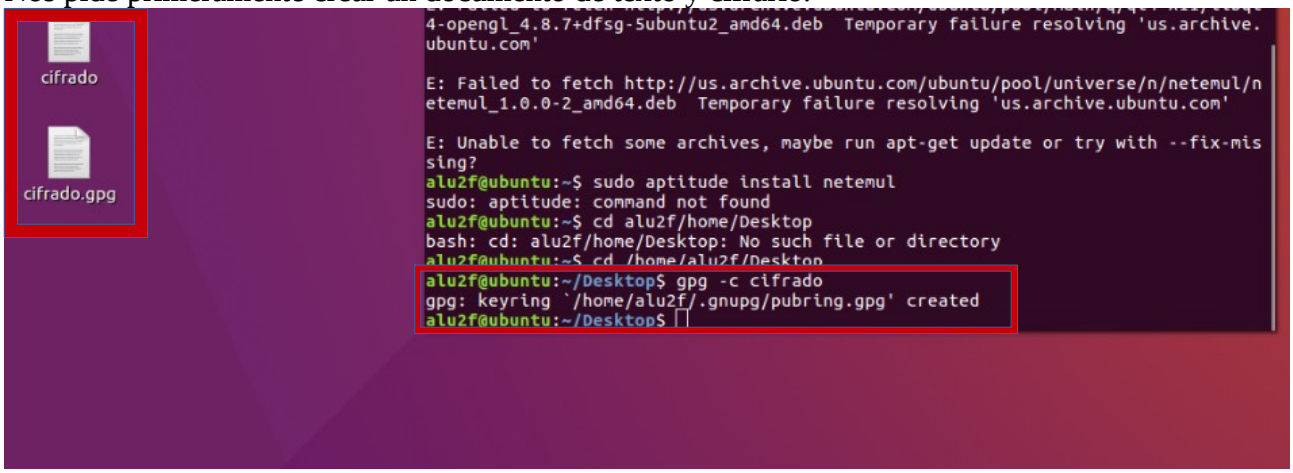
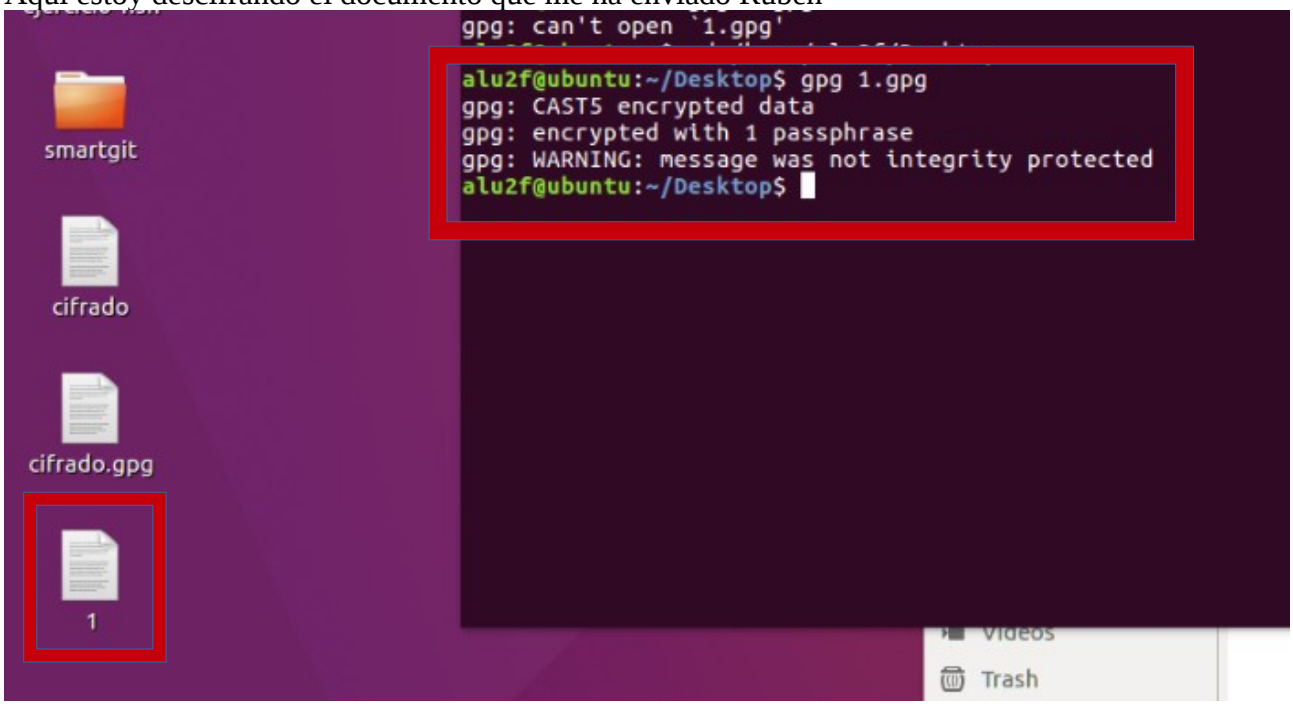
PRÁCTICA DE CRIPTOGRAFÍA

