

## **Tema 3. Criptografía**

### **Cifrado asimétrico**

<b>Práctica</b>		<b>Tema:</b>	<u>  3  </u>	<b>Número:</b>	<u>  2  </u>
<b>Grupo:</b>	<b>Componentes:</b>				
ASIR2__					
<b>Fecha de comienzo</b>				<b>Fecha de entrega</b>	

## ÍNDICE

### Contenido

ÍNDICE .....	2
TÍTULO.....	2
OBJETIVOS .....	2
ENUNCIADO .....	2
DESARROLLO .....	3
1. Cifrado asimétrico Linux (3,2 puntos) .....	3
2. Cifrado asimétrico Windows – Gpg4Win (3,2 puntos).....	14
HERRAMIENTAS .....	18
COMENTARIOS TÉCNICOS / DIFICULTADES ENCONTRADAS .....	18
CONCLUSIÓN.....	18
BIBLIOGRAFÍA / WEBGRAFÍA.....	18

## TÍTULO

Cifrado asimétrico en Linux

## OBJETIVOS

Comprender el funcionamiento del cifrado asimétrico.

Aprender a cifrar ficheros con algoritmos asimétricos mediante la herramienta gpg.

## ENUNCIADO

La herramienta gpg permite realizar tanto criptografía simétrica como asimétrica. En esta práctica veremos la asimétrica.

Vamos a ver el proceso que hay que seguir para establecer comunicaciones cifradas de manera asimétrica. Para ello, el emisor tendrá que tener previamente la clave pública del receptor, para poder cifrar con ella el mensaje a enviar.

Subrayo en **amarillo los puntos que corresponden al emisor** del mensaje y en **verde los que corresponden receptor** del mismo.

## DESARROLLO

Para que las respuestas a las preguntas sean valoradas como correctas, deberás contestarlas con tus propias palabras (no un copia-pegar), quedando claro que entiendes lo que estás contestando.

La totalidad de la práctica está valorada sobre 4,8 puntos.

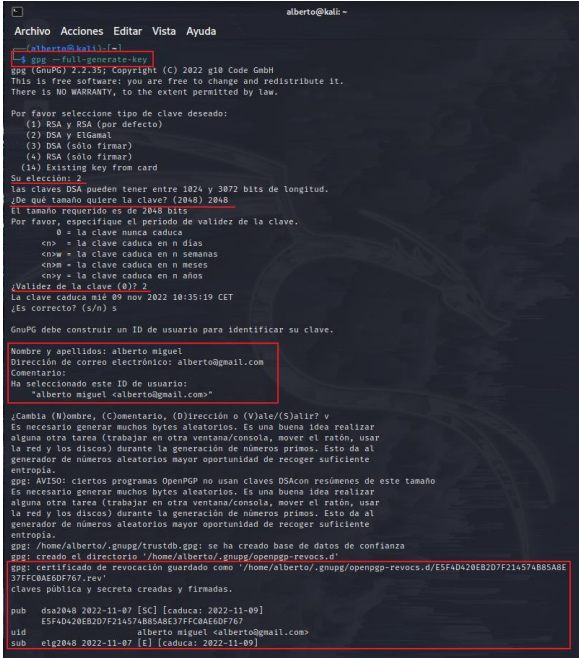
Aunque se saque más de un 2,4 en la práctica, para aprobarla es condición indispensable realizar correctamente el cifrado y descifrado tanto en Linux como en Windows

### 1. Cifrado asimétrico Linux (2,8 puntos)

<https://www.tutonics.com/2012/11/gpg-encryption-guide-part-2-asymmetric.html>

<https://www.evaristogz.com/crear-publicar-firmar-claves-gpg-gpg/>

- Explica **con tus palabras** el proceso que deben seguir, tanto el emisor como el receptor, para comunicarse entre sí protegiendo la comunicación mediante cifrado asimétrico.
- Mira las opciones del programa gpg (usa el comando “man gpg”) y responde para qué sirven los siguientes comandos y modificadores:
  - full-generate-key** → Genera un nuevo par de claves. Es una versión ampliada de --generate-key.
  - export** → Exporta todas las claves de todos los llaveros (el llavero por defecto y los registrados mediante la opción --keyring), o si se da al menos un nombre, los del nombre dado.
  - import** → Importar/fusionar llaves. Esto añade las claves dadas al llavero.

ASIR. SAD	RECEPTOR
EMISOR	Unidad 3. Criptografía
	Cifrado asimétrico
	<div data-bbox="925 177 2119 256"><b>1. Crea las claves públicas y privadas (elige como tipo de clave deseado el 2) del receptor. (0,15 puntos)</b></div> <div data-bbox="1256 264 1832 924"></div> <div data-bbox="925 935 2119 1393"><p>a) ¿En qué influirá la longitud de la clave? <b>(0,1 puntos)</b> Influye en la cantidad de bit que se utilizan para su longitud, cuantos más bits tengan más difícil será de descifrar por lo que tendrá mayor seguridad</p><p>b) ¿Por qué puede ser bueno fijar un tiempo de caducidad para la clave? <b>(0,1 puntos)</b> Porque así solo se podrá usar durante el periodo activo además tendrá mayor seguridad por el hecho de que no dará tiempo a descifrarlo y si se diese el caso de que la llegan a descifrar, caducará y no lo podrán usar ya que habrá que generar una nueva clave</p><p>c) ¿Por qué nos pide que realicemos acciones para generar los números aleatorios? <b>(0,1 puntos)</b> Para que se dificulte de mayor forma el descifrado</p></div>

