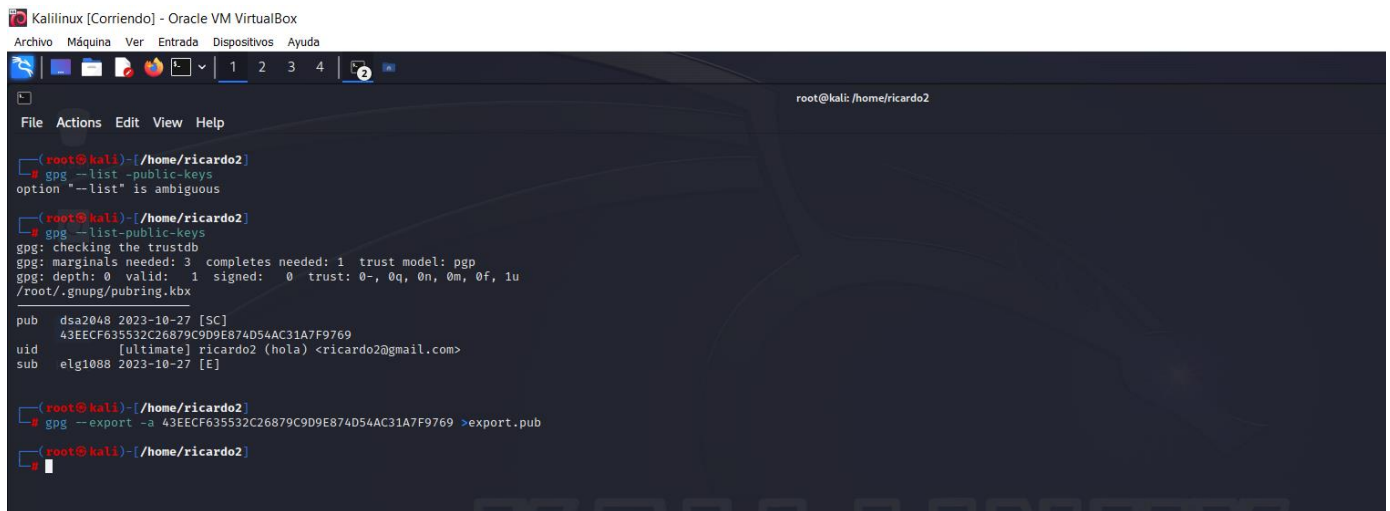


The screenshot shows a terminal window titled 'Kali Linux [Corriendo] - Oracle VM VirtualBox'. The user is logged in as root at kali. The terminal shows the following commands and output:

```
(root@kali)-[/home/ricardo2]
$ gpg -a --export -o clave_publica_ricardo2.asc
(root@kali)-[/home/ricardo2]
$ ls
clave_publica_ricardo2.asc  prueba1.txt.gpg
(root@kali)-[/home/ricardo2]
```

Recordamos que con el comando `gpg --list-public-keys` podemos ver las claves públicas que tenemos.

Podemos de esa forma exportar la clave publica que queramos.



```
Kalilinux [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
1 2 3 4 5
root@kali: /home/ricardo2

File Actions Edit View Help

root@kali: /home/ricardo2
$ gpg --list --public-keys
option "--list" is ambiguous

root@kali: /home/ricardo2
$ gpg --list-public-keys
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
/root/.gnupg/pubring.kbx

pub   dsa2048 2023-10-27 [SC]
      43EECF635532C26879C9D9E874D54AC31A7F9769
uid   [ultimate] ricardo02 (hola) <ricardo2@gmail.com>
sub   elg1088 2023-10-27 [E]

root@kali: /home/ricardo2
$ gpg --export -a 43EECF635532C26879C9D9E874D54AC31A7F9769 >export.pub

root@kali: /home/ricardo2
$
```


Ahora copiaremos las claves públicas de un usuario a otro. Clave publica de ricardo2 a kali.

The screenshot shows a terminal window titled 'Kali Linux [Corriendo] - Oracle VM VirtualBox'. The user is logged in as 'root@kali' in the directory '/home/ricardo2'. The terminal shows the following commands and output:

```

root@kali:~/home/ricardo2
$ cp export.pub /home/kali/export.pub
root@kali:~/home/ricardo2
$

```

The background of the terminal window features the Kali Linux logo and the slogan "the quieter you become, the more you are able to hear".

Clave publica de kali a ricardo2.

The screenshot shows a terminal window titled 'Kali Linux [Corriendo] - Oracle VM VirtualBox'. The user is logged in as 'kali@kali' in the directory '~/pruebassad'. The terminal shows the following commands and output:

```

kali@kali:~/pruebassad
$ ls
asimetrico_kali.pub  clave_publica.asc  pruebba1.txt  pruebba1.txt.gpg
kali@kali:~/pruebassad
$ sudo cp clave_publica.asc /home/ricardo2
[sudo] password for kali:
kali@kali:~/pruebassad
$

```

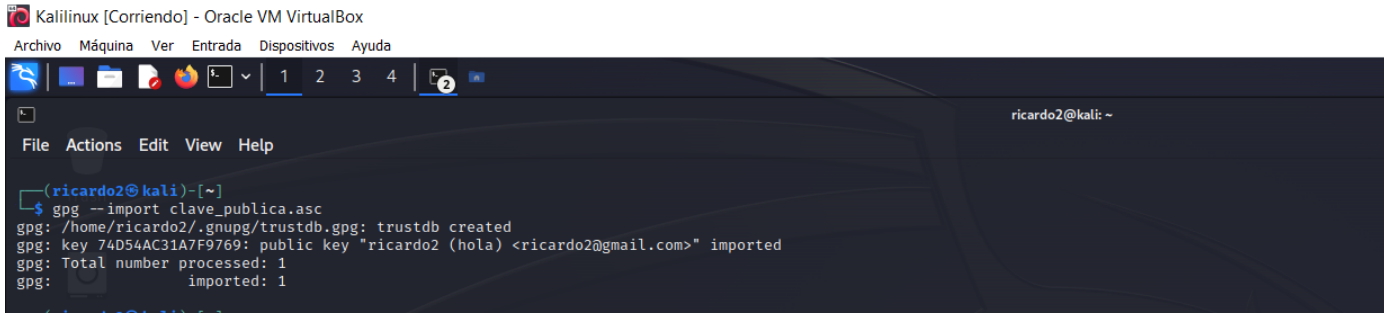
Ahora importaremos las claves copiadas a cada usuario.

The screenshot shows a terminal window titled 'Kali Linux [Corriendo] - Oracle VM VirtualBox'. The user is logged in as 'kali@kali' in the directory '~'. The terminal shows the following commands and output:

```

kali@kali:~
$ ls
Desktop  Documents  Downloads  export.pub  Music  Pictures  pruebassad  Public  Templates  Videos
kali@kali:~
$ gpg --import export.pub
gpg: key 74D54AC31A7F9769: public key "ricardo2 (hola) <ricardo2@gmail.com>" imported
gpg: Total number processed: 1
gpg:      imported: 1
kali@kali:~
$

```


A screenshot of a Kali Linux terminal window. The window title is 'Kali Linux [Corriendo] - Oracle VM VirtualBox'. The menu bar includes 'Archivo', 'Máquina', 'Ver', 'Entrada', 'Dispositivos', and 'Ayuda'. The terminal shows the user 'ricardo2@kali' at the prompt. The command 'gpg --import clave_publica.asc' has been executed, resulting in the following output: 'gpg: /home/ricardo2/.gnupg/trustdb.gpg: trustdb created', 'gpg: key 74D54AC31A7F9769: public key "ricardo2 (hola) <ricardo2@gmail.com>" imported', 'gpg: Total number processed: 1', and 'gpg: imported: 1'.

```
Kali Linux [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
ricardo2@kali: ~
File Actions Edit View Help
(ricardo2@kali)-[~]
$ gpg --import clave_publica.asc
gpg: /home/ricardo2/.gnupg/trustdb.gpg: trustdb created
gpg: key 74D54AC31A7F9769: public key "ricardo2 (hola) <ricardo2@gmail.com>" imported
gpg: Total number processed: 1
gpg: imported: 1
```

Ahora creamos un archivo nuevo en la cuenta kali y lo encriptaremos con la clave publica del usuario ricardo2 que acabamos de importar.

