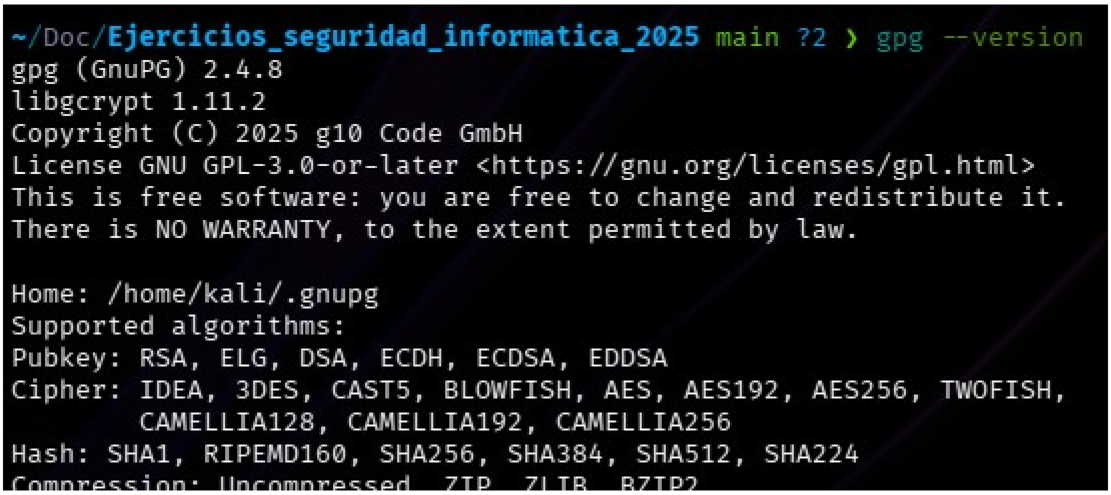
EJERCICIOS CON GPG

- **Generación y gestión de claves**

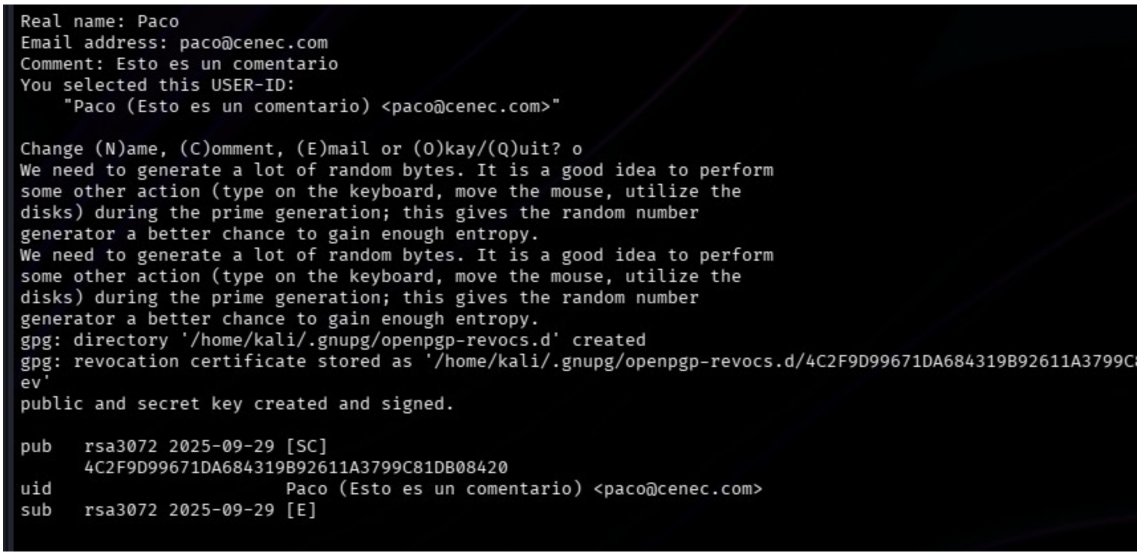
1. **Comprobar versión de GPG**
2. gpg –version



```
~/Doc/Ejercicios_seguridad_informatica_2025 main ?2 ❯ gpg --version
gpg (GnuPG) 2.4.8
libgcrypt 1.11.2
Copyright (C) 2025 g10 Code GmbH
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /home/kali/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
```

