

# Práctica de GnuPG

Traducción al castellano de la [Práctica sobre GnuPG](#), vista en [Bulma](#) escrita por Leopoldo Palomo para sus alumnos universitarios para que conozcan el GnuPg, y basada en el artículo de [Introducción al GnuPg](#) de [René Merou](#).

Práctica GnuPG

1. **Enunciado de la práctica**
2. **Generar la clave**
3. **Generar el certificado de revocación**
4. **Enviar y recibir claves al servidor de claves**
5. **¿Y ahora, qué?**
6. **Fuentes de información**

## 1. Enunciado de la práctica

La práctica consiste en enviar un mensaje criptografiado (cifrado) y firmado. Para poder realizarla, primero hay que generar un par de claves (privada y pública), configurar el gestor de correo. Por último, hay que enviar un correo al profesor, firmado y cifrado. Para hacerlo tendreis que comprobar las respectivas claves públicas e identidades.

## 2. Generar la clave

En el servidor hay instalado el programa GnuPG. Desde un terminal seguid los siguientes pasos:

```
leo@ranganathan:~$ gpg --gen-key
```

```
gpg (GnuPG) 1.4.3; Copyright (C) 2006 Free Software Foundation, Inc.
```

```
This program comes with ABSOLUTELY NO WARRANTY.
```

```
This is free software, and you are welcome to redistribute it under certain conditions.  
See the file COPYING for details.
```

Entonces nos hará una serie de preguntas que tendremos que responder:

- ¿Tipo de clave? seleccionar la opción predeterminada: 1

Seleccionar que tipo de clave quereis:

(1) DSA and Elgamal (default)

(2) DSA (sólo firmar)

(5) RSA (sólo firmar)

¿Vuestra selección?

- ¿Medida de la clave? Aceptar el valor por defecto: 2048

DSA keypair will have 1024 bits.

ELG-E keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048)

La medida solicitada es 2048 bits

- ¿Tiempo de validez? Introducir 1m, que quiere decir 1 mes, y ratificarlo.

Especificad el tiempo de validez de la clave.

0 = la clave no caduca

< n > = la clave caduca a los n días

< n >w = la clave caduca a las n semanas

< n >m = la clave caduca a los n meses

< n >y = la clave caduca a los n años

Indicad la validez de la clave (0) 1m

Key expires at dl 19 jun 2006 12:26:54 CEST

Is this correct? (y/N) y

- ¿Nombre y apellidos? Pon tus datos:

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) ”

Nombre y apellidos: Shiyali Ramamrita Ranganathan

- ¿Dirección electrónica? Poned la de ranganathan o alguna vuestra:

Dirección electrónica: shiyalirr@ranganathan.ub.es

- ¿Comentario? Poned algún comentario que os identifique esa dirección o lo que creais conveniente

Comentario: Dirección Documentación

- A partir de ahí sólo falta seguir la confirmación de todos los pasos. Para terminar pulsar la letra ‘o’

Está usando el juego de caracteres `utf-8`.

Ha elegido el identificador de usuario:

“Shiyali Ramamrita Ranganathan (Dirección Documentación) ”

Cambiar (N)ombre, (C)omentario, (E)mail o (O) de acuerdo / (X) ixo

- Hace falta una contraseña para proteger la clave secreta. Es importante que podais recordar esta contraseña, pero no es recomendable escribirla en ningún sitio.

Introducid la contraseña:

Han de generarse muchos bits aleatoriamente. Es buena idea hacer alguna otra cosa (teclear, mover el ratón, usar los discos) durante la generación de números; esto da oportunidad al generador de números aleatorios de conseguir suficiente entropía.

```
+++++.....+++++.++++.++++.....+++++
+++++ ++.++++.++++.++++.....
.++++.++++.....++++.++++.++++..... >++++>...
+++++....
```