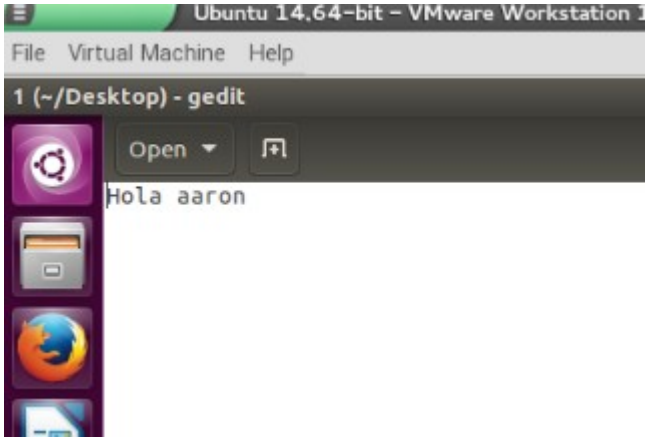
1)El ejercicio 1 lo realicé con mi compañero Rubén.
Nos pide primeramente crear un documento de texto y cifrarlo.



Aquí estoy descifrando el documento que me ha enviado Rubén



Al descifrarlo muestra el mensaje del documento de texto.



Despues tenemos que cifrarlo otra vez pero añadiendo la opción -a.

```
alu2f@ubuntu: ~/Desktop
alu2f@ubuntu:~$ gpg 1.gpg
gpg: can't open `1.gpg'
alu2f@ubuntu:~$ cd /home/alu2f/Desktop
alu2f@ubuntu:~/Desktop$ gpg 1.gpg
gpg: CAST5 encrypted data
gpg: encrypted with 1 passphrase
alu2f@ubuntu:~/Desktop$ gpg -ca cifrado
alu2f@ubuntu:~/Desktop$
```

Seguidamente mostramos el contenido con la opción cat y comprobamos que se ha cifrado correctamente

```
alu2f@ubuntu:~/Desktop$ gpg -ca cifrado
alu2f@ubuntu:~/Desktop$ cat 1.asc
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.11 (GNU/Linux)

jA0EAwMCZWkInTAQfJFgySJg2ghFe49D/1HEFrZgqvP6qk32mw0AxGD6RlsxLMxC
5XXF
=5Tbt
-----END PGP MESSAGE-----
alu2f@ubuntu:~/Desktop$
```

2)El ejercicio 2 nos pide crear un par de claves pública y privada, y que tenga de validez 1 mes. Tenemos que usar el comando gpg –gen-key.

```
Virtual Machine   Help
2f@ubuntu: ~/Desktop

alu2f@ubuntu:~/Desktop$ gpg --gen-key
gpg (GnuPG) 1.4.20; Copyright (C) 2015 Free Software Foundation, Inc.
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)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 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) 1m
Key expires at Thu 06 Apr 2017 12:34:20 PM PDT
Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Aaron
Email address: heyron53@gmail.com
Comment:
You selected this USER-ID:
    "Aaron <heyron53@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.

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.
```

