# Cifrado y firma digital con GnuPG

## Máster en Economía Digital e Industrias Creativas


Miguel Vidal

ETSIT, URJC

Twitter: @mvidallopez

Israel Herraiz

ETSICCP, UPM

Twitter: @herraiz


1 de octubre de 2011

# Índice

# ¿Qué es PGP?

- Pretty Good Privacy: primera implementación popular del cifrado de clave pública (RSA).
- Creado en solitario por Phil Zimmermann en 1991, un activista de los ciberderechos.
- Lo puso a disposición de todo el mundo (código fuente incluido) vía FTP.
- Se popularizó rápidamente y recibió caluroso apoyo de la comunidad de criptógrafos para la versión 2.0 (1992).

# ¿Qué es PGP? (2)

- Zimmermann tuvo muchos problemas con la patente de RSA y fue investigado por presunta violación de las leyes de exportación de armas.
- Su caso fue finalmente archivado en 1996.
- Creó entonces una compañía (PGP Inc.) para explotar comercialmente PGP.
- La patente RSA expiró en septiembre del 2000.
- PGP Inc. fue adquirida por NAI y luego por PGP Corporation.

# Índice

# OpenPGP

- Estándar abierto para PGP propuesto por Zimmermann.
- Aceptado por la IETF en 1998.
- RFC 4880 (noviembre 2007), sucesor del RFC 2440.
- Tiene muchas implementaciones, sobre todo para clientes de correo.
- La implementación libre de OpenPGP se llama GnuPG.

# Índice

# GnuPG

- GNU Privacy Guard: Implementación libre (GPLv3) de OpenPGP.
- Promovido por la FSF desde 1999.
- Fue apoyado por el Gobierno alemán en sus inicios (financiaron la documentación y el *port* a Windows).
- Se basa en interfaz de línea de comandos, con diversos *frontends* gráficos.
- Existen *ports* a los sistemas operativos más utilizados (Windows, MacOS, Linux...).

# Descargar GnuPG

- Descargar e instalar GnuPG (hay versiones para Windows, MacOS, Linux, *BSD...)
  - Para MacOS X, Mac GPG: `http://gpgtools.org`
  - Para Windows, gpg4win: `http://gpg4win.org`
- Hay GUIs: por ejemplo Seahorse para Gnome o WPA para Windows.

*Opcional:* podemos comprobar la integridad del fichero:

```
$ sha1sum gnupg-w32cli-1.4.10b.exe
```

# Instalar GnuPG

La instalación en Unix/Linux por medio del sistema de paquetes correspondiente. Por ejemplo:

```
$ apt-get install gnupg    # Linux (Debian, Ubuntu)
$ pkg_add -vv gnupg-1.4.10p0    # OpenBSD
```

# Generar claves

Generamos nuestro par de claves (una privada y otra pública):

```
$ gpg --gen-key
```

Nos preguntará por el algoritmo a usar (RSA), tamaño de clave (2048 bits), expiración (0), nombre real, email y comentario.

También desde línea de comandos podemos pasar parámetros:

```
$ gpg --gen-key -t rsa -b <bits 2048 o 4096>
```

# Obtén tu key ID

```
$ gpg --list-keys

/home/mvidal/.gnupg/pubring.gpg
---------------------------------
pub  1024D/F724244F 1999-08-27
uid  Miguel Vidal <mvidal@gsyc.urjc.es>
uid  Miguel Vidal (URJC) <miguel.vidal@urjc.es>
uid  Miguel Vidal <mvidal@computer.org>
sub  1024g/A2B68952 1999-08-27
```

# Índice

# Exporta tu clave pública

Para poder enviar una clave pública a otra persona (sin usar un servidor de claves), tenemos que exportarla:

```
$ gpg --armor --output mvidal.asc --export mvidal
```

# Importa una clave pública

Para importar una clave pública (sin servidor de claves) a nuestro anillo de claves:

```
$ gpg --import mvidal.gpg
```

# Sube tu clave a un servidor de claves

Publica tu clave en un servidor de claves:

```
$ gpg --keyserver pgp.rediris.es --send-keys F724244F
```