Lo encriptamos con la clave publica de ricardo2.

```
$ gpg -v -s -o hola.encrypt --encrypt --recipient franky hola.txt
```

Kali Linux [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

File Actions Edit View Help

```

(kali@kali)-[~]
$ cat hola.txt
archivo es para ricardo2

(kali@kali)-[~]
$ gpg -v -s -o hola.encrypt --encrypt --recipient ricardo2 hola.txt
gpg: using gpg trust model
gpg: using subkey 2D33DE5E8B361892 instead of primary key 74D54AC31A7F9769
gpg: 2D33DE5E8B361892: There is no assurance this key belongs to the named user

sub elg1088/2D33DE5E8B361892 2023-10-27 ricardo2 (hola) <ricardo2@gmail.com>
Primary key fingerprint: 43EE CF63 5532 C268 79C9 D9E8 74D5 4AC3 1A7F 9769
Subkey fingerprint: 8C8F 8933 C24A CF1D 94B6 CC55 2D33 DE5E 8B36 1892

It is NOT certain that the key belongs to the person named
in the user ID. If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
gpg: writing to 'hola.encrypt'
gpg: ELG/AES256 encrypted for: "2D33DE5E8B361892 ricardo2 (hola) <ricardo2@gmail.com>"
gpg: pinentry launched (185510 gnome3 1.2.1 /dev/pts/2 xterm-256color :0.0 20600/1000/5

```

Passphrase:

Please enter the passphrase to unlock the OpenPGP secret key:
"ricardo1 (hola) <ricardo1@gmail.com>"
2048-bit DSA key, ID 61B20FCC358BB2B5,
created 2023-10-27.

Password:

☐ Save in password manager

Cancel OK

Pasamos el archivo a ricardo2 y lo desciframos con su clave privada (lo podemos realizar ya que el archivo lo ciframos con la clave publica de ricardo2).

Kali Linux [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

File Actions Edit View Help

```

(ricardo2@kali)-[~]
$ ls
clave_publica.asc clave_publica_ricardo2.asc export.pub hola.encrypt prueba1.txt.gpg

(ricardo2@kali)-[~]
$ gpg --decrypt hola.encrypt
gpg: encrypted with 1088-bit ELG key, ID 2D33DE5E8B361892, created 2023-10-27
"ricardo2 (hola) <ricardo2@gmail.com>"
gpg: decryption failed: No secret key

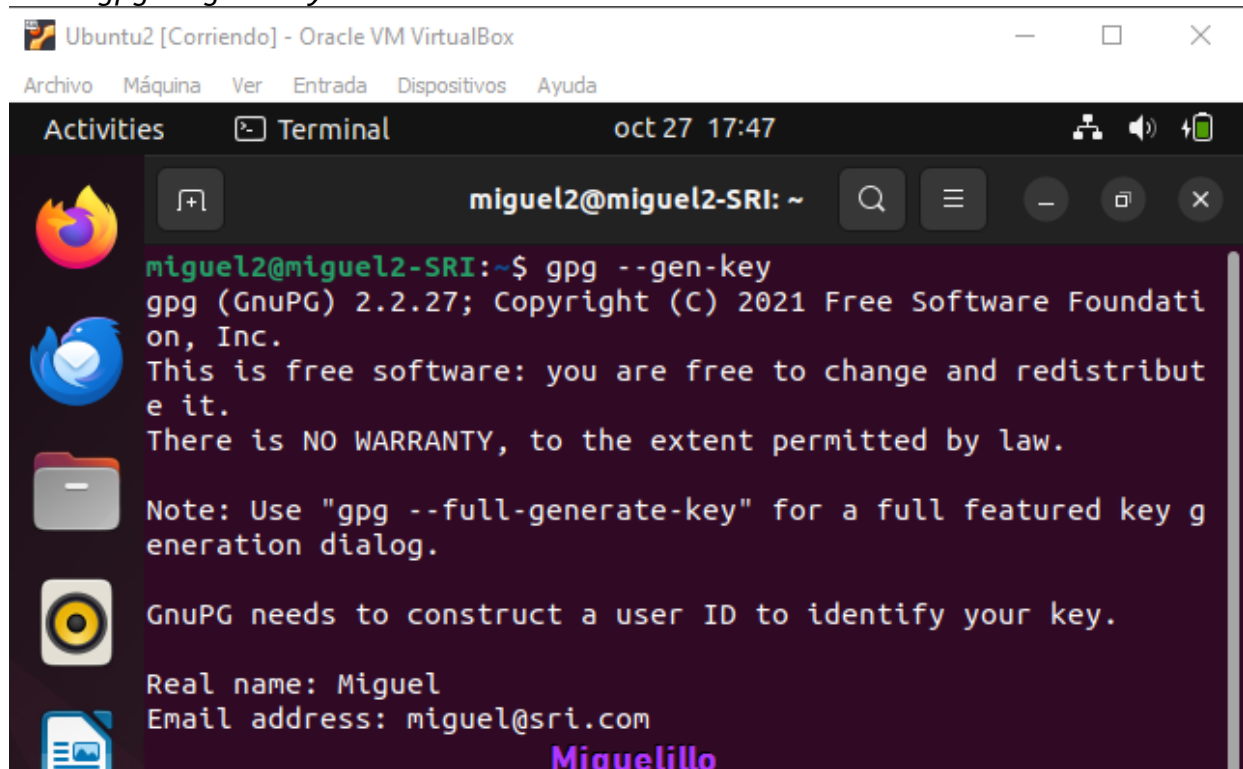
```

- 3. Utiliza un programa en Windows o GNU/Linux para simular la autenticidad mediante "cifrado asimétrico".

Para asegurar autenticidad usará una firma digital si el documento luego es modificado de cualquier forma, la verificación de la firma fallará

Para eso voy a utilizar gpg

Lo primero que haremos será generar un par de claves pública y privada lo haremos con `sudo gpg --gen-key`



```

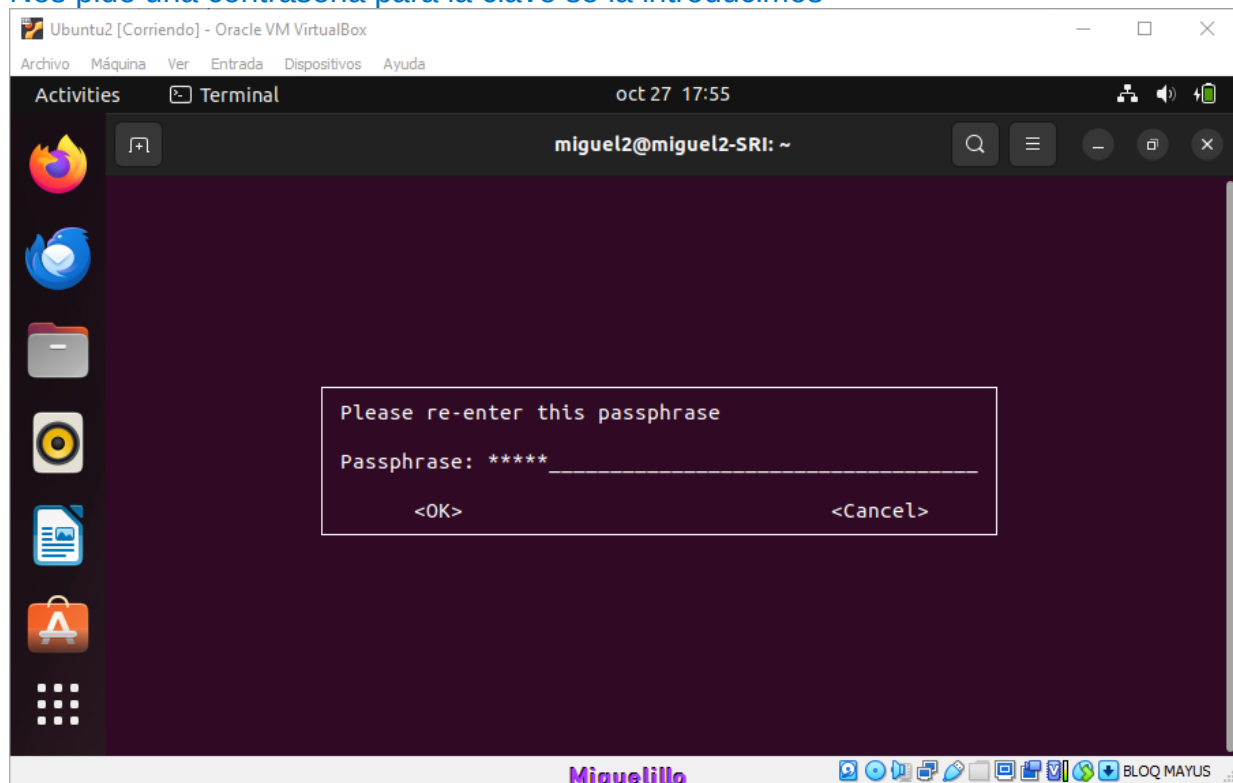
miguel2@miguel2-SRI: ~
miguel2@miguel2-SRI:~$ gpg --gen-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundati
on, Inc.
This is free software: you are free to change and redistribut
e it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key g
eneration dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Miguel
Email address: miguel@sri.com
Miguelillo
  
```

Nos pide una contraseña para la clave se la introducimos



```

miguel2@miguel2-SRI: ~
Please re-enter this passphrase
Passphrase: *****
<OK> <Cancel>
Miguelillo
  