3. **Crear un par de claves RSA nuevo**
4. gpg --full-generate-key

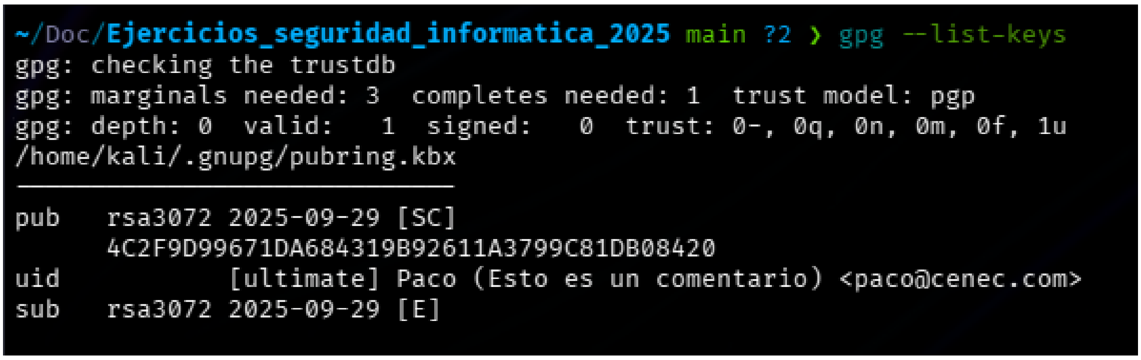
(usar nombre ficticio y correo de práctica)



```
Real name: Paco
Email address: paco@cenec.com
Comment: Esto es un comentario
You selected this USER-ID:
    "Paco (Esto es un comentario) <paco@cenec.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.
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: directory '/home/kali/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/kali/.gnupg/openpgp-revocs.d/4C2F9D99671DA684319B92611A3799C
ev'
public and secret key created and signed.

pub   rsa3072 2025-09-29 [SC]
      4C2F9D99671DA684319B92611A3799C81DB08420
uid                      Paco (Esto es un comentario) <paco@cenec.com>
sub   rsa3072 2025-09-29 [E]
```

5. **Listar claves públicas propias**

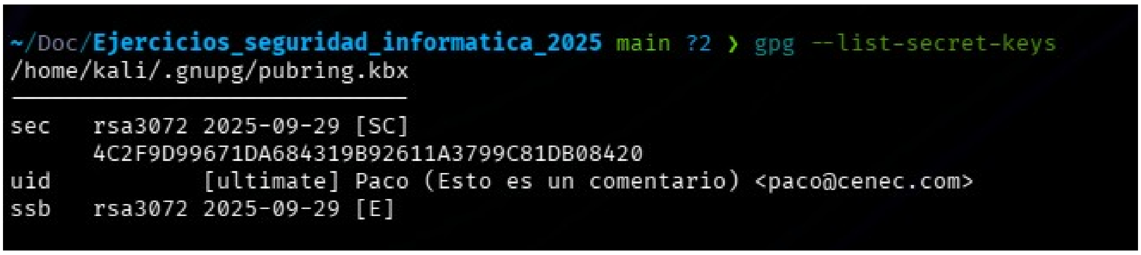
6. gpg –list-keys



```
~/Doc/Ejercicios_seguridad_informatica_2025 main ?2 ❯ gpg --list-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
/home/kali/.gnupg/pubring.kbx
-----------------------------
pub   rsa3072 2025-09-29 [SC]
      4C2F9D99671DA684319B92611A3799C81DB08420
uid           [ultimate] Paco (Esto es un comentario) <paco@cenec.com>
sub   rsa3072 2025-09-29 [E]
```

7. **Listar claves privadas (secretas)**

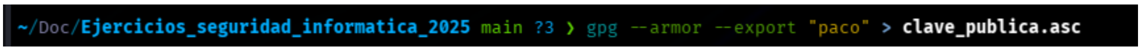
8. gpg –list-secret-keys



```
~/Doc/Ejercicios_seguridad_informatica_2025 main ?2 ❯ gpg --list-secret-keys
/home/kali/.gnupg/pubring.kbx
-----------------------------
sec   rsa3072 2025-09-29 [SC]
      4C2F9D99671DA684319B92611A3799C81DB08420
uid           [ultimate] Paco (Esto es un comentario) <paco@cenec.com>
ssb   rsa3072 2025-09-29 [E]
```