Que debe devolverte algo como:

```
$> gpg: sending key F724244F
```

# Descarga una clave pública

Descarga una clave pública de un servidor de claves:

```
gpg --keyserver pgp.rediris.es --recv-keys [key_id]
```

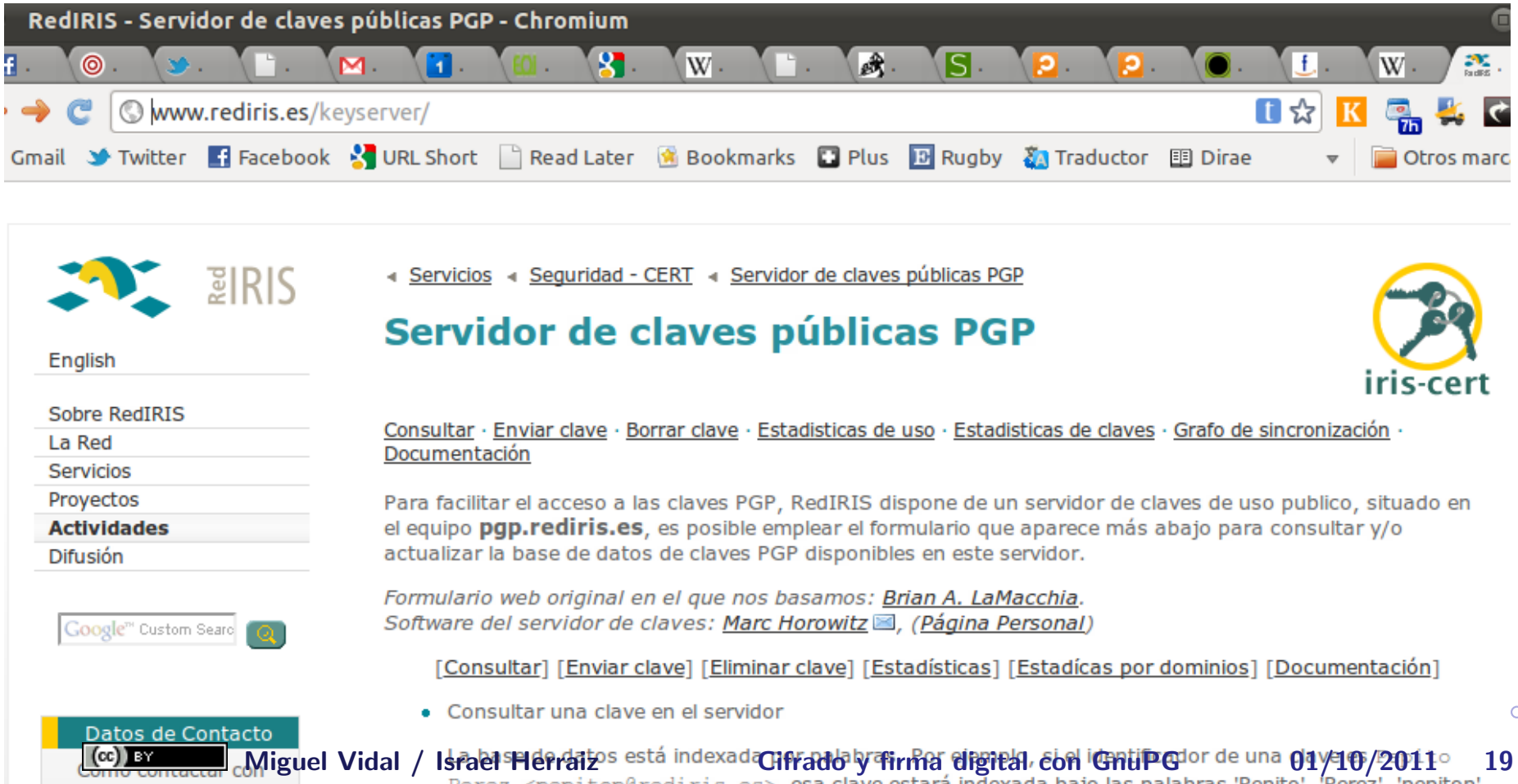
O también:

```
$ gpg --keyserver pgp.rediris.es --search-keys FE0A7AF3
gpg: buscando "FE0A7AF3" de hkp servidor pgp.rediris.es
(1)   Israel Herraiz <israel.herraiz@upm.es>
```

# Localizar un identificador de clave pública

¿Cómo localizo la key ID de alguien?