**2. Ejecuta el siguiente comando:**`gpg -k`a) ¿Qué obtienes al ejecutarlo? **(0,1 puntos)****Muestra las claves almacenadas en el llavero, tanto las claves públicas como privadas**

```
(alberto@kali)-[~]
$ gpg -k
/home/alberto/.gnupg/pubring.kbx

pub   dsa2048 2022-11-07 [SC] [caduca: 2022-11-16]
       E5F4D420EB2D7F214574B85A8E37FFC0AE6DF767
uid    [ absoluta ] alberto miguel <alberto@gmail.com>
sub    elg2048 2022-11-07 [E] [caduca: 2022-11-16]
```

b) ¿Con qué otro comando puedes obtener lo mismo? Muéstralo por la pantalla. **(0,1 puntos)**`gpg --list-keys`

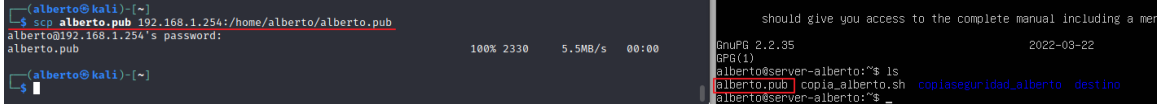
```
(alberto@kali)-[~]
$ gpg --list-keys
/home/alberto/.gnupg/pubring.kbx

pub   dsa2048 2022-11-07 [SC] [caduca: 2022-11-16]
       E5F4D420EB2D7F214574B85A8E37FFC0AE6DF767
uid    [ absoluta ] alberto miguel <alberto@gmail.com>
sub    elg2048 2022-11-07 [E] [caduca: 2022-11-16]
```

**3. Al generar las claves, se crea un directorio oculto llamado .gnupg.****Muestra su contenido.**

```
(alberto@kali)-[~]
$ ls -la .gnupg
total 32
drwx----- 4 alberto alberto 4096 nov  7 10:52 .
drwxr-xr-x 21 alberto alberto 4096 nov  7 10:52 ..
drwx----- 2 alberto alberto 4096 nov  7 10:36 openpgp-revocs.d
drwx----- 2 alberto alberto 4096 nov  7 10:36 private-keys-v1.d
-rw-r--r-- 1 alberto alberto 1883 nov  7 10:36 pubring.kbx
-rw----- 1 alberto alberto  32 nov  2 12:09 pubring.kbx~
-rw----- 1 alberto alberto  600 nov  2 12:44 random_seed
-rw----- 1 alberto alberto 1280 nov  7 10:46 trustdb.gpg
```

a) ¿En qué fichero se guardan las claves públicas y en cual las privadas? **(0,1 puntos)****pubring.kbx Almacena sus propias claves públicas**