9. **Exportar clave pública a un archivo**

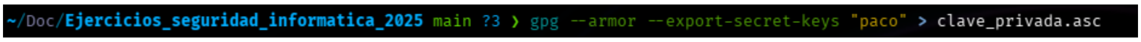
10. gpg --armor --export "NOMBRE" > clave_publica.asc



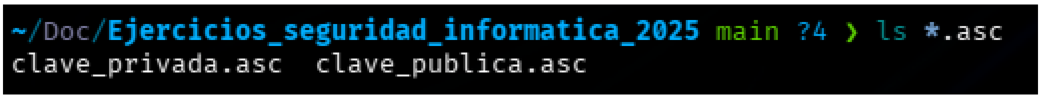
```
~/Doc/Ejercicios_seguridad_informatica_2025 main ?3 ❯ gpg --armor --export "paco" > clave_publica.asc
```

11. **Exportar clave privada (para backup)**

12. gpg --armor --export-secret-keys "NOMBRE" > clave_privada.asc



```
~/Doc/Ejercicios_seguridad_informatica_2025 main ?3 ❯ gpg --armor --export-secret-keys "paco" > clave_privada.asc
```



```
~/Doc/Ejercicios_seguridad_informatica_2025 main ?4 ❯ ls *.asc
clave_privada.asc  clave_publica.asc
```

- **Cifrado y descifrado**

7. **Crear un archivo de texto simple**

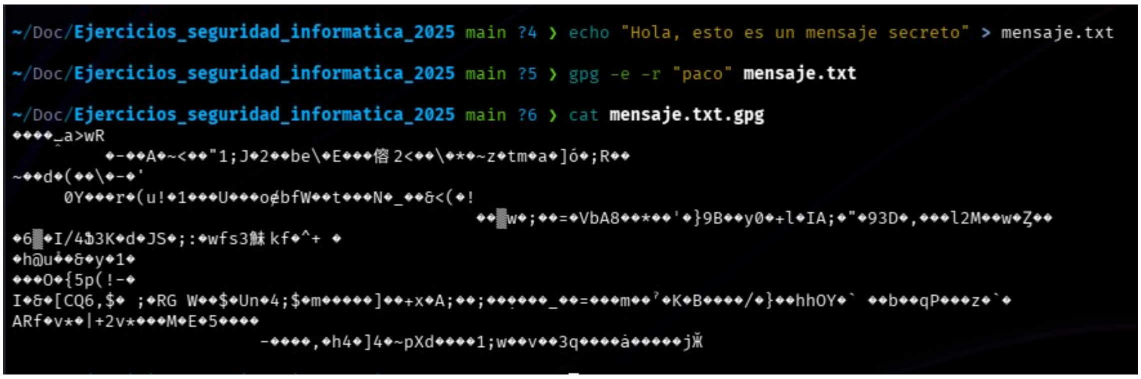
8. echo "Hola, esto es un mensaje secreto" > mensaje.txt