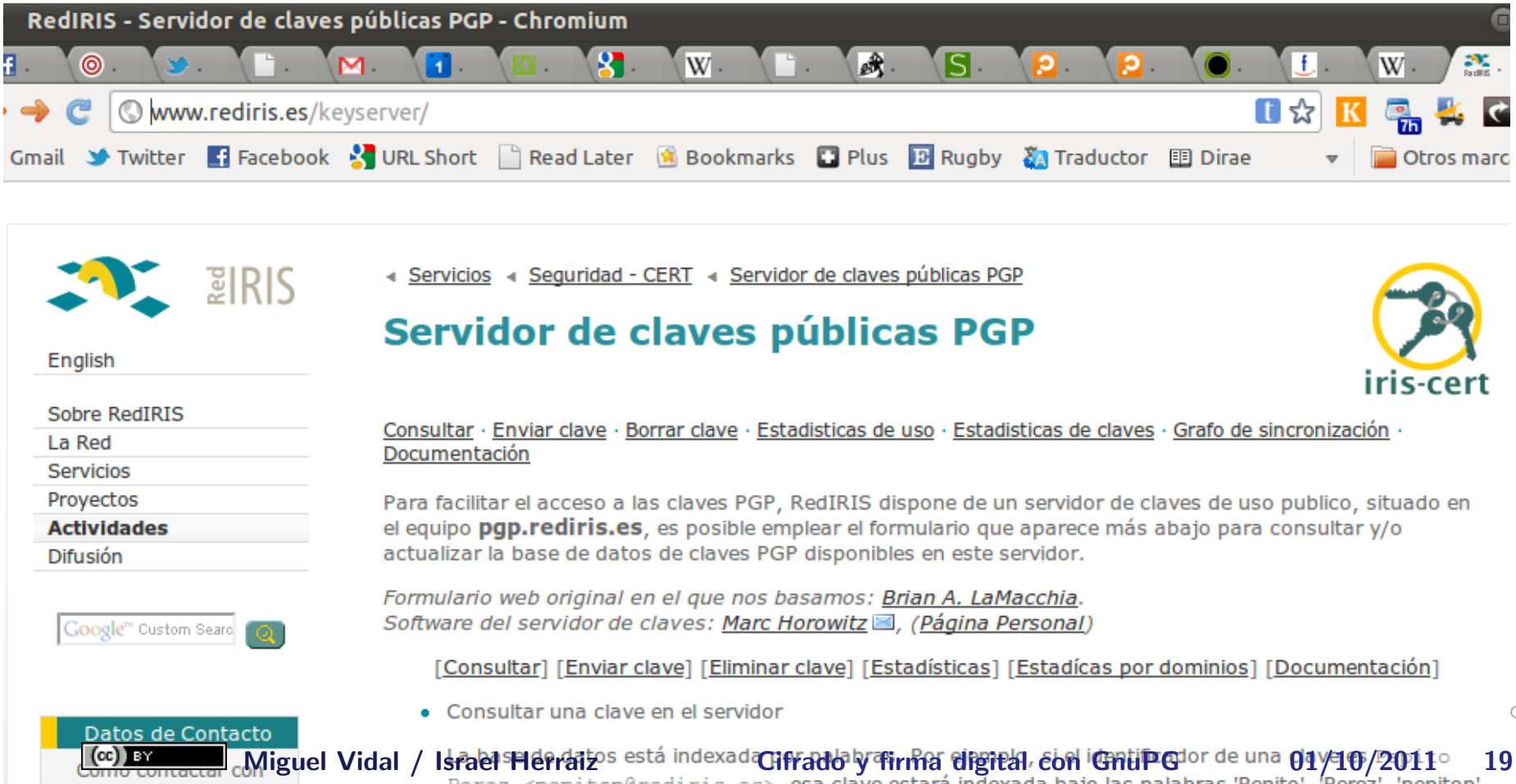
http://www.rediris.es/keyserver



# Localizar un identificador de clave pública

¿Cómo localizo la key ID de alguien?