```

Exportamos la clave pública con el comando

`sudo gpg --armor --export "Miguel <miguel@sri.com>" > tuclavepublica.asc`

```

Ubuntu2 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Activities  Terminal  oct 27 17:57

miguel2@miguel2-SRI: ~
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: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 6327C051D10592B2 marked as ultimately trusted
gpg: directory '/root/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/B59DA0D203B7C3B9499EE
9A46327C051D10592B2.rev'
public and secret key created and signed.

pub   rsa3072 2023-10-27 [SC] [expires: 2025-10-26]
      B59DA0D203B7C3B9499EE9A46327C051D10592B2
uid           Miguel <miguel@sri.com>
sub   rsa3072 2023-10-27 [E] [expires: 2025-10-26]

miguel2@miguel2-SRI:~$ gpg --armor --export "Miguel <miguel@sri.com>" > tuclavepublica.asc
gpg: WARNING: nothing exported
miguel2@miguel2-SRI:~$ sudo gpg --armor --export "Miguel <miguel@sri.com>" > tuclavepublic
a.asc
miguel2@miguel2-SRI:~$ ls
Desktop  Downloads  Pictures  snap          tuclavepublica.asc
Documents Music      Public    Templates  Videos
miguel2@miguel2-SRI:~$

```

Miguelillo

Importamos la clave publica al otro equipo con el comando

`gpg --import tuclavepublica.asc`

`gpg --list-keys`

```

Ubuntu [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Actividades  Terminal

usuario@usuario-miguel-SRI:~$ gpg --import tuclavepublica.asc
gpg: clave 6327C051D10592B2: "Miguel <miguel@sri.com>" sin cambios
gpg: Cantidad total procesada: 1
gpg:      sin cambios: 1
usuario@usuario-miguel-SRI:~$ gpg --list-keys
/home/usuario/.gnupg/pubring.kbx
-----
pub   rsa3072 2023-09-15 [SC] [caduca: 2025-09-14]
      B4DD9E234D9D0402F0D32FB44ABBE18C79E43C4D
uid           [ absoluta ] Miguel Lillo <miguel@gmail.com>
sub   rsa3072 2023-09-15 [E] [caduca: 2025-09-14]

pub   rsa3072 2023-09-15 [SC] [caduca: 2025-09-14]
      8E08CEDBDD97F6912C8F6907B8986FF103480F72
uid           [ absoluta ] usuario <usuario@gmail.com>
sub   rsa3072 2023-09-15 [E] [caduca: 2025-09-14]

pub   rsa3072 2023-10-27 [SC] [caduca: 2025-10-26]
      B59DA0D203B7C3B9499EE9A46327C051D10592B2
uid           [desconocida] Miguel <miguel@sri.com>
sub   rsa3072 2023-10-27 [E] [caduca: 2025-10-26]

```

Miguelillo

Para ver las claves públicas que tenemos disponibles ponemos el siguiente comando

```
sudo gpg --list-secret-keys
```

Firmo el archivo indicando el nombre y correo, el nombre del archivo y del resultado con el siguiente comando

```
sudo gpg --clearsign miguelSRI
```

Se crea un archivo con la extensión .asc ara ver el contenido y que se ha firmado

```
cat miguelSRI.asc
```

```

Ubuntu2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal oct 27 19:58
miguel2@miguel2-SRI: ~
gpg: miguelSRI: clear-sign failed: No secret key
miguel2@miguel2-SRI:~$ sudo gpg --clearsign miguelSRI
[sudo] password for miguel2:
miguel2@miguel2-SRI:~$ cat miguelSRI.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Este es el documento firmado de Miguel
-----BEGIN PGP SIGNATURE-----
iQGzBAEBCgAdFiEETz2g0gO3w7LJnumkYyFAUdEFkrIFAmU79d0ACgkQYyFAUdEF
krKpTAV+ISmJPvB9Ego7Srv6PonvmEFnn2pP8vB//Fvclyahrt1Co7fucZ3MY0ip
GxQSTuhYkI9dK1KvaEYqw3N+DEYZDGV8F31JJJuMyb39GsYEW0t3oKHz+tpnKHX0
tXb9TsJrWS2dRzUP+Way0H18CEMmidoMmEMHQ1MVA/q8FYt1uzLYvQ51e7IShYpN
nSwZ1ALCMg9ahfQzVE1LTk3FV27svw5RDzdLLTyFdLp3i/D7eqXNKvW58453Uk4P
47dkIws164kdw0hV7fdjyW1GB0QhhW85NwD0oRuJc864dku57hW77232FXpgQIq
V4tSy5n7Utvfheen/kwd3YG48MsG0ThqIiG064WNf5oKLRHw3z17b0zUW5uunjP
Y0Drpqb1vshlWuHtKghdgD6nunhmD5okTrznnCDUHy4kmj2ZRTwMrbejJUj+cdx
Wqn4DPm9rLCmp9TLJ7iNTMptXPb1hn0JEo1A+NzMuFmdqxnH6NijsBPG3gVP9vSF
HLALuhBF
=aLXL
-----END PGP SIGNATURE-----
miguel2@miguel2-SRI:~$

```

Para ver la clave que lo ha firmado y el remitente usamos el siguiente comando:

```
sudo gpg --verify miguelSRI.asc
```

```

Ubuntu [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal 27 de oct 20:03
usuario@usuario-miguel-SRI: ~
usuario@usuario-miguel-SRI:~$ cat miguelSRI.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

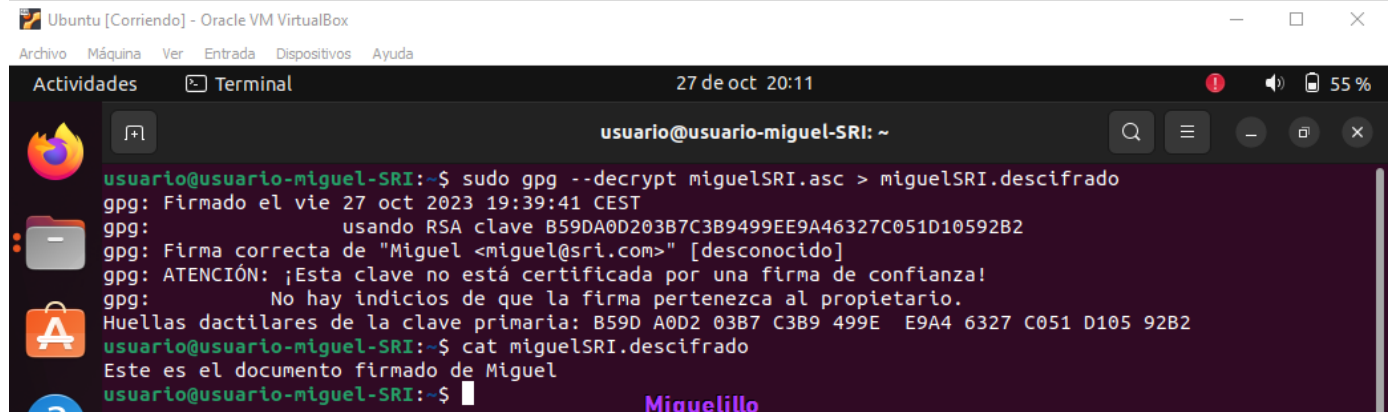
Este es el documento firmado de Miguel
-----BEGIN PGP SIGNATURE-----
iQGzBAEBCgAdFiEETz2g0gO3w7LJnumkYyFAUdEFkrIFAmU79d0ACgkQYyFAUdEF
krKpTAV+ISmJPvB9Ego7Srv6PonvmEFnn2pP8vB//Fvclyahrt1Co7fucZ3MY0ip
GxQSTuhYkI9dK1KvaEYqw3N+DEYZDGV8F31JJJuMyb39GsYEW0t3oKHz+tpnKHX0
tXb9TsJrWS2dRzUP+Way0H18CEMmidoMmEMHQ1MVA/q8FYt1uzLYvQ51e7IShYpN
nSwZ1ALCMg9ahfQzVE1LTk3FV27svw5RDzdLLTyFdLp3i/D7eqXNKvW58453Uk4P
47dkIws164kdw0hV7fdjyW1GB0QhhW85NwD0oRuJc864dku57hW77232FXpgQIq
V4tSy5n7Utvfheen/kwd3YG48MsG0ThqIiG064WNf5oKLRHw3z17b0zUW5uunjP
Y0Drpqb1vshlWuHtKghdgD6nunhmD5okTrznnCDUHy4kmj2ZRTwMrbejJUj+cdx
Wqn4DPm9rLCmp9TLJ7iNTMptXPb1hn0JEo1A+NzMuFmdqxnH6NijsBPG3gVP9vSF
HLALuhBF
=aLXL
-----END PGP SIGNATURE-----
usuario@usuario-miguel-SRI:~$ sudo gpg --verify miguelSRI.asc
[sudo] contraseña para usuario:
Lo siento, pruebe otra vez.
[sudo] contraseña para usuario:
gpg: Firmado el vie 27 oct 2023 19:39:41 CEST
gpg: usando RSA clave B59DA0D203B7C3B9499EE9A46327C051D10592B2
gpg: Firma correcta de "Miguel <miguel@sri.com>" [desconocido]
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de confianza!
gpg: No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: B59D A0D2 03B7 C3B9 499E E9A4 6327 C051 D105 92B2
usuario@usuario-miguel-SRI:~$ sudo gpg --output miguelSRI.asc --decrypt miguelSRI.asc