9. **Cifrar con la clave pública propia**

10. gpg -e -r "NOMBRE" mensaje.txt

11. **Ver el archivo cifrado (mensaje.txt.gpg)**

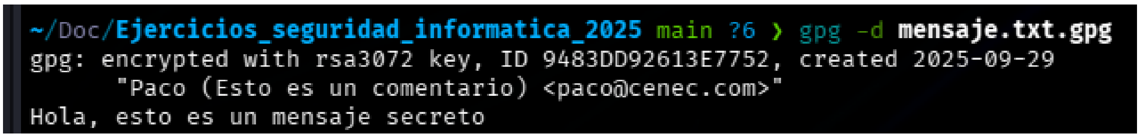
12. cat mensaje.txt.gpg



```
~/Doc/Ejercicios_seguridad_informatica_2025 main ?4 ❯ echo "Hola, esto es un mensaje secreto" > mensaje.txt

~/Doc/Ejercicios_seguridad_informatica_2025 main ?5 ❯ gpg -e -r "paco" mensaje.txt

~/Doc/Ejercicios_seguridad_informatica_2025 main ?6 ❯ cat mensaje.txt.gpg
����_a>wR
        �-��A�~<��"1;J�2��be\�E���偺2<��\�*�~z�tm�a�]ó�;R��
~��d�(��\�-�'
     0Y���r�(u!�1���U���o¢bfW��t���N�_��&<(�!
                                             ��▒w�;��=�VbA8��*��'�}9B��y0�+l�IA;�"�93D�,���l2M��w�Ȥ��
�6▒�I/4ƀ3K�d�JS�;:�wfs3鮇kf�^+ �
�h@u��&�y�1�
���O�{5p(!-�
I�&�[CQ6,$� ;�RG W��$�Un�4;$�m�����]��+x�A;��;������_��=���m��ˀ�K�B����/�}��hhOY�` ��b��qP���z�`�
ARf�v*�|+2v*���M�E�5����
                         -����,�h4�]4�~pXd����1;w��v��3q����ȧ�����j㵘
```

13. Descifrar con la clave privada

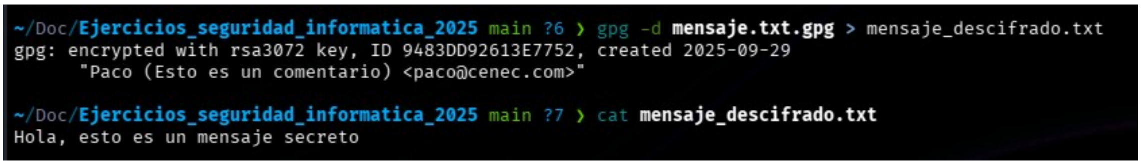
gpg -d mensaje.txt.gpg



```
~/Doc/Ejercicios_seguridad_informatica_2025 main ?6 ❯ gpg -d mensaje.txt.gpg
gpg: encrypted with rsa3072 key, ID 9483DD92613E7752, created 2025-09-29
      "Paco (Esto es un comentario) <paco@cenec.com>"
Hola, esto es un mensaje secreto
```

11. Guardar el descifrado en archivo

gpg -d mensaje.txt.gpg > mensaje_descifrado.txt



```
~/Doc/Ejercicios_seguridad_informatica_2025 main ?6 ❯ gpg -d mensaje.txt.gpg > mensaje_descifrado.txt
gpg: encrypted with rsa3072 key, ID 9483DD92613E7752, created 2025-09-29
      "Paco (Esto es un comentario) <paco@cenec.com>"

~/Doc/Ejercicios_seguridad_informatica_2025 main ?7 ❯ cat mensaje_descifrado.txt
Hola, esto es un mensaje secreto
```