	<p><b>private-keys-v1.d</b> es donde se almacenan las claves privadas</p> <p>b) ¿Para qué sirve el <b>openpgp-revocs.d</b>? <b>(0,1 puntos)</b>          se guarda el certificado de revocación que se creó junto con el par de claves. Los permisos de este directorio son bastante restrictivos, ya que cualquiera que tenga acceso al certificado podría revocar la clave</p> <p>c) ¿Y el <b>trustdb.gpg</b>? <b>(0,1 puntos)</b>          La base de datos de confianza</p>
<p>4. De momento, sólo hemos generado las claves del receptor. Si queremos que alguien pueda mandarnos un mensaje cifrado, ¿qué tendremos que hacer? <b>(0,15 puntos)</b></p> <p>Mandarle nuestra clave publica para que pueda cifrar con ella, se puede hacer enviándosela de propio o que la recoja de un servidor.          Si no tenemos clave publica habrá que generarla.</p>	
	<p>5. <b>Exporta tu clave pública en versión ASCII y guárdala en un fichero llamado tunombre.pub y mándasela al receptor. (0,15 puntos)</b></p> <p>Para exportarla es con el comando <b>gpg -a --export -o alberto.pub alberto miguel</b></p>  <p>Muestra el contenido de <b>tunombre.pub</b></p> <p>Para exportarla la clave publica en ASCII es <b>gpg -a --export -o alberto.pub alberto miguel</b></p>

```

(alberto@kali)-[~]
$ gpg --export -o alberto.pub alberto miquel

(alberto@kali)-[~]
$ ls
alberto.pub  copia2_alberto  copia_alberto.sh  destino  Escritorio  Música  Plantillas  Videos
cifrado      copia_alberto    Descargas          Documentos  Imágenes    original_alberto  Público

(alberto@kali)-[~]
$ cat alberto.pub
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQMubGNo0YRCAwb/rGOfqBIBACeH8cCRkk0L56R3J0T0zr9oBwknkKXlfnrD
lZE0w2jcyjyG0TtLWInJKQ0dJAMaPueNp0LSRpKvVd/QQBKjvX5NwC3bWpu
7A+o35R2Puxzpeh4bQ9CwClt/RxYVKS0Uu+320x97JiTWm8Vfomkky
F0B50jKlXG/gcjkGw/up68lVbLEVLz6GuTeGLYAfaGL07U5zH5iabKxvsj+RL
RRPu5Fd/8WQzH2KsgvTGMjL5eKlodegJBDK+HoY+MhLlyVxa655GE+Y0zLLHf
J87gztwQumy0m4R0e1luy080lNASbVLXAC/V+3d8SwCHRRfMqHmRy5tP0
2uQ0M0Bv+3LPPGQf/ahzuy+3L1220A077+35hKw0020277AaPMh53bKp
YBXmXbn13Mty20s1Gw+v/PMLr+3m20e0NNAw4M0wGwWnZCBAE3f8k+F7mrvA7+
61ESTLHswCu4n5Np/C3ZxPggeuT7UfaqmQ4SueRPYzI85gBvdDpY8mJ0A8+s
bAqCw2D9vZ5b0wng1U1V/CQXivYLElg0W2K3jK3psbskqFT42mTqpWERT2N
Q5sttUBlt4sg4c0N++P8FfrpnczT83jehfz2/M8B8c3Fb3j0mrxgubh5mwt
VAp2M0G0M3ec3zG7Rkn7U71l1o1Us+gKuF0kq0wGAg90byX1NuIcN/fJFP2X
yTH0u9C19rt8vuk/1Hw8BALZ32P/M5Qewjfm5MjXW4j0+8xgYpR8gT7VWVXF
VyzvPv6dgt/MFTmhwKwjmNv1z+H618AE3YUNjPzcy4j/AhQcmh3XFLwD
1CTEETQgMT/728P94a7pWkz9/Sh5wVhtpBcdnv6a8No3HFz14VIBngUMH1V9oR
2DhVAg0ePywZgTxqRRp25f2fe9W5EP2mUbmV6JPE0Hyus+u+Z/up941jB6R0
Lk0w5vBMD15ozv2E/PWzzfj026yNDClcVw+HXL9XmVECBypm3+1Zk+BO9HQL4F
uRQjWkLZA30yBt0w12hwgPGSvYbY69AZ31hnmvY2Pfo1068MRCAA+FiE
5FTU10stfYFFdLh3jF/WK5t92cfAmN0YVC0mFCQAcwAFcWk1BwIGFQ03CA5C
BBYCAwEChGEFCFAACgKQjJf/WK5t92zvgD/aw+19x1/SFLx0vMbx82z0v4D8nPs
e0IO9PR0W68+HIA/2ev27cvTtYf400BM+ImjPctvUSX1m0h44k4+ccuQIN
B000YVCAQbnzn1BmKsAY16400q1msh0bTRkdyPexX87fX00B3ygeK1L1b
Sj+ypg2LOFFi+A/bHvr1Suvu40M5tMlJTKGx+fCArnhgD3Cnku3JYkdWGOQ10
jwCR3jLHRfCck1YDlFkYMDKa4mc57pT41FfpuJ6PhGpMkwf5XlW0yxf00Zfm
HrIdvW0gHh81p0f0DQk82JPMU2ly7ZfE0wP9RAMZ0u0cSHj0PZAP
Q5s+0U/PduhFwB359K+mcX1HQ94b0UkY31LlWwT0gml+3DfYXZ0fMD129TbF
3u3n21mwZmPQWprBkEKS+ayGwF6HjVAAQNCACRLvLmHdcmOUYfT70mh6H/UB
j0wKefdyB8mb8Mx1JNjgezbguMk1G685oVlgY2aNaPj5M/vKwQyrf/uv5
eVfAPV2D0uK21LwF0hZLSM3m21B465gWmJ/EdJ3cpQw4KtC3/cwP85
AfPaU/rMSKcX+AKE3K6yfhLGO0y91kph00vQ+18y7ABCLdwbDw1RKjKQ0xRf+
nqVGP0053qcxgTrqRufnkcPw3ju18pV8B1gpk+DJCO1BLLPmwk2du193FKtQ
OYKUKwUZLAVJ2Qe+bmTCLQ1821bFute35R89YX2ppNECHCLchnv6p7cw1H4E
G6EACWtQTL9N06yZ/1U0uF0N/Aw322uCY3Rh1D0UJAAKAAACRCO
N//Arm32472AP9eq0YjgSNpWcgpM0uGWhdQWvd05d1y6f1bmX93B7QEARial
Ed2AFVv+nEHUM5xu/8hKq1zsYaWwK1j8d20LQw=
-----END PGP PUBLIC KEY BLOCK-----

```

6. También podemos compartir la clave subiéndola a un servidor de claves. Sube tu clave pública a un servidor de claves (usa el servidor de claves del mit, es el que menos problemas me ha dado). (0,2 puntos)

Como identificador de la clave, usa los números de después de la “/”

```

pub 2048D/C89F0B11 2022-11-06
uid receptor_sad pabloserrano (receptor sad) <receptor_sad@gmail
l.com>
sub 2048g/5AF64C23 2022-11-06

```

[https://wiki.archlinux.org/index.php/GnuPG\\_\(Español\)](https://wiki.archlinux.org/index.php/GnuPG_(Español))