- Finalmente, después de generar algo de entropía, obtendremos nuestra clave

```
gpg: key 70EDBB59 marked as ultimately trusted
se han creado y firmado las claves y firmado las claves pública y secreta
```

```
gpg: se está comprobando la base de datos de confianza
```

```
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model gpg: depth: 0
valid: 3 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 3u
```

```
gpg: la próxima comprobación de la base de datos de confianza será el 2006-06-09
```

```
pub 1024D/70EDBB59 2006-05-20 [expires: 2006-06-19] Key fingerprint =
E555 61C8 0914 490A C525 6AF8 358E 895D 70ED BB59
```

```
uid Shiyali Ramamrita Ranganathan (Dirección Documentación)
sub 2048g/2E182CC0 2006-05-20 [expires: 2006-06-19]
```

### 3. Generar el certificado de revocación

Un certificado de revocación sirve por si un dia se pierde la clave, no la utilizamos, etc. Para crearlo hacemos:

```
gpg -output cert_revoc_arch.asc -gen-revoke -armor 70EDBB59
```

```
sec 1024D/70EDBB59 2006-05-20 Shiyali Ramamrita Ranganathan (Dirección
Documentación)
```

```
Create a revocation certificate for this key? (y/N)
```

Ahora hay que responder algunas preguntas:

Seleccionad la razón de la revocación:

0 = No se ha especificado ninguna razón

1 = La clave ha estado comprometida

2 = La clave ha estado substituida

3 = La clave ya no se usa

Q = Cancela

(Seguramente quereis seleccionar 1 aquí)

¿Vuestra decisión?

Podemos poner 3, ya que no la usamos, y en el texto *Certificado por si perdemos la clave o la contraseña*

¿Vuestra decisión? 3

Introducid una descripción opcional; finalizad con una línea en blanco:

> certificado por si perdemos la clave o el password

>

Razón de la revocación: La clave ya no se usa certificado por si perdemos la clave o el password

Is this okay? (y/N) y

You need a passphrase to unlock the secret key for user: "Shiyali Ramamrita Ranganathan (Adreça Documentació)" 1024-bit DSA key, ID 70EDBB59, created 2006-05-20

Y vemos el resultado final:

Se ha creado un certificado de revocación. Por favor, guardarlo en un medio que podais esconder; si Mallory consigue acceso a ete certificado puede utilizarlo para hacer vuestra clave inservible. Es inteligente imprimir este certificado y esconderlo, por si el vuestro medio se vuelve ilegible. Pero ir con cuidado: el sistema de impresión de vuestra máquina podria almacenar los datos y hacerlos accesibles a otros.

Si hace falta más adelante activar definitivamente la revocación, sólo hará falta importar este (cert-revoc-arch.asc) y después mandarlo al servidor.

```
$gpg --import cert-revoc-arch.asc
```

```
$gpg --send-keys 70EDBB59
```

#### 4. Enviar y recibir claves al sertvidor de claves

Si no tenemos ningún problema de configuración, solo falta enviar la clave al servidor por defecto (en el nuestro caso `hkp://subkeys.pgp.net`) utilizando el comando:

```
gpg --send-keys 70EDBB59
```

donde 70EDBB59 es el ID de nuestra clave. Para acabar, si queremos mostrar la impronta de la clave (fingerprint), sólo falta:

```
gpg -fingerprint 70EDBB59
```

## 5. ¿Y ahora, qué?

Una vez tenemos nuestra clave generada podemos configurar el gestor de correo para firmar y/o encriptar nuestros mensajes. Se ha de tener en cuenta que:

- Firmar un correo es poner unas marcas de forma que, cualquier persona que tenga la parte pública con la que se ha firmado ese mensaje, pueda verificarlo.
- Cifrar un mensaje es un proceso donde el emisor codifica el mensaje a través de la clave pública del destinatario y solamente se puede descifrar con la clave privada del destinatario.
- Disponer de la clave pública de una persona no quiere decir nada en especial. Para asegurarnos que es esa persona la que ha emitido el mensaje, hemos de verificar con un documento de identidad que esa persona es quien dice que es y que esa es su clave pública.
- Una vez verificada la autenticación de la persona, entonces podemos firmar la clave de esa persona.

## 6. Fuentes de información

Fuente original del documento: <http://bulma.net/body.phtml?nIdNoticia=1684>