```

Aparece como firma correcta, la clave no es de confianza ya que no es de un distribuidor de claves de confianza ya que la he creado yo

Para ver el contenido lo desenscriptamos

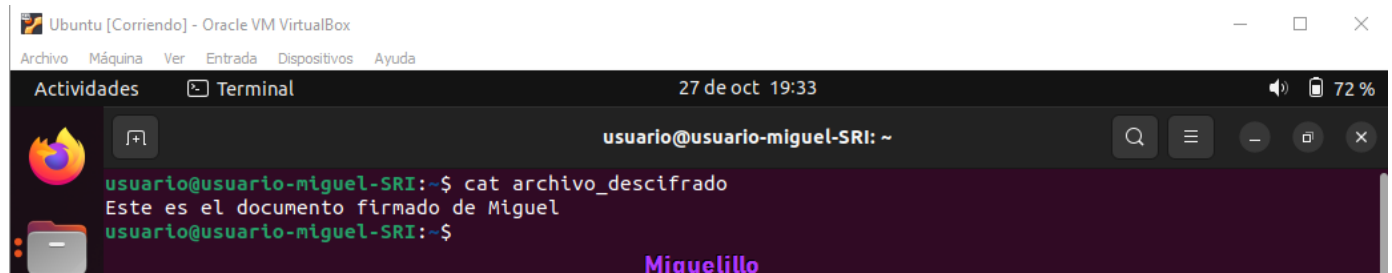
`sudo gpg --decrypt miguelSRI.asc > miguelSRI.descifrado`



The screenshot shows a terminal window titled 'usuario@usuario-miguel-SRI: ~'. The user has executed the command `sudo gpg --decrypt miguelSRI.asc > miguelSRI.descifrado`. The terminal output shows the GPG decryption process, including the RSA key used, the signature verification, and the fingerprint of the primary key. The user then runs `cat miguelSRI.descifrado`, and the terminal displays the decrypted content: 'Este es el documento firmado de Miguel'.

```
usuario@usuario-miguel-SRI:~$ sudo gpg --decrypt miguelSRI.asc > miguelSRI.descifrado
gpg: Firmado el vie 27 oct 2023 19:39:41 CEST
gpg: usando RSA clave B59DA0D203B7C3B9499EE9A46327C051D10592B2
gpg: Firma correcta de "Miguel <miguel@sri.com>" [desconocido]
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de confianza!
gpg: No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: B59D A0D2 03B7 C3B9 499E E9A4 6327 C051 D105 92B2
usuario@usuario-miguel-SRI:~$ cat miguelSRI.descifrado
Este es el documento firmado de Miguel
usuario@usuario-miguel-SRI:~$
```

Con un Cat veo el contenido



The screenshot shows a terminal window titled 'usuario@usuario-miguel-SRI: ~'. The user has executed the command `cat archivo_descifrado`. The terminal output shows the content of the file: 'Este es el documento firmado de Miguel'.

```
usuario@usuario-miguel-SRI:~$ cat archivo_descifrado
Este es el documento firmado de Miguel
usuario@usuario-miguel-SRI:~$
```


- Utiliza un programa en Windows o GNU/Linux para simularla “autenticidad”+“confidencialidad” mediante “cifrado asimétrico”.

Para autenticidad + confidencialidad primero firmamos con la clave privada de Miguel miguel@asir.com

```

Ubuntu2 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal  oct 27 20:13
migu2@migu2-SRI:~$ sudo gpg --clearsign miguelSRI
[sudo] password for migu2:
File 'miguelSRI.asc' exists. Overwrite? (y/N) y
migu2@migu2-SRI:~$
  
```

Verificamos que se ha creado bien

```

Ubuntu2 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal  oct 27 20:16
migu2@migu2-SRI:~$ file miguelSRI*
miguelSRI:      ASCII text
miguelSRI.asc:  PGP signed message
migu2@migu2-SRI:~$
  
```

Ahora ciframos el documento firmado con la clave publica de la otra máquina para que este pueda descifrarlo.

```

Ubuntu2 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal  oct 27 20:25
migu2@migu2-SRI:~$ file miguelSRI*
miguelSRI:      ASCII text
miguelSRI.asc:  PGP signed message
migu2@migu2-SRI:~$ sudo gpg --import clavemiguelillo.asc
gpg: key C0AFE8FF4905403B: public key "miguelillo <miguelillo@sri.com>" imported
gpg: Total number processed: 1
gpg:      imported: 1
  
```

Lo ciframos con

`sudo gpg -v -a -o miguelSRI.cifrado --encrypt --recipient miguelillo miguelSRI.asc`

```

Ubuntu2 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Activities  Terminal  oct 27 20:28
migu2@migu2-SRI:~$ sudo gpg -v -a -o miguelSRI.cifrado --encrypt --recipient miguelillo miguelSRI.asc
gpg: using gpg trust model
gpg: using subkey A9FD8307E867410D instead of primary key C0AFE8FF4905403B
gpg: A9FD8307E867410D: There is no assurance this key belongs to the named user

sub rsa3072/A9FD8307E867410D 2023-10-27 miguelillo <miguelillo@sri.com>
Primary key fingerprint: D204 08D9 8595 83CD E37D  61C5 C0AF E8FF 4905 403B
Subkey fingerprint: B2F3 89DC C69B C29A 876F  E932 A9FD 8307 E867 410D

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
gpg: reading from 'miguelSRI.asc'
gpg: writing to 'miguelSRI.cifrado'
gpg: RSA/AES256 encrypted for: "A9FD8307E867410D miguelillo <miguelillo@sri.com>"
migu2@migu2-SRI:~$
  
```

Vemos con cat que se ha cifrado

```

Ubuntu2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

oct 27 20:31
miguel2@miguel2-SRI: ~

miguel2@miguel2-SRI:~$ cat miguelSRI.cifrado
-----BEGIN PGP MESSAGE-----

hQGMA6n9gwf0Z0ENAQv7BJLuWJVp7r6LLQeRcwvuu14TkqjFo72rtTEbK+whV3sY
MHzhst+61kdIdHdHhNoeYCyT3oCvg22344GzK+gfcDPLcqVMLHunFp82ZyIpNvnc
4mPhDKyfojt6abyfba06vVf7vufbWf0ZzLroDAksu2IB4h2SoGQlfbg+WMmfzbHg
NBuUJ4ZIAA3rJcI3cHe1QpI3FtVnWfVdLDAvB0FR6D19ETOZWMXdeAKRuWVdcjQ8
PiqfLLfFBCe7vY2kSLDFBoAN+3BWh3PW1wDBJn02RqdoxnAATT62PwukV9x3Ab+
xcBMQwEDenx9RNAi3mabHNJp5rNoyQYr6q7AcLbRpmJd7wT13B3ryohy4uptnUT
XilLwLXRzydLgYrgr2leyPv7krwLXaUlhyhGK0GSwx8bECJmXgiqx7PUu5V7zuvM
HjmFv4i7QA9zYjZlvztkqLMGMTh3AWFt0Gh8IP67r88Bs6d9q0tty67T06eccNAh
qlENqMjcpJUrKFR6ffjC0ukBJEh2MurWcsbFXV4tFr86FiYaQumHJFCYxtKjF8f6
PmigdJUEZ/NRMwwBBZla+0Yny+EBnTQLz4id1sZ08pVLIOy05YwE3c3d++gh5IQx
/wOV2013aibuLgFNl+s3kHbe/WqRSEfYLPxy9oX4uL59d1wnY2gx/arnEFwJzls
5TjJNmWm9fgr4Y9Vlo8HeRnbj0ISu9N/0Zvg9gAFK0+A9i2wdUECsdL1frA6rZ
VrPL2fjoUJ0ywp0CgMwhKjJQ369PpaHvmkZQ9XS4usHLDpLpQuBuvYzktjYjueVh
RplLWwVigwobeCLZWpl5HWg+k7qma1DxneGrh9McNshlBqZ0jP8+NLI8i/FGoVWT
ubIdvr0jU5FXNR3dvH/4e6LhBn9fSLDcyZLVp41jhpNttfH0+f3ZW1ouGSp8ohN
y4PFpFLmqZ8aKe0Sd6RUE+7Y/TE10cgNBfVb0W3WP7JbfIQ0VqI66P6Qn2Cq52ch
NYZPNfKA8AtBAquZJLMKjvnoVX/VBpwPER+8kcIWgjKrF5dtuEaa/R3l7GNMMPV
srPwBj/AmnCcyqZnmKj3QVr67++jVHviw2DZq6FXZdeupewz7ePeMwUcrTFJwgPG
M3000151rBBp7JZRXfIZ0+6CZ0bGt7w6sDK8BFXW30aM3FNd02pvQ==
=J0ED
-----END PGP MESSAGE-----
miguel2@miguel2-SRI:~$