<https://davidinformatico.com/cifrar-con-gpg/>

Para ello utilizamos el comando `gpg --keyserver pgp.mit.edu --send-keys` (copiamos todo el código de la clave publica y lo pegamos) como se muestra en la imagen

```
(alberto@kali)-[~]
$ gpg --list-keys
/home/alberto/.gnupg/pubring.kbx

pub   dsa2048 2022-11-07 [SC] [caduca: 2022-11-09]
       E5F4D420EB2D7F214574B85A8E37FFC0AE6DF767
uid     [ absoluta ] alberto miguel <alberto@gmail.com>
sub     elg2048 2022-11-07 [E] [caduca: 2022-11-09]

(alberto@kali)-[~]
$ gpg --keyserver pgp.mit.edu --send-keys E5F4D420EB2D7F214574B85A8E37FFC0AE6DF767
gpg: enviando clave 8E37FFC0AE6DF767 a hkp://pgp.mit.edu
```

Una vez subida la clave, entra en la web del servidor ([hkp://pgp.mit.edu](http://pgp.mit.edu)) y haz una búsqueda, bien poniendo el mail puesto al generar la clave, el user id o bien poniendo la clave.

**(0,1 puntos)**

Para comprobar en terminal es `gpg --keyserver pgp.mit.edu --search-keys`

```
(alberto@kali)-[~]
$ gpg --keyserver pgp.mit.edu --search-keys E5F4D420EB2D7F214574B85A8E37FFC0AE6DF767
gpg: data source: http://pgp.mit.edu:11371
(1)  alberto miguel <alberto@gmail.com>
     2048 bit DSA key 8E37FFC0AE6DF767, creado: 2022-11-07, caduca: 2022-11-09
Keys 1-1 of 1 for "E5F4D420EB2D7F214574B85A8E37FFC0AE6DF767". Introduzca número(s), 0)tro, o F)in > f
```

[pgp.mit.edu/pks/lookup?search=alberto+miguel&op=index](http://pgp.mit.edu/pks/lookup?search=alberto+miguel&op=index)

### Search results for 'miguel alberto'

Type	bits/keyID	Date	User ID
------	------------	------	---------

pub	2048D/ <a href="#">AE6DF767</a>	2022-11-07	<a href="#">alberto miguel &lt;alberto@gmail.com&gt;</a>
-----	---------------------------------	------------	--

7. Observa este pantallazo obtenido del servidor de claves

```
pub   3072R/48359505 2019-01-31 *** KEY REVOKED *** |
       receptor <receptor@s:
```

¿Qué significa “key revoked” y para qué se utiliza? **(0,1 puntos)**

Significa dejar sin valor la clave deseada, al dejarla sin valor esa clave ya no serviría, se usa antes de que caduque la clave en algunos casos como puede ser que la contraseña haya sido comprometida, la clave se quiere cambiar, la clave ya no se usa, entre otros motivos



8. Obtén desde el servidor de claves la clave pública del receptor para cifrar lo que le vayas a enviar  
(0,15 puntos)

En caso de error a la hora de importar la clave, probaríamos lo siguiente:

- Delante de la dirección del servidor, añade el protocolo hkp, es decir:  
[...] hkp://pgp.mit.edu [...]
- Si sigue sin funcionar, delante de la clave pública pon 0x (para que lo coja en hexadecimal):

... 0xE4829 ...

```
alberto@server-alberto:~$ gpg --keyserver hkp://pgp.mit.edu --search-keys 0xE5F4D420EB2D7F214574B85A8E37FFC0AE6DF767
gpg: data source: https://pgp.mit.edu:443
(1) alberto miguel <alberto@gmail.com>
      2048 bit DSA key 8E37FFC0AE6DF767, created: 2022-11-07, expires: 2022-11-09 (expired)
Keys 1-1 of 1 for "0xE5F4D420EB2D7F214574B85A8E37FFC0AE6DF767". Enter number(s), N)ext, or Q)uit >
alberto@server-alberto:~$ gpg --keyserver hkp://pgp.mit.edu --recv-keys 0xE5F4D420EB2D7F214574B85A8E37FFC0AE6DF767
gpg: key 8E37FFC0AE6DF767: public key "alberto miguel <alberto@gmail.com>" imported
gpg: Total number processed: 1
gpg:      imported: 1
alberto@server-alberto:~$
```

Si no pudiéramos obtener la clave desde el servidor, nos la podrían mandar por mail, copia remota, etc. Prueba a importar en caso de que nos la hubieran enviado por alguno de los mecanismos anteriores (como en el punto 5). (0,1 puntos)

Para importarla seria usando el comando `gpg --import alberto.pub`

9. Muestra el listado de claves del llavero para verificar que se ha importado bien.

```
alberto@server-alberto:~$ gpg --list-key
/home/alberto/.gnupg/pubring.kbx
-----
pub   dsa2048 2022-11-07 [SC] [expires: 2022-11-16]
      E5F4D420EB2D7F214574B85A8E37FFC0AE60F767
uid    [ unknown] alberto miguel <alberto@gmail.com>
sub    elg2048 2022-11-07 [E] [expires: 2022-11-16]
```