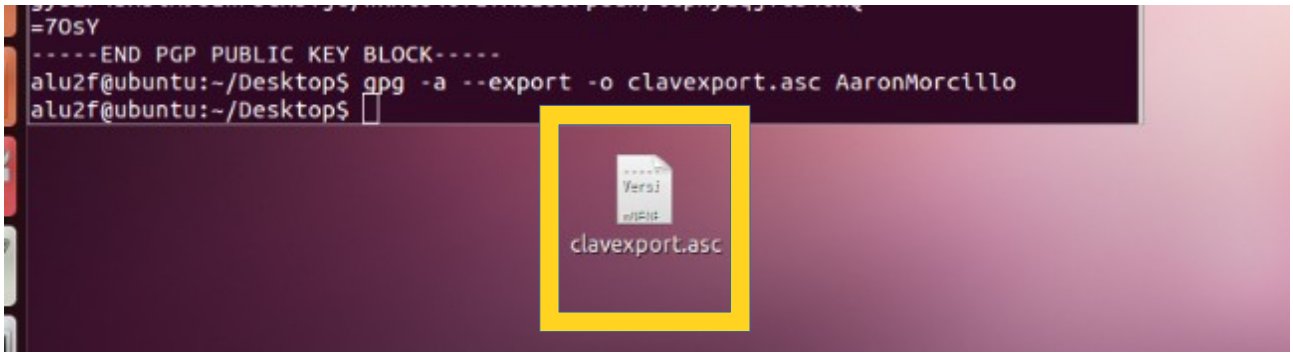
Aquí marcamos que queremos que la clave dure 1 mes.

También tenemos que añadir un nombre y un gmail para la identificación.

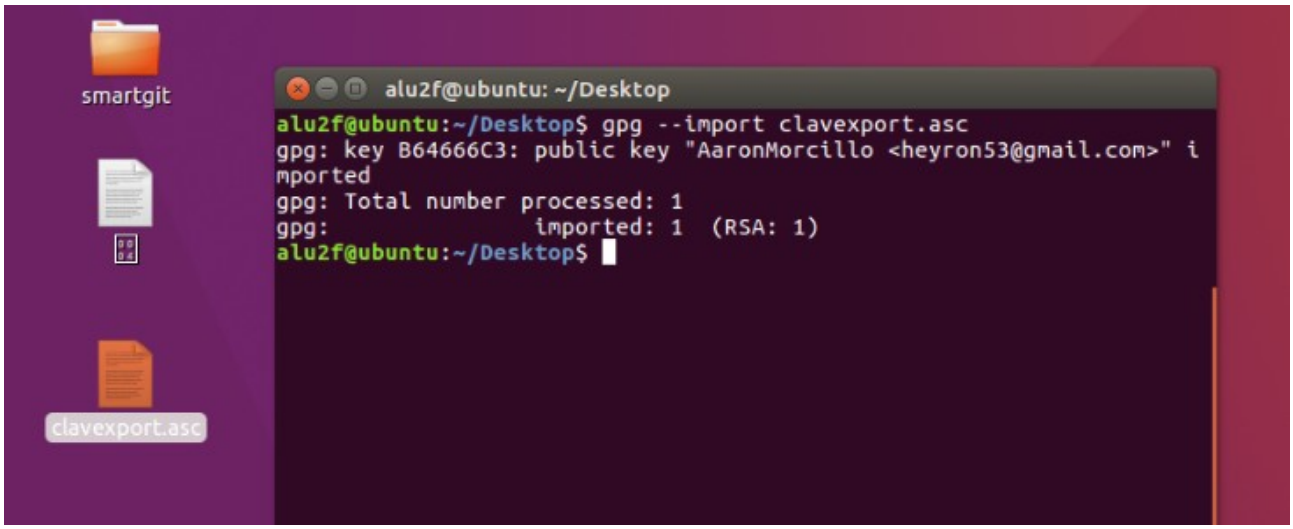
```
Virtual Machine  Help
2f@ubuntu: ~/Desktop
You selected this USER-ID:
    "Aaron <heyron53@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.

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.

Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 186 more bytes)


.+++++
.............+++++
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.

Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 82 more bytes)
...+++++

Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 55 more bytes)
....+++++
gpg: /home/alu2f/.gnupg/trustdb.gpg: trustdb created
gpg: key AB876D91 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2017-04-06
pub   2048R/AB876D91 2017-03-07 [expires: 2017-04-06]
      Key fingerprint = 3C0D BD44 410A 0E35 ED67  A350 687F 95E7 AB87 6D91
uid                  Aaron <heyron53@gmail.com>
sub   2048R/103F04A5 2017-03-07 [expires: 2017-04-06]
```