- 🔹 **Firmas digitales**

12. Firmar un archivo en claro

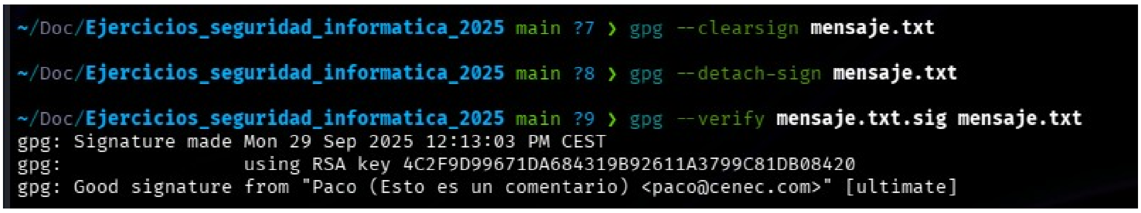
gpg --clearsign mensaje.txt

13. Generar una firma separada (.sig)

gpg --detach-sign mensaje.txt

14. Verificar una firma

gpg --verify mensaje.txt.sig mensaje.txt

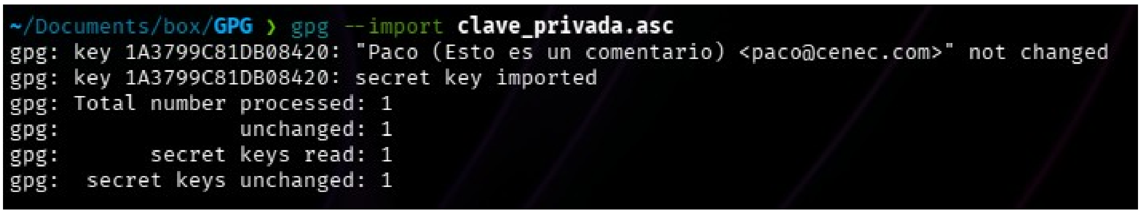


```
~/Doc/Ejercicios_seguridad_informatica_2025 main ?7 ❯ gpg --clearsign mensaje.txt

~/Doc/Ejercicios_seguridad_informatica_2025 main ?8 ❯ gpg --detach-sign mensaje.txt

~/Doc/Ejercicios_seguridad_informatica_2025 main ?9 ❯ gpg --verify mensaje.txt.sig mensaje.txt
gpg: Signature made Mon 29 Sep 2025 12:13:03 PM CEST
gpg:                using RSA key 4C2F9D99671DA684319B92611A3799C81DB08420
gpg: Good signature from "Paco (Esto es un comentario) <paco@cenec.com>" [ultimate]
```

🔹 Trabajo con otros compañeros

15. Importar la clave pública de otro alumno

gpg --import clave_compañero.asc



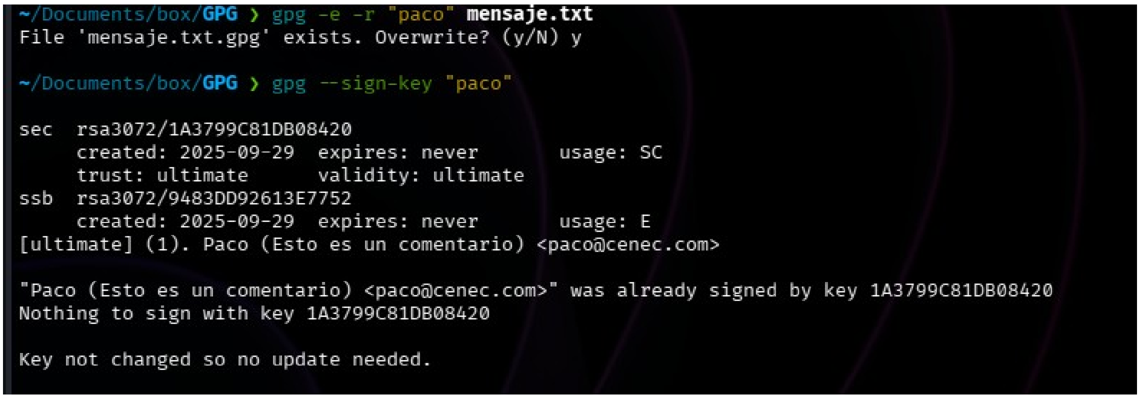
```
~/Documents/box/GPG ❯ gpg --import clave_privada.asc
gpg: key 1A3799C81DB08420: "Paco (Esto es un comentario) <paco@cenec.com>" not changed
gpg: key 1A3799C81DB08420: secret key imported
gpg: Total number processed: 1
gpg:              unchanged: 1
gpg:       secret keys read: 1
gpg:  secret keys unchanged: 1
```

16. Enviar un mensaje cifrado a su clave

gpg -e -r "NombreDelCompañero" mensaje.txt

17. Firmar la clave de un compañero

gpg --sign-key "NombreDelCompañero"