```

Miguelillo

Pasamos el archivo al otro equipo y lo desencriptamos

```
gpg --output autenticidad.desencriptado --decrypt autenticidad.encrypt
```

```

Ubuntu [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

27 de oct 20:35
usuario@usuario-miguel-SRI: ~

usuario@usuario-miguel-SRI:~$ gpg --output miguelSRI.cifrado --decrypt miguelSRI.cifrado
gpg: cifrado con clave RSA, ID A9FD8307E867410D
gpg: descifrado fallido: No tenemos la clave secreta
usuario@usuario-miguel-SRI:~$ sudo gpg --output miguelSRI.cifrado --decrypt miguelSRI.cifrado
gpg: cifrado con clave de 3072 bits RSA, ID A9FD8307E867410D, creada el 2023-10-27
"miguelillo <miguelillo@sri.com>"
El fichero 'miguelSRI.cifrado' ya existe. ¿Sobreescribir? (s/N) n
Introduzca nuevo nombre de fichero: miguelSRI.sri
usuario@usuario-miguel-SRI:~$

```

Miguelillo

```

27 de oct 20:34
usuario@usuario-miguel-SRI: ~

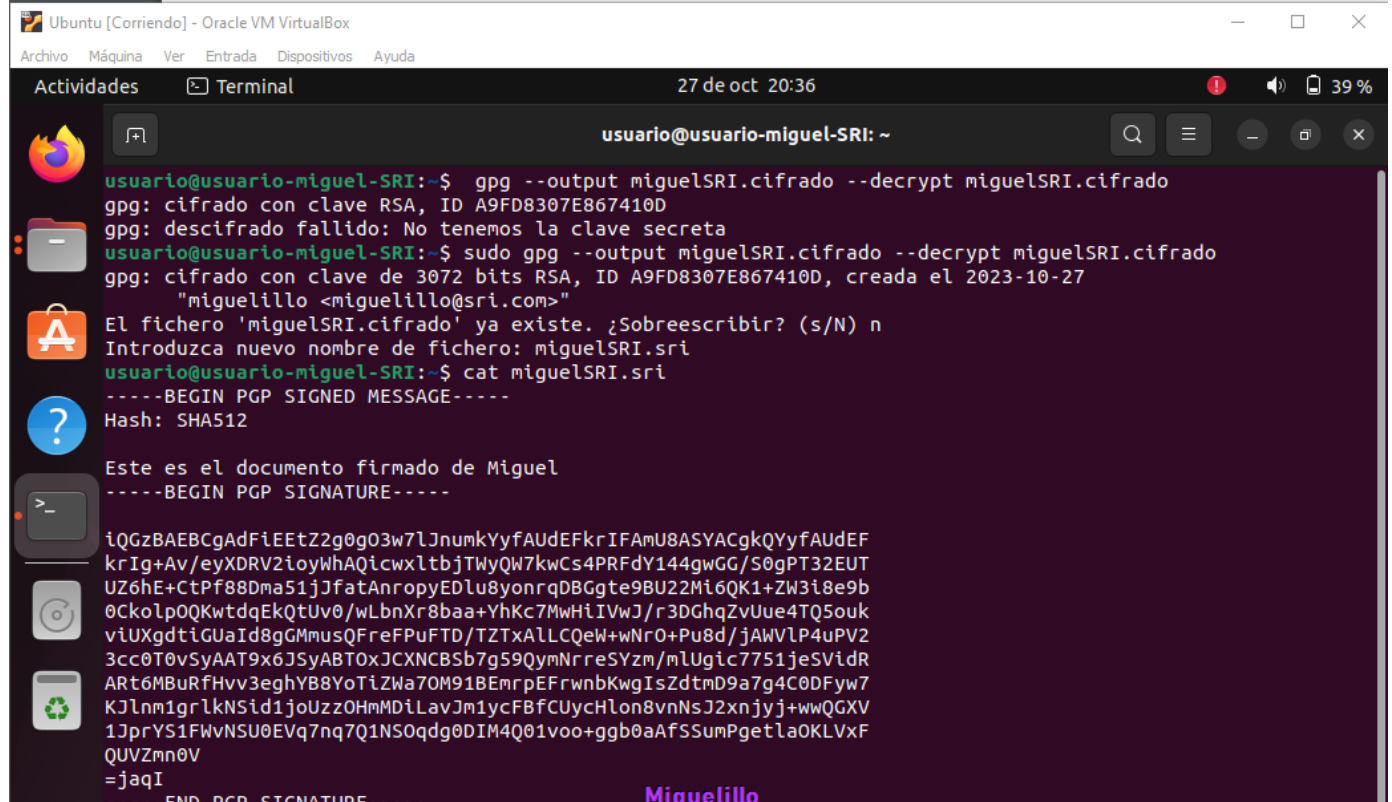
Introduzca frase contraseña para desbloquear la clave secreta OpenPGP:
"miguelillo <miguelillo@sri.com>"
clave de 3072-bit RSA, ID A9FD8307E867410D,
creada el 2023-10-27 (ID de clave primaria C0AFE8FF4905403B).

Frase de paso: *****
<OK> <Cancelar>

```

Miguelillo

Si hacemos un cat podemos ver el contenido



```

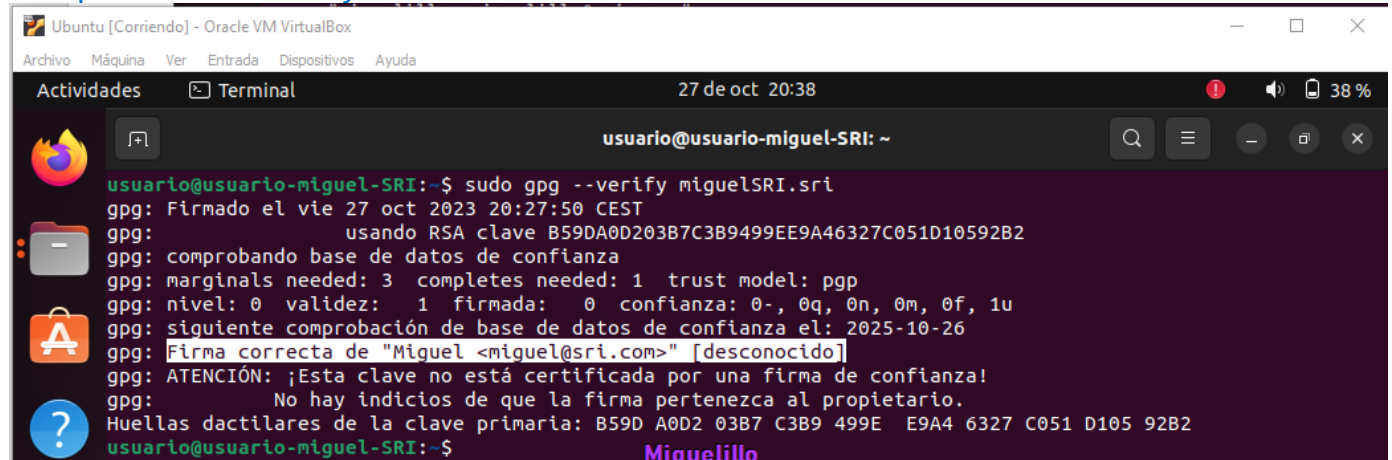
usuario@usuario-miguel-SRI: ~
usuario@usuario-miguel-SRI:~$ gpg --output miguelSRI.cifrado --decrypt miguelSRI.cifrado
gpg: cifrado con clave RSA, ID A9FD8307E867410D
gpg: descifrado fallido: No tenemos la clave secreta
usuario@usuario-miguel-SRI:~$ sudo gpg --output miguelSRI.cifrado --decrypt miguelSRI.cifrado
gpg: cifrado con clave de 3072 bits RSA, ID A9FD8307E867410D, creada el 2023-10-27
"miguelillo <miguelillo@sri.com>"
El fichero 'miguelSRI.cifrado' ya existe. ¿Sobreescribir? (s/N) n
Introduzca nuevo nombre de fichero: miguelSRI.sri
usuario@usuario-miguel-SRI:~$ cat miguelSRI.sri
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Este es el documento firmado de Miguel
-----BEGIN PGP SIGNATURE-----

iQGzBAEBCgAdFiEEtZ2g0gO3w7LJnumkYyfAUdEFkrIFAMU8ASYACgkQYyfAUdEF
krIg+Av/eyXDRV2ioywhAQicwxtbjTWyQW7kwCs4PRFdy144gwGG/S0gPT32EUT
UZ6hE+CtPf88Dma51jJfatAnropyEDLu8yonrqDBGgte9BU22Mi6QK1+ZW3i8e9b
0CkolpOQKwtDqEkQtUv0/wLbnXr8baa+YhKc7MwHiIVwJ/r3DGhqZvUue4TQSouk
viUXgdtiGUaId8gGmmusQFreFPuFTD/TZTxALLCQeW+wNrO+Pu8d/jAWVLP4uPV2
3cc0T0vSyAAT9x6JSyABTOxJCXNCBSb7g59QymNrreSYzm/mlUgic7751jeSVidR
ARt6MBuRfHvv3eghYB8YoTiZwa7OM91BEmrpEfrwnbKwgIsZdtmD9a7g4C0DFyw7
KJlNm1grlKnsid1joUzz0HmMDiLavJm1ycFBfCUycHlon8vnNsJ2xnjy+wwQGxV
1JprYS1FWvNSU0EVq7nq7Q1NS0qdqg0DIM4Q01voo+ggb0aAfSSumPgetlaOKLVxF
QUVZmn0V
=jaqI
-----END PGP SIGNATURE-----
Miguelillo

```

Comprobamos la firma y es correcta.