Tras completar la configuración las claves se habrán creado

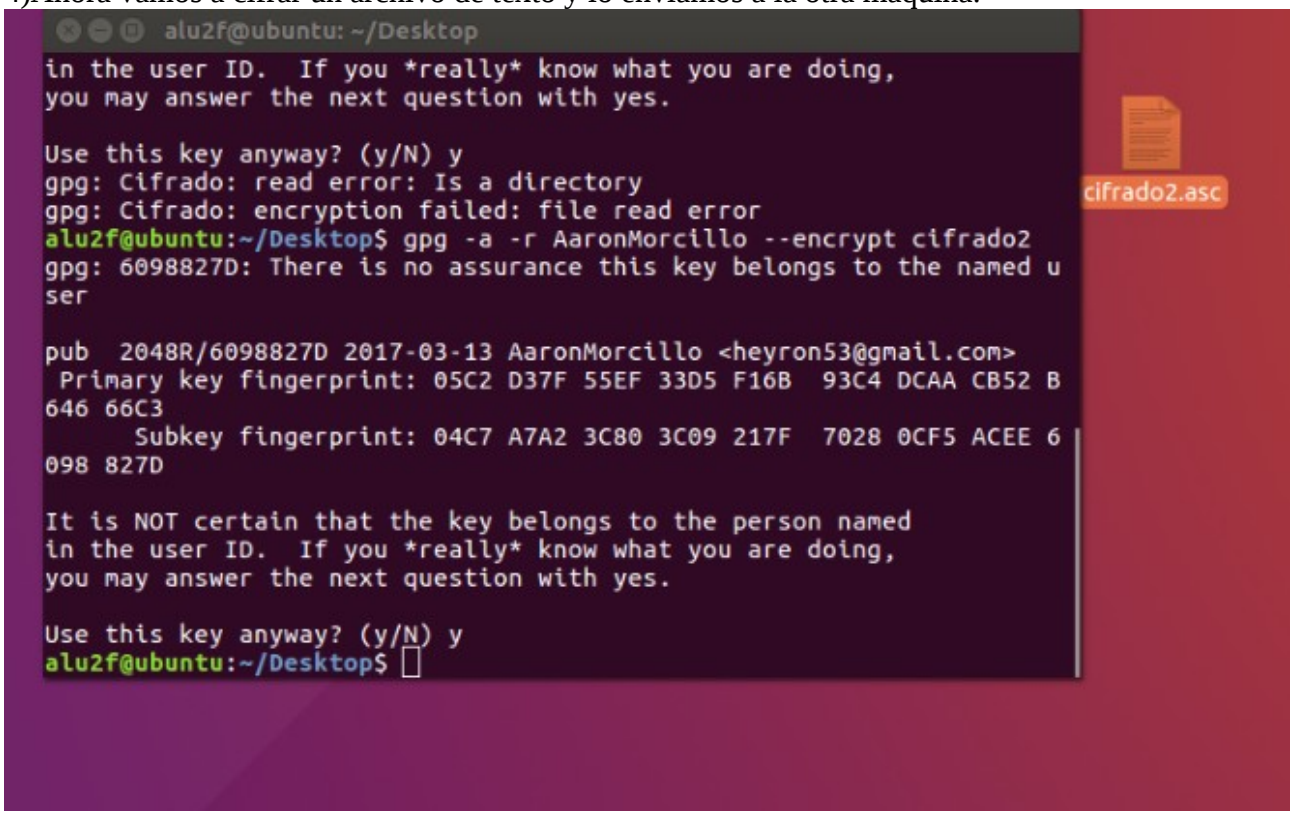
3)En este ejercicio he utilizado 2 máquinas virtuales.
Primero tenemos que exportar las claves públicas, ese archivo lo tenemos que enviar a la otra máquina.