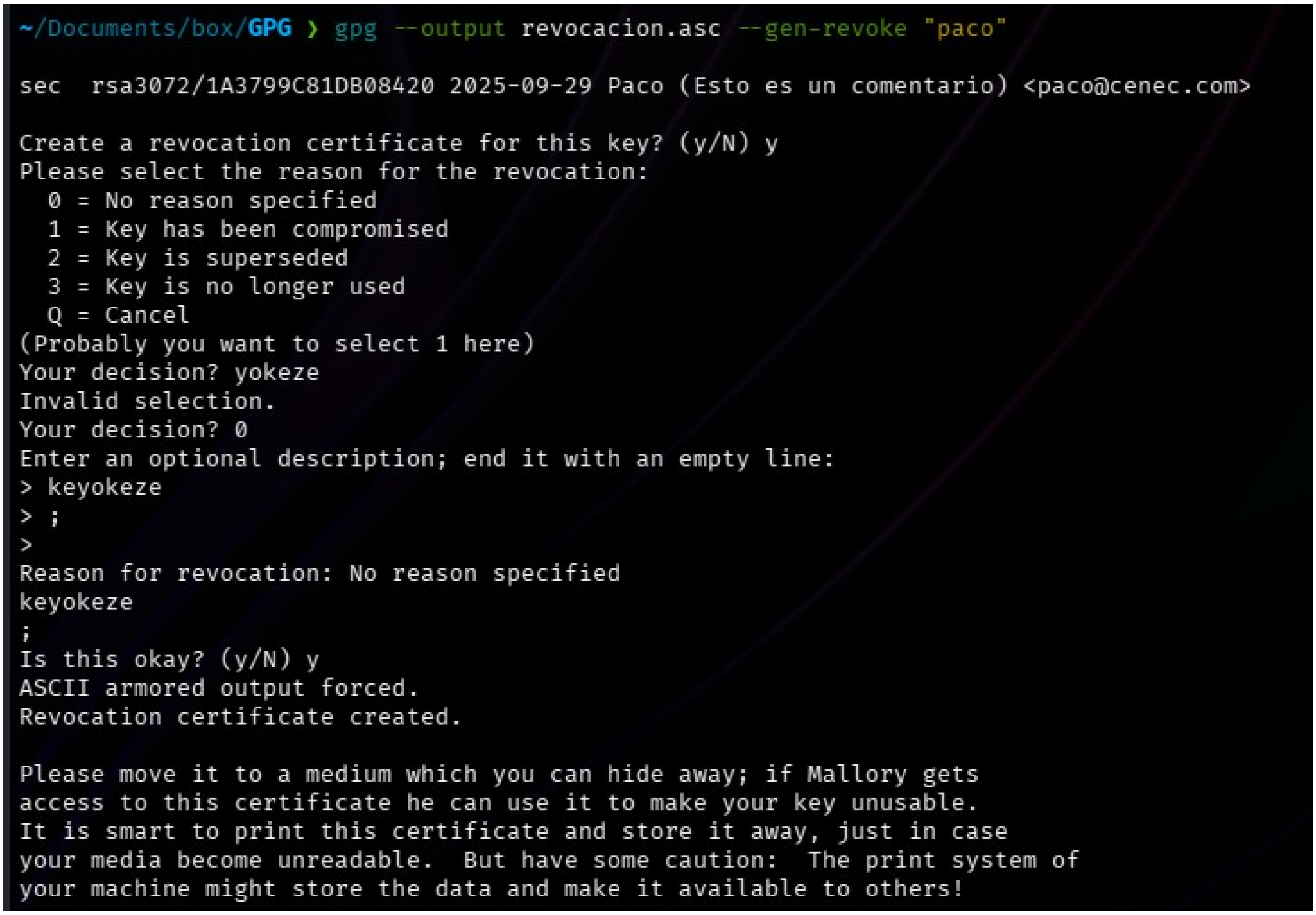


```
~/Documents/box/GPG ❯ gpg -e -r "paco" mensaje.txt
File 'mensaje.txt.gpg' exists. Overwrite? (y/N) y

~/Documents/box/GPG ❯ gpg --sign-key "paco"

sec  rsa3072/1A3799C81DB08420
     created: 2025-09-29  expires: never       usage: SC
     trust: ultimate      validity: ultimate
ssb  rsa3072/9483DD92613E7752
     created: 2025-09-29  expires: never       usage: E
[ultimate] (1). Paco (Esto es un comentario) <paco@cenec.com>

"Paco (Esto es un comentario) <paco@cenec.com>" was already signed by key 1A3799C81DB08420
Nothing to sign with key 1A3799C81DB08420

Key not changed so no update needed.
```

🔹 Gestión de claves y revocación

18. Revocar una clave (generar certificado de revocación)

gpg --output revocacion.asc --gen-revoke "NOMBRE"