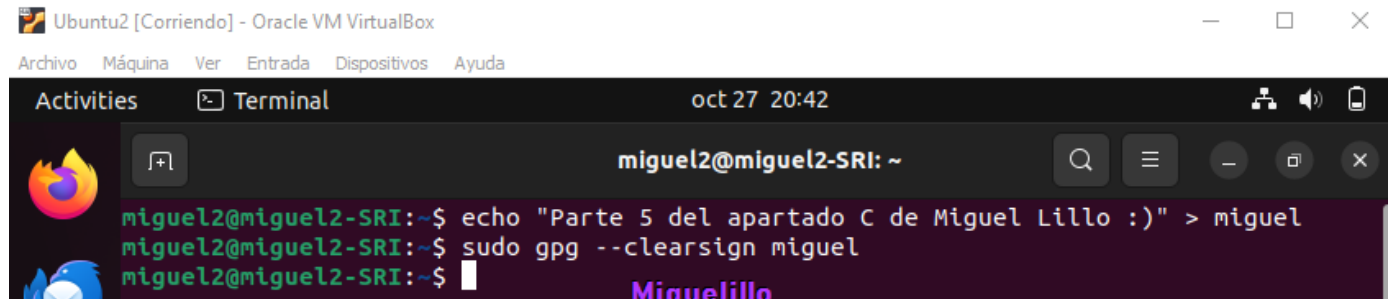
```

usuario@usuario-miguel-SRI: ~
usuario@usuario-miguel-SRI:~$ sudo gpg --verify miguelSRI.sri
gpg: Firmado el vie 27 oct 2023 20:27:50 CEST
gpg: usando RSA clave B59DA0D203B7C3B9499EE9A46327C051D10592B2
gpg: comprobando base de datos de confianza
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: nivel: 0 validez: 1 firmada: 0 confianza: 0-, 0q, 0n, 0m, 0f, 1u
gpg: siguiente comprobación de base de datos de confianza el: 2025-10-26
gpg: Firma correcta de "Miguel <miguel@sri.com>" [desconocido]
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de confianza!
gpg: No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: B59D A0D2 03B7 C3B9 499E E9A4 6327 C051 D105 92B2
usuario@usuario-miguel-SRI:~$
Miguelillo

```

5. Utiliza un programa en Windows o GNU/Linux para simular una comunicación segura utilizando cifrados híbridos “autenticidad” + “confidencialidad”+“integridad”: asimétricos (clave pública) y simétricos (clave privada).

Creemos un archivo y lo firmamos



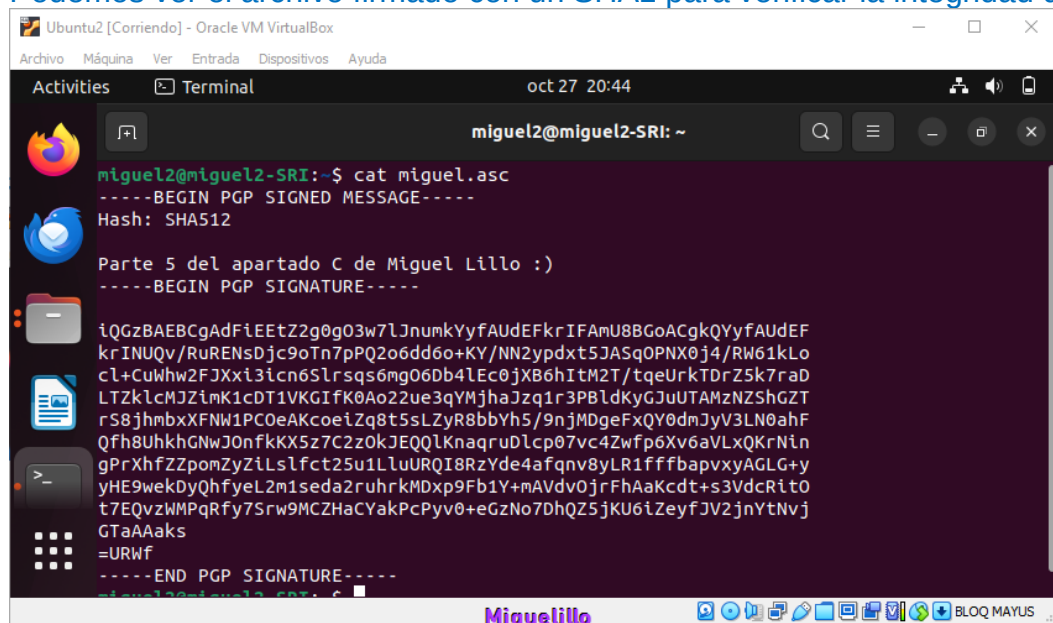
```

Ubuntu2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal oct 27 20:42
miguel2@miguel2-SRI: ~
miguel2@miguel2-SRI:~$ echo "Parte 5 del apartado C de Miguel Lillo :)" > miguel
miguel2@miguel2-SRI:~$ sudo gpg --clearsign miguel
miguel2@miguel2-SRI:~$

```

Podemos ver el archivo firmado con un SHA1 para verificar la integridad del contenido y la firma



```

Ubuntu2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

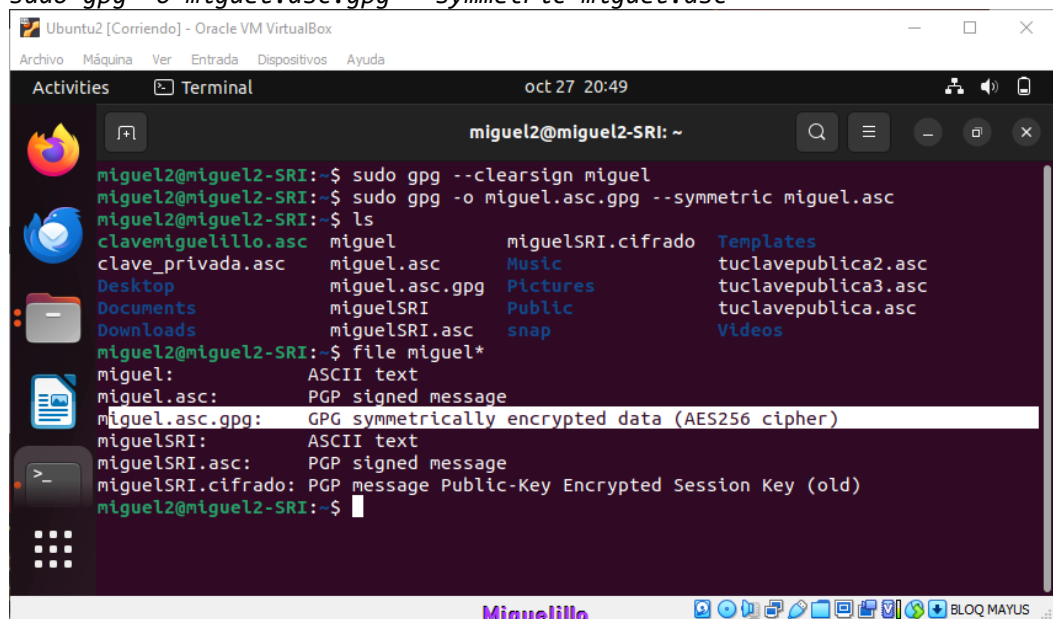
Activities Terminal oct 27 20:44
miguel2@miguel2-SRI: ~
miguel2@miguel2-SRI:~$ cat miguel.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Parte 5 del apartado C de Miguel Lillo :)
-----BEGIN PGP SIGNATURE-----

iQGzBAEBCgAdFiEetZ2g03w7lJnumkYyFAUdEFkrIFAmU8BGACgkQYyFAUdEF
krINUQv/RuRENSDjc9oTn7pPQ2o6dd6o+KY/NN2ypdxt5JASqOPNX0j4/RW61kLo
cL+CuWhw2FJXxi3icn6Slrsqs6mg06Db4LEc0jXB6hITM2T/tqeUrktDrZ5k7raD
LTZkLcM3ZimK1cDT1VKGIK0Ao22ue3qYMjhaJzq1r3PBldKyGJuUTAMzNZShGZT
rS8jhmbxXFNW1PCoeAKcoeIZq8t5sLZyR8bbYh5/9njMDgeFxQY0dmJyV3LN0ahF
QfH8UhhkGNWJ0nfKX5z7C2z0kJEQQLKnaqrUDlcp07vc4Zwfp6Xv6aVLxQKrNin
gPrXhfZZpomZyZiLsLfct25u1LluURQI8RzYde4afqnv8yLR1ffffbapvxyAGLG+y
yHE9wekDyQhfyEL2m1seda2ruhrkMDxp9Fb1Y+mAVdv0jrFhAaKcdt+s3VdcRit0
t7EQvzWMPqRfy7Srw9MCZHaCYakPcPyv0+eGzNo7DhQZ5jKU6iZeyfJv2jnYtNvj
GTaAAaks
=URWf
-----END PGP SIGNATURE-----