10. Crea el mensaje que enviarás al receptor y cífralo.  
Guarda el mensaje cifrado en un fichero ASCII llamado tunombre\_cifrado.  
(0,15 puntos)

```
alberto@server-alberto:~$ gpg -v -a -o alberto_cifrado --encrypt --recipient alberto@gmail.com msj_a_cifrado
gpg: using gpg trust model
gpg: using subkey 674B2EBA80003A40 instead of primary key 8E37FFC0AE60F767
gpg: automatically retrieved 'alberto@gmail.com' via local
gpg: 674B2EBA80003A40: There is no assurance this key belongs to the named user

Subkey fingerprint: E5F4 D420 EB2D 7F21 4574  B85A 8E37 FFC0 AE60 F767
Primary key fingerprint: E5F4 D420 EB2D 7F21 4574  B85A 8E37 FFC0 AE60 F767
Subkey fingerprint: DE99 C0EF BAA5 689C 7E32  65D5 674B 2EBA 8000 3A40

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 'msj_a_cifrado'
File 'alberto_cifrado' exists. Overwrite? (y/N) y
gpg: writing to 'alberto_cifrado'
gpg: ELG/RES256 encrypted for: '674B2EBA80003A40 alberto miguel <alberto@gmail.com>'
alberto@server-alberto:~$ scp alberto_cifrado 192.168.1.19:/home/alberto/alberto_cifrado
alberto@192.168.1.19's password:
alberto_cifrado                                100% 919    1.7MB/s   00:00
alberto@server-alberto:~$
```

```
alberto@server-alberto:~$ cat alberto_cifrado
-----BEGIN PGP MESSAGE-----

hQ10A2dLLnqN3TPAEAgA130k17zn1Q06/YBNRPuIMJad08QY9TpB2oJLfokqbuJB
Y8Kc0kNNTKMomTroPfr7smkn11c2McRt3kaMUKF05Hp2d0DRuvv+ha8C/3/7+
cd9n2/s4y81FUARDJYfsIn/Ufvg1mM9qaQTeFpaL1gzLLbXpGgKvuuNynTUGqq2Y
Iout24EfssqHAszU/9r+tnkxgaRtI4HY8BK2cw9B615A5VnV7toyu03sfdvx8lm1
G310BX1F90/V0c2avSussEEnuhoVRL0KMHXUu000+wfndu0x00JH457Hw/f1
ub2Tr4HEaPul7hNT+R25pmNRXBIdz1nm3JW+11SrhQf/MVJB560Cuo5JGENG39E8
qfHgVnoG7M3KMQ0VNV6p3tuwRk/oh101dncf5nt3RmkoxYKvJTuYhBMJ+u52g2
0TasoozVBNpgeXMBL12VgkvVv+VMD+7Q1PyAUUvJtSDJbLp2VBTEHvdeADPK+Hn
JHq/4SueoMdsqzeehJg4TOYSSRePRFXVJR0UD306/63C10KJKRts3JMKZ08DMQN
yduVKGCEP1Fms+KkXaKvUjSLOW7NtAlp50ZuPK/4LGN5r3DFV5Gvns/MxTbVxuc
aFpF9aInCYk11SNp+Yf0gHqXmudYg+Hojen7SHUAZ77ISaEKYfExuBec56
p5Tma21K5oV161NDP1NAX1EebHxNa/QoufB8Aa/a8nx8tc2UC+/J/S1ToRc3Vq
YBD1v4M8YIab3umcf5d40p117SiHhQ78GsWvX5JfSoMf05Y+I1br1aFFs+52UTKT
nLTmVvDuxUoJ
E22Wj
-----END PGP MESSAGE-----
alberto@server-alberto:~$
```

a) ¿Con qué clave se está cifrando el archivo? (0,1 puntos)  
Con la clave pública del receptor

b) ¿Qué significa cada parámetro? **(0,1 puntos)**

-v → modo verboso

-a → crearlo en modo ASCII

-o el fichero que se va a escribir

--encrypt --recipient → encriptar y decir con que clave publica se hace

c) ¿Por qué avisa de que la clave igual no pertenece a la persona que se nombra en el identificador de usuario?  
**(0,1 puntos)**

Por seguridad ya que no tiene manera de saber si es de verdad quien dice ser

11. Algún atacante, podría hacerse pasar por el receptor y pasarte su clave pública en vez de la del receptor original, con lo que cifrarías el mensaje con ella y luego podría descifrarlo con su clave privada.

Para asegurarnos de que eso no ha sucedido, teclea el siguiente comando:

*gpg --fingerprint*



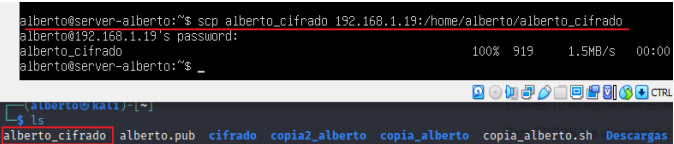
```

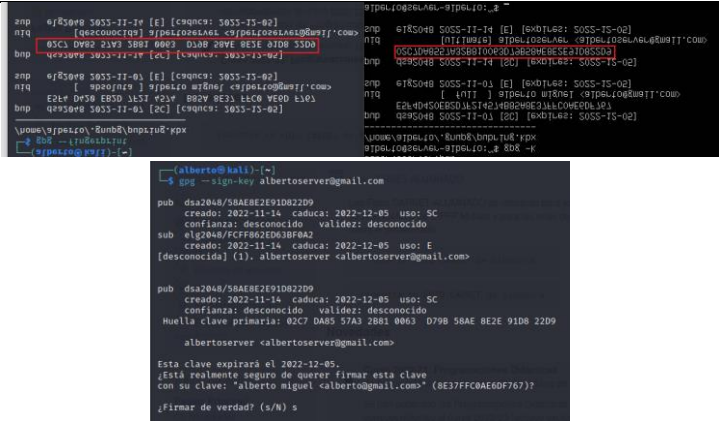
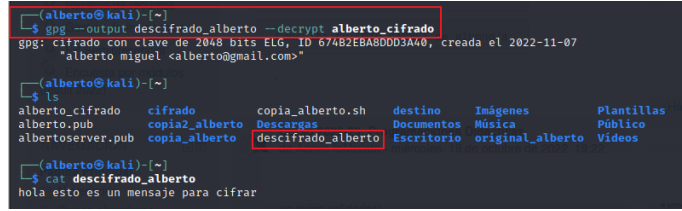
(alberto@kali)~$ gpg --list-keys
/home/alberto/.gnupg/pubring.kbx
pub dsa2048 2022-11-07 [SC] [caduca: 2022-11-16]
     E5F4D420EB2D7F214574B85A8E37FFC0AE6D767
uid [ absoluta ] alberto miguel <alberto@gmail.com>
sub e1g2048 2022-11-07 [E] [caduca: 2022-11-16]

(alberto@kali)~$ gpg --fingerprint
-----BEGIN PGP MESSAGE-----
alberto@server-alberto:~$ gpg --fingerprint
/home/alberto/.gnupg/pubring.kbx
-----BEGIN PGP MESSAGE-----
pub dsa2048 2022-11-07 [SC] [expires: 2022-11-16]
     E5F4 D420 EB2D 7F21 4574 B85A 8E37 FFC0 AE6D F767
uid [ unknown] alberto miguel <alberto@gmail.com>
sub e1g2048 2022-11-07 [E] [expires: 2022-11-16]
alberto@server-alberto:~$
  
```

a) ¿Coincide con la huella del que ha cifrado el mensaje?  
(fíjate en el pantallazo del punto 2)

Si coinciden

<p>b) ¿si no coincide qué significa? <b>(0,1 puntos)</b></p> <p><b>Que algún atacante ha interferido en la conexión haciéndose pasar por el receptor</b></p>	
<p>12. Ahora que ya está todo en orden, el emisor enviaría el mensaje al receptor por mail o cualquier otro medio. Para pasar el mensaje del emisor al receptor vamos a usar el comando scp (security copy). La sintaxis es muy sencilla (prácticamente como el rsync):</p> <pre>scp [other options] [source username@IP]:/[directory and file name] [destination username@IP]:/[destination directory]</pre> <p>(El usuario y la ip sólo serán necesarios cuando el usuario no sea en con el que estamos trabajando y la ip no sea la de nuestro equipo). <b>(0,1 puntos)</b></p>  <p>The screenshot shows a terminal window with the following commands and output:</p> <pre>alberto@server-alberto:~\$ scp alberto_cifrado 192.168.1.19:/home/alberto/alberto_cifrado alberto@192.168.1.19's password: alberto_cifrado alberto@server-alberto:~\$</pre> <p>Below the terminal output, there is a file explorer view showing the contents of the 'alberto_cifrado' directory:</p> <pre>alberto_cifrado alberto.pub cifrado copia2_alberto copia_alberto copia_alberto.sh Descargas</pre>	
	<p>13. <b>Asegúrate de que, efectivamente, la clave coincide con la pública:</b></p> <pre>gpg --sign-key [ID de a clave pública emisor]</pre> <p>Para ello lo primero que debemos hacer es compartirle al receptor la clave pública del emisor, (se puede hacer que el receptor la coja del servidor o compartírsela directamente, con esta última le tenemos que importar la clave), a continuación, usamos el comando <code>gpg --fingerprint</code> en el receptor y el comando <code>gpg -k</code> en el emisor para comprobar que no se ha modificado. Una vez realizado lo anterior y ver que no se ha modificado ejecutamos <code>gpg --sign-key albertoserver@gmail</code> en el receptor.</p>

	
	<p>14. Ahora que vemos que todo está en orden, descifra el mensaje enviado por el emisor guardándolo en un fichero llamado descifrado_tunombre (0,25 puntos)</p> 
	<p>15. Haz una copia del mensaje cifrado.</p> <p>Ábrelo con un editor y modifica algún carácter del mensaje cifrado. Guárdalo y descifralo.</p>  <p>¿Qué ocurre? ¿Qué se está garantizando?</p> <p>Ocurre un error, con ello se aseguran de que una vez cifrado el mensaje no se pueda modificar.</p>

Tendrás que realizar la práctica con tu compañero y, los dos, seréis emisores y receptores.

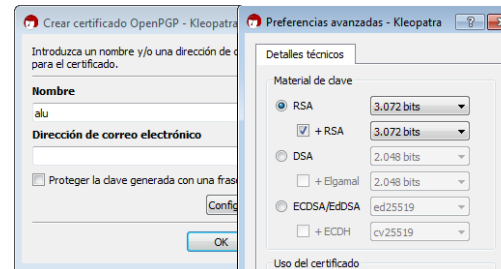
## 2. Cifrado asimétrico Windows – Gpg4Win (2 puntos)

Descárgate e instala el programa gratuito Gpg4Win.