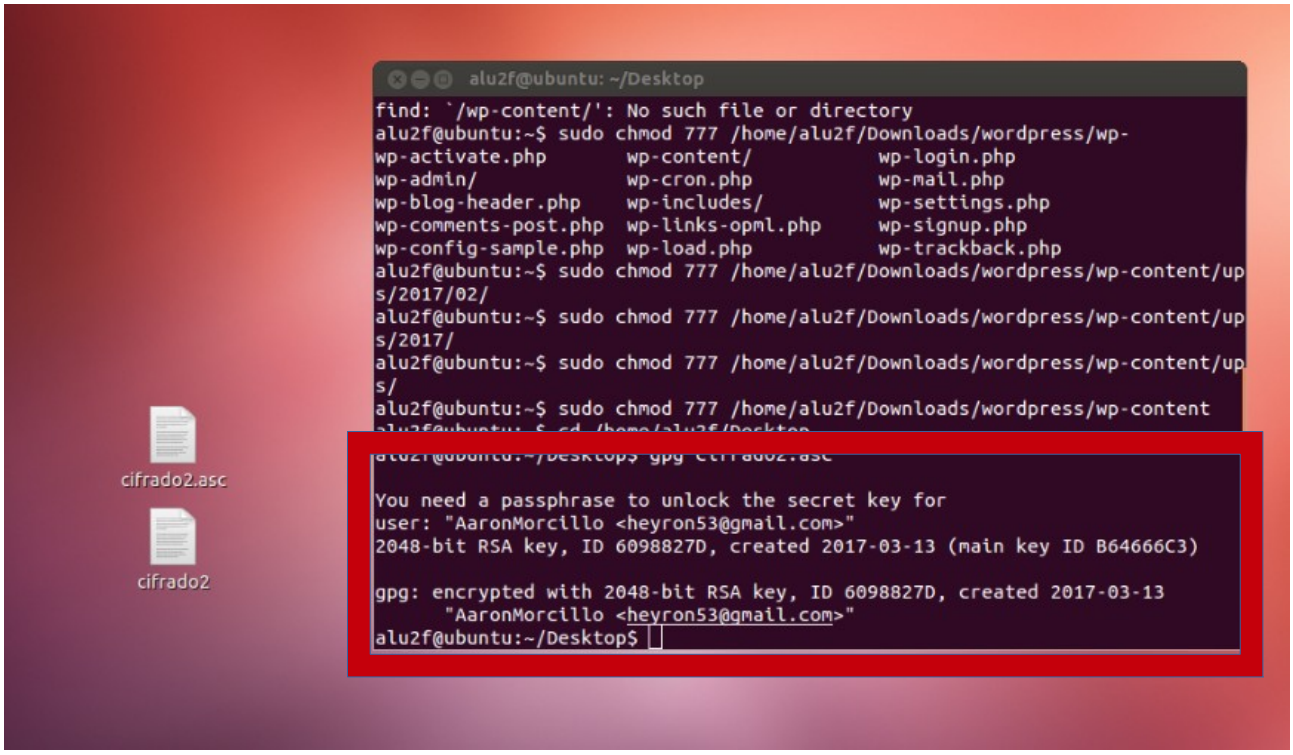
En la otra máquina importamos las claves con el siguiente comando, de esta forma las máquinas podrán enviarse archivos cifrados sin problemas.