```

Lo siguiente es hacer un cifrado simétrico para que sea un cifrado híbrido. Con el comando:
`sudo gpg -o miguel.asc.gpg --symmetric miguel.asc`



```

Ubuntu2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal oct 27 20:49
miguel2@miguel2-SRI: ~
miguel2@miguel2-SRI:~$ sudo gpg --clearsign miguel
miguel2@miguel2-SRI:~$ sudo gpg -o miguel.asc.gpg --symmetric miguel.asc
miguel2@miguel2-SRI:~$ ls
clavemiguelillo.asc  miguel          miguelSRI.cifrado  Templates
clave_privada.asc    miguel.asc      Music               tuclavepublica2.asc
Desktop              miguel.asc.gpg  Pictures            tuclavepublica3.asc
Documents             miguelSRI       Public              tuclavepublica.asc
Downloads             miguelSRI.asc   snap               Videos
miguel2@miguel2-SRI:~$ file miguel*
miguel:          ASCII text
miguel.asc:      PGP signed message
miguel.asc.gpg:  GPG symmetrically encrypted data (AES256 cipher)
miguelSRI:       ASCII text
miguelSRI.asc:   PGP signed message
miguelSRI.cifrado: PGP message Public-Key Encrypted Session Key (old)
miguel2@miguel2-SRI:~$

```

Ahora con las claves haremos un cifrado asimétrico con la clave pública de miguelillo con el comando:

```

michel2@michel2-SRI: ~
michel2@michel2-SRI:~$ sudo gpg -v -a -o miguel.asc.gpg.as --encrypt --recipient miguelillo miguel.asc.gpg
gpg: using gpg trust model
gpg: using subkey A9FD8307E867410D instead of primary key C0AFE8FF4905403B
gpg: A9FD8307E867410D: There is no assurance this key belongs to the named user

sub rsa3072/A9FD8307E867410D 2023-10-27 miguelillo <miguelillo@sri.com>
Primary key fingerprint: D204 08D9 8595 83CD E37D 61C5 C0AF E8FF 4905 403B
Subkey fingerprint: B2F3 89DC C69B C29A 876F E932 A9FD 8307 E867 410D

It is NOT certain that the key belongs to the person named
in the user ID. If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
gpg: reading from 'miguel.asc.gpg'
gpg: writing to 'miguel.asc.gpg.as'
gpg: RSA/AES256 encrypted for: "A9FD8307E867410D miguelillo <miguelillo@sri.com>"
michel2@michel2-SRI:~$

```

Revisamos el tipo de archivo

```

michel2@michel2-SRI: ~
michel2@michel2-SRI:~$ file miguel.asc*
miguel.asc:      PGP signed message
miguel.asc.gpg:  GPG symmetrically encrypted data (AES256 cipher)
miguel.asc.gpg.as: PGP message Public-Key Encrypted Session Key (old)
michel2@michel2-SRI:~$

```

Generamos el código hash del archivo y pasamos los dos archivos al otro equipo

`sha1sum miguel.asc.gpg.as > hashmiguel`

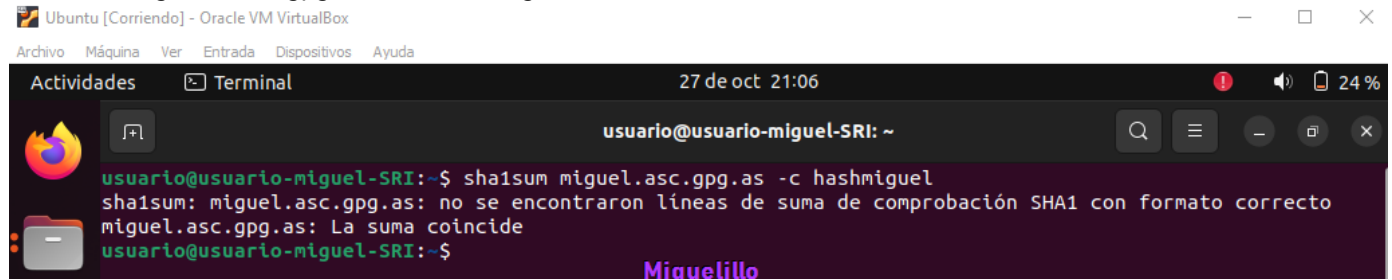
```

michel2@michel2-SRI: ~
michel2@michel2-SRI:~$ sha1sum miguel.asc.gpg.as > hashmiguel
michel2@michel2-SRI:~$ cat hashmiguel
904c8582efb881922483ddd256167d93da72930c  miguel.asc.gpg.as
michel2@michel2-SRI:~$

```

Una vez en este equipo comparamos el hash para comprobar que no ha sido alterado el archivo con el comando

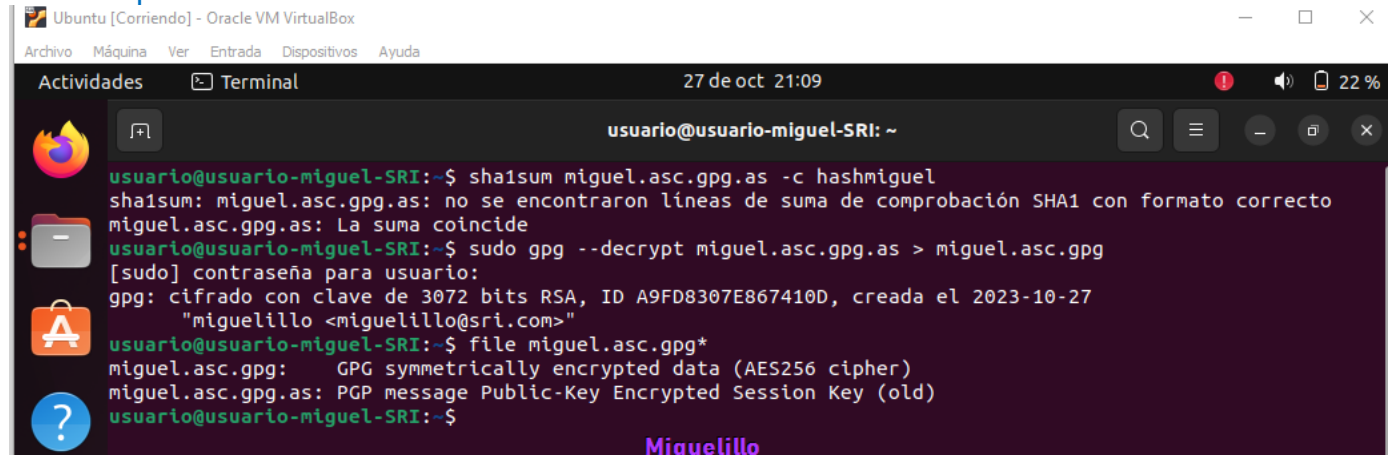
```
sha1sum miguel.asc.gpg.as -c hashmiguel
```



Terminal window showing the command `sha1sum miguel.asc.gpg.as -c hashmiguel` and its output:

```
usuario@usuario-miguel-SRI:~$ sha1sum miguel.asc.gpg.as -c hashmiguel
sha1sum: miguel.asc.gpg.as: no se encontraron líneas de suma de comprobación SHA1 con formato correcto
miguel.asc.gpg.as: La suma coincide
usuario@usuario-miguel-SRI:~$
```

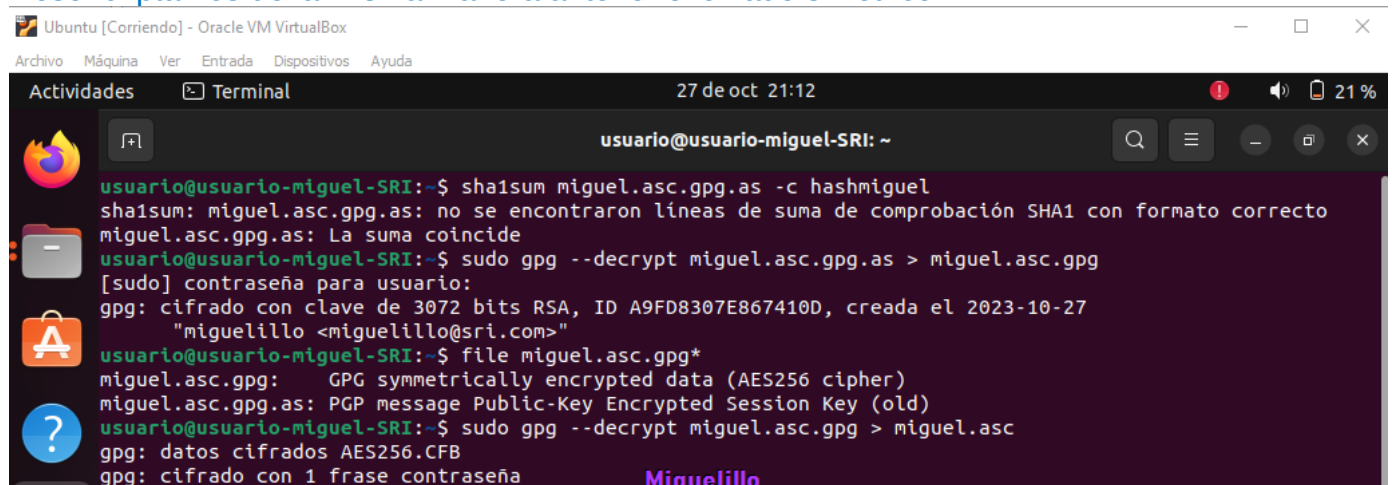
Desencriptamos el cifrado asimétrico



Terminal window showing the decryption process:

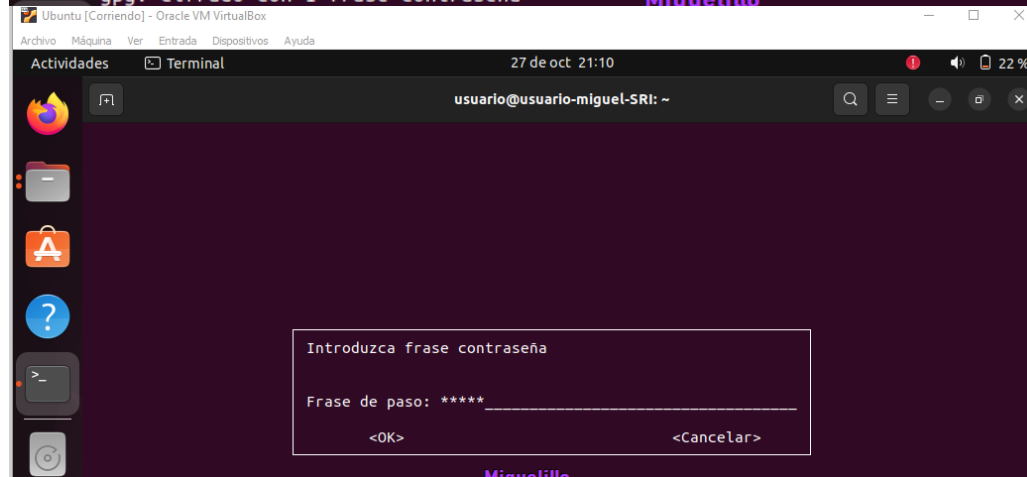
```
usuario@usuario-miguel-SRI:~$ sha1sum miguel.asc.gpg.as -c hashmiguel
sha1sum: miguel.asc.gpg.as: no se encontraron líneas de suma de comprobación SHA1 con formato correcto
miguel.asc.gpg.as: La suma coincide
usuario@usuario-miguel-SRI:~$ sudo gpg --decrypt miguel.asc.gpg.as > miguel.asc.gpg
[sudo] contraseña para usuario:
gpg: cifrado con clave de 3072 bits RSA, ID A9FD8307E867410D, creada el 2023-10-27
"miguelillo <miguelillo@sri.com>"
usuario@usuario-miguel-SRI:~$ file miguel.asc.gpg*
miguel.asc.gpg: GPG symmetrically encrypted data (AES256 cipher)
miguel.asc.gpg.as: PGP message Public-Key Encrypted Session Key (old)
usuario@usuario-miguel-SRI:~$
```

Desencriptamos de la misma manera anterior el cifrado simétrico



Terminal window showing the decryption process:

```
usuario@usuario-miguel-SRI:~$ sha1sum miguel.asc.gpg.as -c hashmiguel
sha1sum: miguel.asc.gpg.as: no se encontraron líneas de suma de comprobación SHA1 con formato correcto
miguel.asc.gpg.as: La suma coincide
usuario@usuario-miguel-SRI:~$ sudo gpg --decrypt miguel.asc.gpg.as > miguel.asc.gpg
[sudo] contraseña para usuario:
gpg: cifrado con clave de 3072 bits RSA, ID A9FD8307E867410D, creada el 2023-10-27
"miguelillo <miguelillo@sri.com>"
usuario@usuario-miguel-SRI:~$ file miguel.asc.gpg*
miguel.asc.gpg: GPG symmetrically encrypted data (AES256 cipher)
miguel.asc.gpg.as: PGP message Public-Key Encrypted Session Key (old)
usuario@usuario-miguel-SRI:~$ sudo gpg --decrypt miguel.asc.gpg > miguel.asc
gpg: datos cifrados AES256.CFB
gpg: cifrado con 1 frase contraseña
```



Terminal window showing the password prompt:

```
Introduzca frase contraseña
Frase de paso: *****
<OK> <Cancelar>
```


Podemos ver el SHA1 para compararlo con la firma y verificar la autenticidad de la firma

Ubuntu [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

```

Actividades Terminal 27 de oct 21:13
usuario@usuario-miguel-SRI: ~

usuario@usuario-miguel-SRI:~$ cat miguel.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Parte 5 del apartado C de Miguel Lillo :)
-----BEGIN PGP SIGNATURE-----

iQGzBAEBCgAdFiEETZ2g0gO3w7LJnumKYyFAUdEFkRIFAmU8BhIACgkQYyFAUdEF
krLPzAv/bN4Z6rr9qhU7ku1TyMPX9FHD8FamJ4QECY6x9plb9J6dYw1sYDFvKIF5
vu27LTB10WHZ39DNEXmAA4pwh4WlNgixwLxmE+IqB+KRkoDnL4rfNFpvhzMg8L6V
qr5nSdcVwc0HERN0+J0rrNGoCLpX9DF3Em20QxmyFNDnGjo9bS3LP+JgEelStAgU
s003qy8EV0XrtVCUDsXB0YJaF0D8eWvs+lweavJ3FjkWfVqKI+uugtNcV9rRztur
CS1DNINR99YpjXiBpcvAtmICg+3xnAyI+2FwgJRw53K8TPcp/5PthFtLpBu6BViZ
4vPWJAiRcdx5WtXVh+sAZzBnasqv5nzWEpiSCUALpShRvbhyc4dwyJIra6+HaHXZ
yLnzQy4NbLJknWzCaYFIo146HMscqOCg+VNF6aBjJqs31egKcUKmSBwVE7/+1Qt/
CS5JCIkp0QfrzJjymUUYwRrpUne7L85w+KA5C30p5pnw8SD3p9BvLAEmYZHNWnMx
mNuR4NhY
=UFNU
-----END PGP SIGNATURE-----
Miguelillo

```

Podemos ver que la firma es válida

Ubuntu [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

```

Actividades Terminal 27 de oct 21:15
usuario@usuario-miguel-SRI: ~

usuario@usuario-miguel-SRI:~$ sudo gpg --verify miguel.asc
gpg: Firmado el vie 27 oct 2023 20:48:50 CEST
gpg: usando RSA clave B59DA0D203B7C3B9499EE9A46327C051D10592B2
gpg: Firma correcta de "Miguel <miguel@sri.com>" [desconocido]
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de confianza!
gpg: No hay indicios de que la firma pertenezca al propietario.
Huellas dactilares de la clave primaria: B59D A0D2 03B7 C3B9 499E E9A4 6327 C051 D105 92B2
usuario@usuario-miguel-SRI:~$
Miguelillo

```

También podemos visualizar el mensaje

Ubuntu [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

```

Actividades Terminal 27 de oct 21:13
usuario@usuario-miguel-SRI: ~

usuario@usuario-miguel-SRI:~$ cat miguel.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Parte 5 del apartado C de Miguel Lillo :)
-----BEGIN PGP SIGNATURE-----

iQGzBAEBCgAdFiEETZ2g0gO3w7LJnumKYyFAUdEFkRIFAmU8BhIACgkQYyFAUdEF
krLPzAv/bN4Z6rr9qhU7ku1TyMPX9FHD8FamJ4QECY6x9plb9J6dYw1sYDFvKIF5
vu27LTB10WHZ39DNEXmAA4pwh4WlNgixwLxmE+IqB+KRkoDnL4rfNFpvhzMg8L6V
qr5nSdcVwc0HERN0+J0rrNGoCLpX9DF3Em20QxmyFNDnGjo9bS3LP+JgEelStAgU
s003qy8EV0XrtVCUDsXB0YJaF0D8eWvs+lweavJ3FjkWfVqKI+uugtNcV9rRztur
CS1DNINR99YpjXiBpcvAtmICg+3xnAyI+2FwgJRw53K8TPcp/5PthFtLpBu6BViZ
4vPWJAiRcdx5WtXVh+sAZzBnasqv5nzWEpiSCUALpShRvbhyc4dwyJIra6+HaHXZ
yLnzQy4NbLJknWzCaYFIo146HMscqOCg+VNF6aBjJqs31egKcUKmSBwVE7/+1Qt/
CS5JCIkp0QfrzJjymUUYwRrpUne7L85w+KA5C30p5pnw8SD3p9BvLAEmYZHNWnMx
mNuR4NhY
=UFNU
-----END PGP SIGNATURE-----
Miguelillo

```