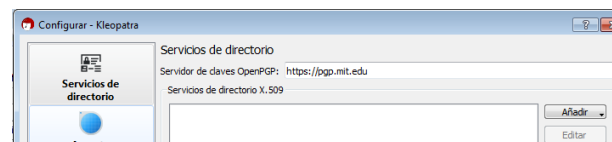
<https://programarivm.com/cifrar-archivos-en-windows-con-gpg4win-y-kleopatra>

- a) Sabiendo que, el servidor que he usado para almacenar mi clave pública está en pgp.mit.edu, y que el user id que he usado es el quique211104, envíame un archivo en el que ponga tu nombre cifrándolo de manera asimétrica con el programa Kleopatra. Explica los pasos que has dado apoyándote con pantallazos. (1 puntos)

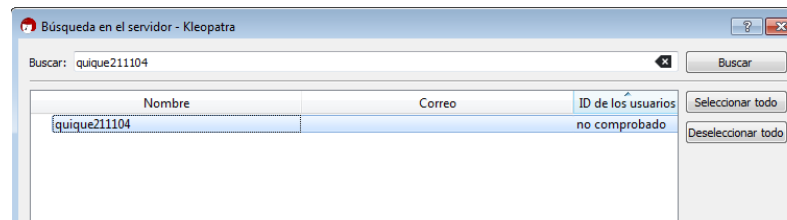
**Lo primero que debemos hacer una vez instalado el programa es crearnos unas claves**



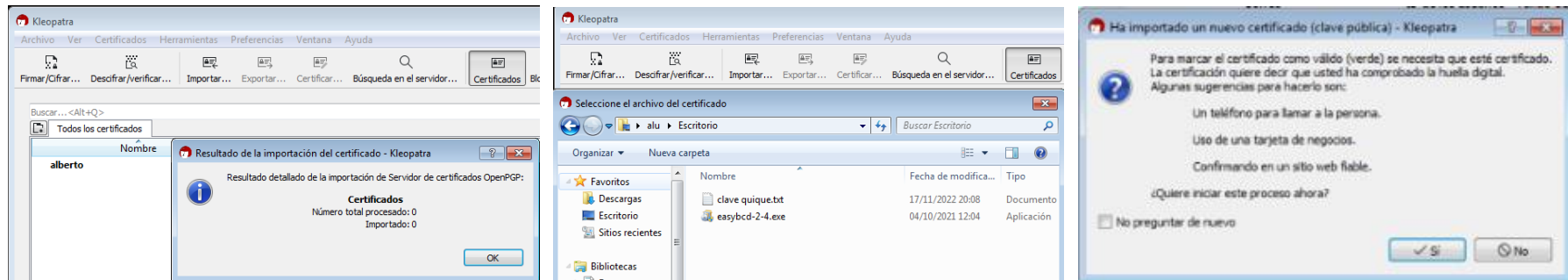
**A continuación, nos dirigimos a preferencias y a configurar kleopatra y en servicios de directorio tendremos que modificar el servidor de claves por el que deseamos.**



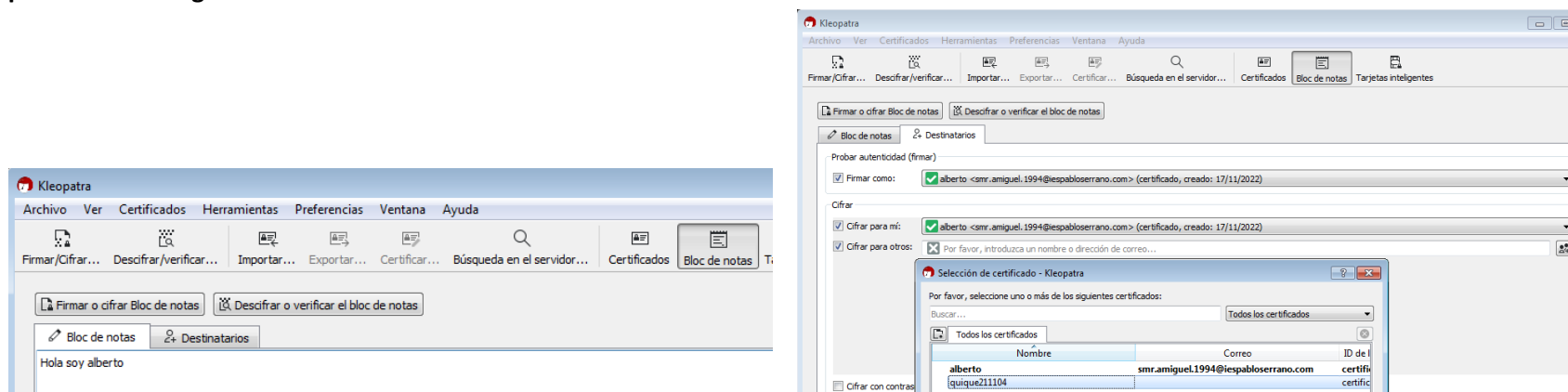
**Una vez realizado lo anterior nos dirigimos a búsqueda en el servidor y buscamos el userId quique211104, una vez encontrada le damos a importar**