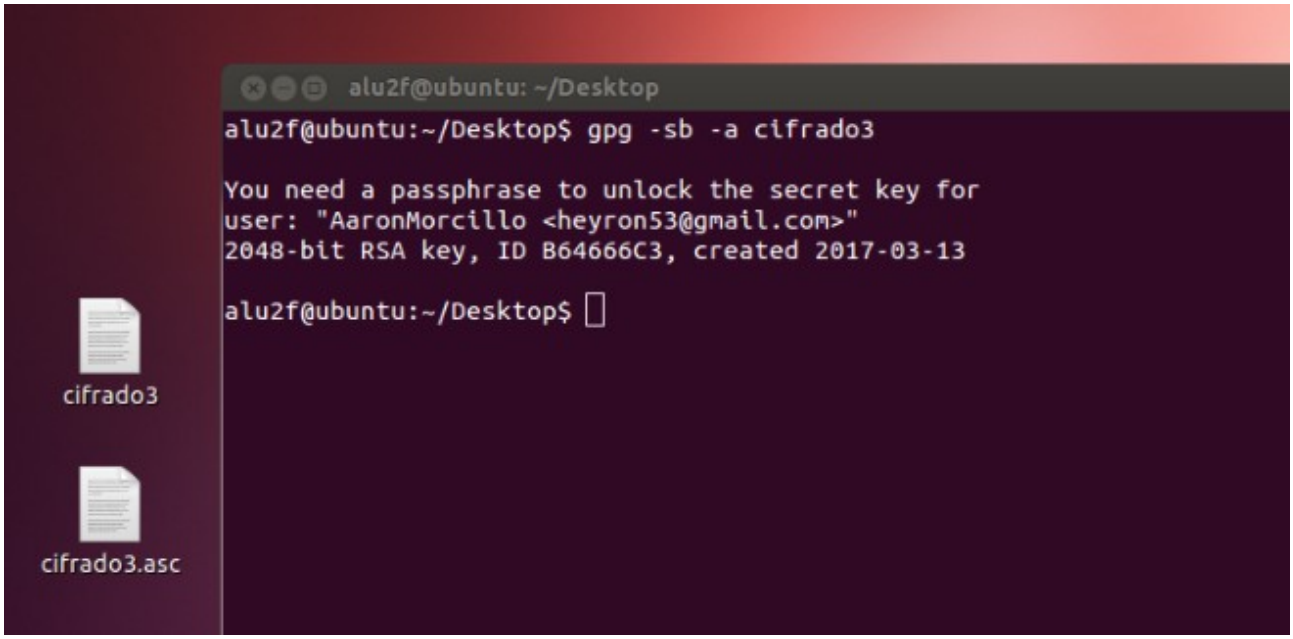
4)Ahora vamos a cifrar un archivo de texto y lo enviamos a la otra máquina.



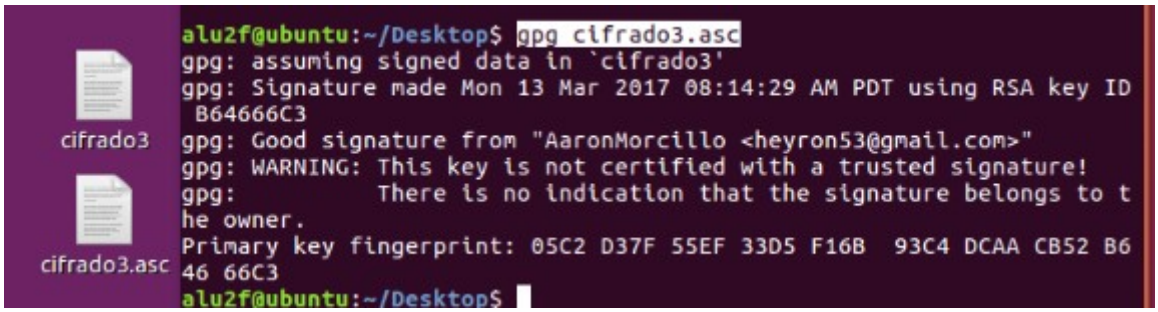
Después tenemos que descifrar el archivo que hemos recibido.



5)En el ultimo ejercicio hemos creado una firma digital con el siguiente comando, después enviamos el archivo a la otra máquina.



Despues verificamos que la firma es correcta.



En el ultimo paso tenemos que modificar el archivo y comprobar que la firma no se verifica.