`http://www.rediris.es/keyserver`



# Índice

# Firmar la clave de otra persona

1. Mostramos y solicitamos que se nos muestre un documento que acredite la identidad de cada cual.
2. Intercambiamos nuestras claves públicas o bien la huella digital con la persona a la que vamos a firmarle la clave (y va a firmar la nuestra).
3. La huella digital puede entregárnosla en un papel y después podemos comprobar en nuestro ordenador que efectivamente coincide con la clave pública que poseemos de esa persona.
4. Una vez comprobado, podemos proceder a firmar su clave y otorgarle confianza.

# Firmar la clave de otra persona

Descarga y comprueba las huellas y claves de tus conocidos:

```
$ gpg --keyserver pgp.rediris.es --recv-keys [key_id]
$ gpg --fingerprint [key_id]
```

# Firmar la clave de otra persona

Firma cada una de las claves verificadas de tus conocidos, y súbelas al servidor de claves:

```
$ gpg --sign-key [key_id]
$ gpg --keyserver pgp.rediris.es --send-keys [key_id]
```

# Observaciones

- Solo se debe firmar una clave cuando se esté totalmente seguro de que dicha clave es auténtica.
- Esto solo puede suceder si se recibe la clave en mano.
- Por eso, normalmente el procedimiento de firma se realiza presencialmente.

# Índice

# Ejemplo de firmado de una clave

```
$ gpg --sign-key herraiz
Orden> sign
¿Está realmente seguro de querer firmar esta clave
con su clave: "Miguel Vidal (URJC) <miguel.vidal@urjc.es>"(F724244F)?

¿Firmar de verdad? sí
Orden> quit
¿Grabar cambios? sí
```

Si distribuimos nuestra clave pública, ya aparecerá con las firmas efectuadas.

# Resumen

1. Hay que asegurarnos de que quien nos da la clave es efectivamente quien dice ser (algo imposible de verificar si nos descargamos su clave de un repositorio público o si nos la envía por email).
2. Es importante entender y mantener la consistencia de las claves de confianza (y nunca firmar si el canal por el que la hemos recibido no es fiable).
3. A diferencia de otros sistemas de criptografía de clave pública que confían en una autoridad certificadora (CA), aquí todo se basa en un sistema descentralizado de fuentes de confianza.

# Índice

# Cifrar y descifrar un documento

El documento que se desea cifrar es la entrada, `recipient` es el destinatario y la salida es el documento cifrado:

```
$ gpg --output documento.gpg --encrypt --recipient \
      fulano@foo.es documento
```

Para descifrar (-d):

```
$ gpg --output documento --decrypt documento.gpg
```

# Índice

# Cifrado simétrico

1. Funcionalidad no muy conocida de pgp/gnupg.
2. No requiere uso de clave pública ni privada.
3. Útil para cifrar ficheros para uno mismo.
4. Método rápido para usar cifrado fuerte con usuarios que no usan gpg/pgp (mucho mejor que el cifrado fácilmente crackeable de Word o de Winzip).

# Cómo cifrar con clave simétrica

```
gpg --symmetric filename
```

Salida ASCII (para intercambiar por email):

```
gpg --symmetric --armor filename
```

(Se nos solicitará una clave: no usar la misma contraseña de nuestra clave privada.)

Para descifrar, se hace de la forma habitual:

```
gpg -d filename
```

# Índice

# Firmar un documento

El documento a firmar es la entrada y la salida es el documento firmado:

```
gpg --armor --output document.sig --sign document
```

Verificar la integridad de un documento firmado:

```
gpg --verify document.sig
```

Si además de verificar la firma queremos recuperar el documento:

```
gpg --output document --decrypt document.sig
```

# Cifrado y firma digital con GnuPG

## Máster en Economía Digital e Industrias Creativas


Miguel Vidal

ETSIT, URJC

Twitter: @mvidallopez

Israel Herraiz

ETSICCP, UPM

Twitter: @herraiz


1 de octubre de 2011