En mi caso no ha sucedido nada ya que en el resultado me muestra que ha importado 0 elementos por lo que decidí entrar al buscador de preferencia, entrar en la pagina del servidor de claves <https://pgp.mit.edu> buscar el userID y coger de ahí la clave pública pasándolo a un archivo e importar ese archivo, la primera vez que se haga con ese certificado nos pedirá que lo certifiquemos de alguna manera para poderla usar



A continuación, nos dirigiremos a bloc de notas en el cual escribiremos el mensaje que queremos cifrar, después iremos a la pestaña de Destinatarios para poder elegir con que cifrar, en este caso la cifraremos tanto con mi clave como con la clave publica de quique211104 para que ambos la podamos descifrar, después haremos clic en “Firmar o cifrar Bloc de notas”, y el mensaje que nos aparezca lo copiaremos en un documento para así poderlo enviar por un medio seguro.



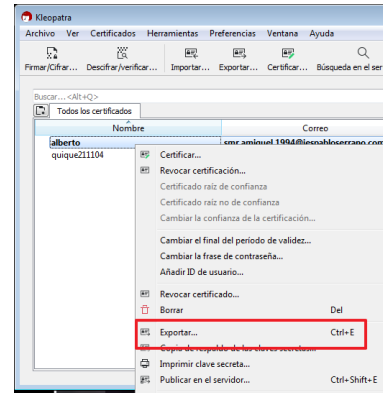
- b) Haz lo necesario para que yo pueda enviarte un mensaje cifrado de manera asimétrica. Explicame qué es lo que has hecho, apoyándote con pantallazos, y qué es lo que tengo que hacer para enviarte dicho mensaje y que tú lo puedas descifrar.

No puedes usar un servidor de claves, tendrás que hacerlo de otra manera.

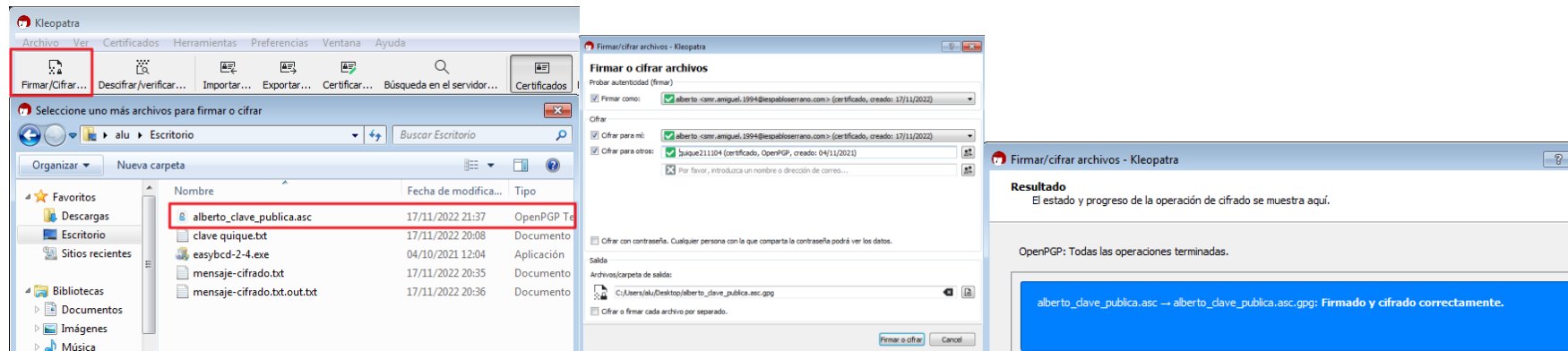
**(1 puntos)**

Como no se puede hacer uso del servidor de claves, lo que se me ocurre es exportar mi clave pública, cifrarla con la clave pública de quique211104 y enviarla por un medio seguro.

Para ello lo que debemos hacer es clic derecho sobre mi clave la cual nos dará la opción de exportar.



Ahora lo que vamos a hacer es clic en la opción /Firmar/Cifrar... para poder cifrar el documento que tiene nuestra clave, en la cual tendremos que cifrar con las 2 claves. Una vez cifrado lo mandamos por un medio seguro





**NOTA:**

Para trabajar con mi clave pública en el Gpg4Win (Kleopatra) podéis hacerlo de varias maneras:

1. Desde el propio Kleopatra, importando la clave. Para eso tenéis que configurar Kleopatra para que vaya a buscar la clave al servidor de claves del mit (<https://pgp.mit.edu>), que es donde la subí.
2. Entrando en el servidor del mit vía web, acceder a la clave y copiando y pegando el contenido en un archivo y luego importarla en Kleopatra.

La manera idónea de hacerlo es la 1, pues así toda la gestión de claves la hacéis desde el propio programa, pero podéis emplear cualquiera de las 2.

**HERRAMIENTAS**

(si no está correctamente -0,2 puntos)

**COMENTARIOS TÉCNICOS / DIFICULTADES ENCONTRADAS**

(si no está correctamente -0,2 puntos)

**CONCLUSIÓN**

(si no está correctamente -0,2 puntos)

**BIBLIOGRAFÍA / WEBGRAFÍA**

(si no está correctamente -0,2 puntos)