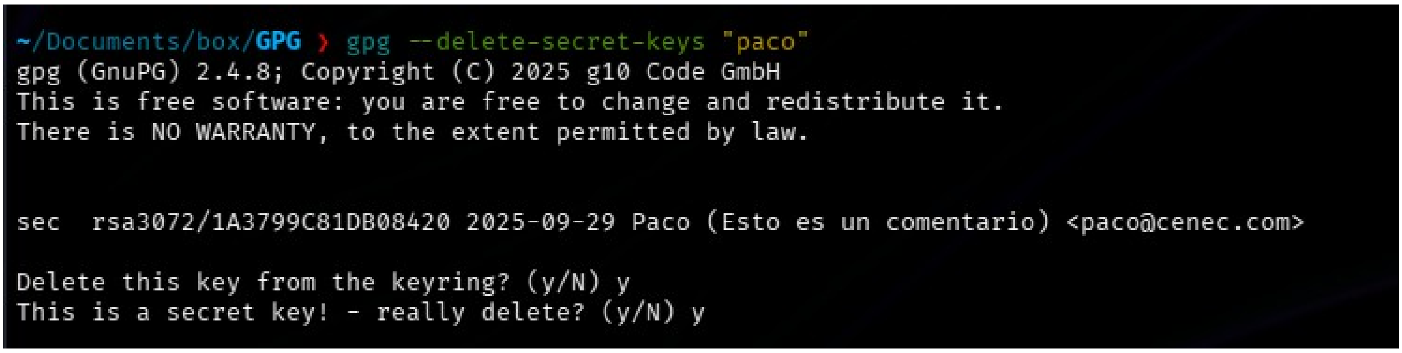
```
~/Documents/box/GPG ❯ gpg --output revocacion.asc --gen-revoke "paco"

sec  rsa3072/1A3799C81DB08420 2025-09-29 Paco (Esto es un comentario) <paco@cenec.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? yokeze
Invalid selection.
Your decision? 0
Enter an optional description; end it with an empty line:
> keyokeze
> ;
>
Reason for revocation: No reason specified
keyokeze
;
Is this okay? (y/N) y
ASCII armored output forced.
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!
```

19. Borrar una clave pública importada

gpg --delete-key "NombreDelCompañero"

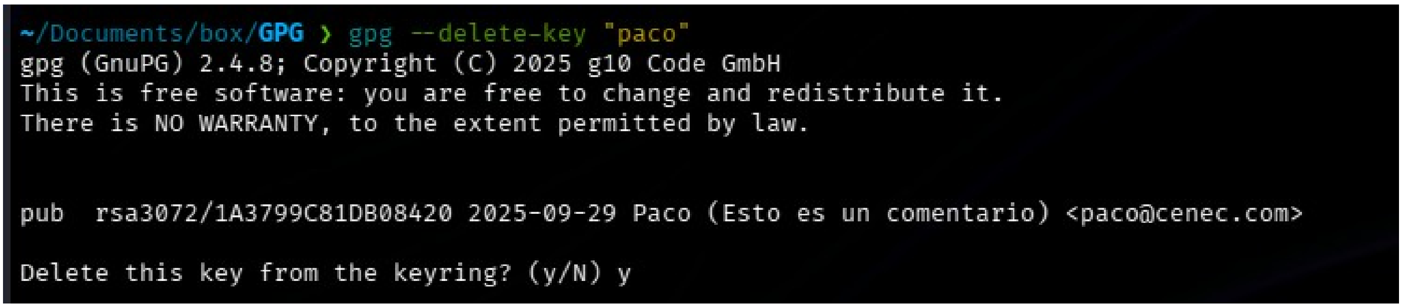
20. Borrar tu propia clave (secreta + pública)

gpg --delete-secret-keys "NOMBRE"



```
~/Documents/box/GPG ❯ gpg --delete-secret-keys "paco"
gpg (GnuPG) 2.4.8; Copyright (C) 2025 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.


sec  rsa3072/1A3799C81DB08420 2025-09-29 Paco (Esto es un comentario) <paco@cenec.com>

Delete this key from the keyring? (y/N) y
This is a secret key! - really delete? (y/N) y
```

gpg --delete-key "NOMBRE"



```
~/Documents/box/GPG ❯ gpg --delete-key "paco"
gpg (GnuPG) 2.4.8; Copyright (C) 2025 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.


pub  rsa3072/1A3799C81DB08420 2025-09-29 Paco (Esto es un comentario) <paco@cenec.com>

Delete this key from the keyring? (y/N) y
```