

Seguridad en Sistemas de Información: Prueba Computacional 2

Jorge E. Chávez Saab

CINVESTAV-IPN

7 de Julio 2020

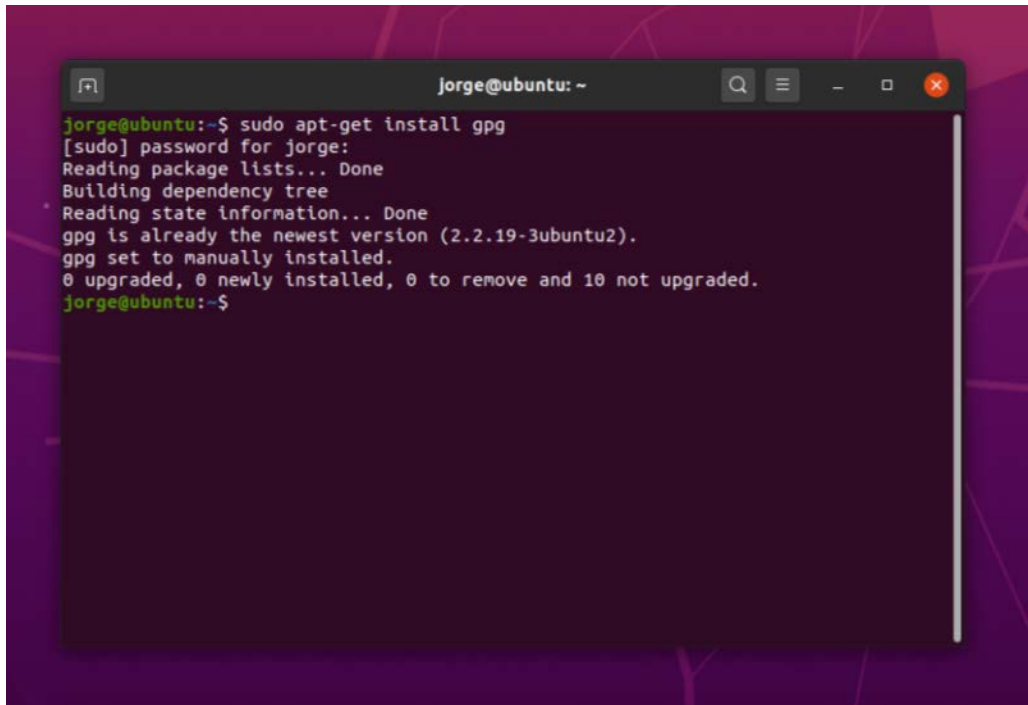
En esta práctica describiremos los pasos, experiencia e impresiones al instalar GnuPG en una distribución Ubuntu y hacerlo interactuar con la aplicación de correo electrónico Thunderbird.

1. Instalación

Con una búsqueda en Google me fue muy fácil encontrar la página web para GnuPG, la cual contiene una descripción muy completa de la herramienta.



En la sección de Descargas, se anuncia que muchas distribuciones de Linux, como Ubuntu, ya incluyen un paquete instalable de GnuPG. Al usar el comando `sudo apt-get install gpg` en mi máquina comprobé que esto era cierto, y que el paquete de hecho ya había sido instalado.

A terminal window titled 'jorge@ubuntu: ~' showing the command 'sudo apt-get install gpg' and its output. The output indicates that gpg is already installed and is the newest version (2.2.19-3ubuntu2). The terminal text is as follows:

```
jorge@ubuntu:~$ sudo apt-get install gpg
[sudo] password for jorge:
Reading package lists... Done
Building dependency tree
Reading state information... Done
gpg is already the newest version (2.2.19-3ubuntu2).
gpg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
jorge@ubuntu:~$
```

2. Pruebas en la línea de comandos

Al usar el comando `gpg --help` me encontré con la siguiente documentación:

```
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
```

```
Syntax: gpg [options] [files]
Sign, check, encrypt or decrypt
Default operation depends on the input data
```

Commands:

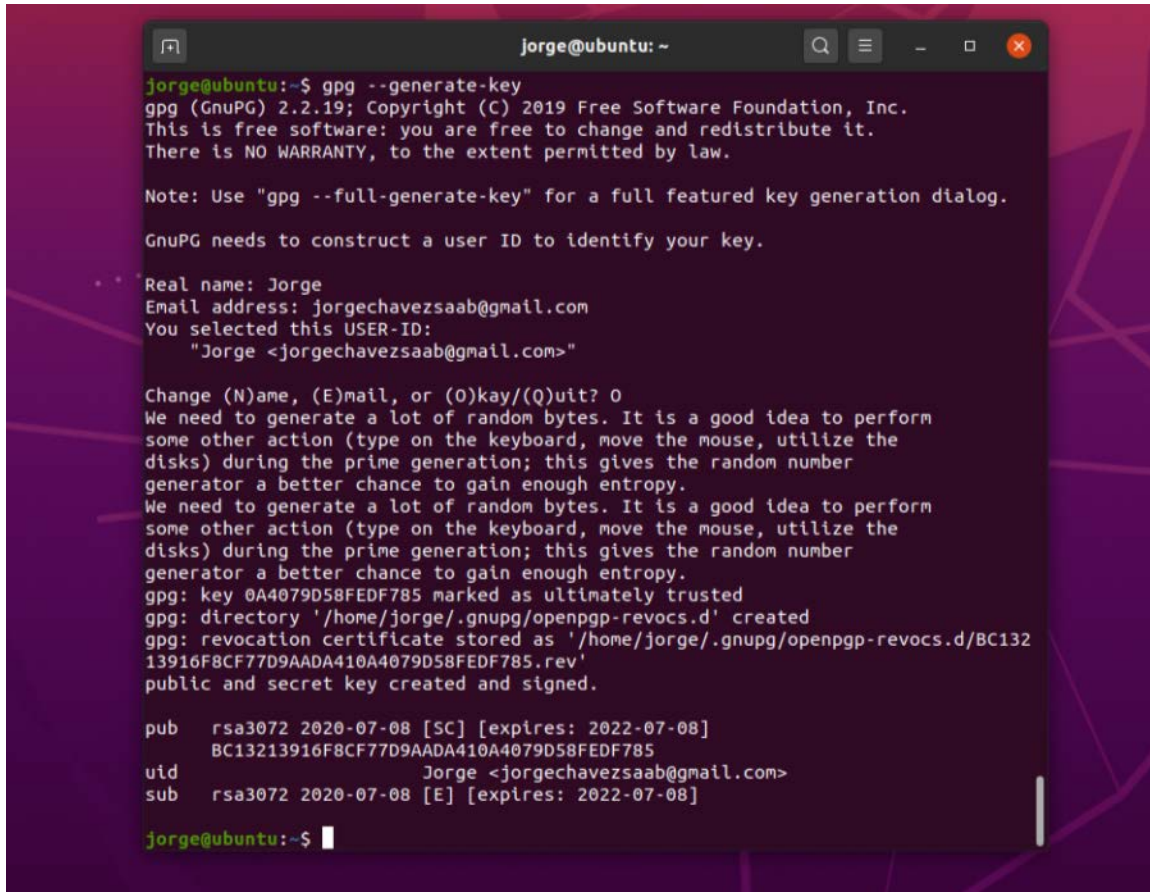
-s, --sign	make a signature
--clear-sign	make a clear text signature
-b, --detach-sign	make a detached signature
-e, --encrypt	encrypt data
-c, --symmetric	encryption only with symmetric cipher
-d, --decrypt	decrypt data (default)
--verify	verify a signature

-k, --list-keys	list keys
--list-signatures	list keys and signatures
--check-signatures	list and check key signatures
--fingerprint	list keys and fingerprints
-K, --list-secret-keys	list secret keys
--generate-key	generate a new key pair
--quick-generate-key	quickly generate a new key pair
--quick-add-uid	quickly add a new user-id
--quick-revoke-uid	quickly revoke a user-id
--quick-set-expire	quickly set a new expiration date
--full-generate-key	full featured key pair generation
--generate-revocation	generate a revocation certificate
--delete-keys	remove keys from the public keyring
--delete-secret-keys	remove keys from the secret keyring
--quick-sign-key	quickly sign a key
--quick-lsign-key	quickly sign a key locally
--sign-key	sign a key
--lsign-key	sign a key locally
--edit-key	sign or edit a key
--change-passphrase	change a passphrase
--export	export keys
--send-keys	export keys to a keyserver
--receive-keys	import keys from a keyserver
--search-keys	search for keys on a keyserver
--refresh-keys	update all keys from a keyserver
--import	import/merge keys
--card-status	print the card status
--edit-card	change data on a card
--change-pin	change a card's PIN
--update-trustdb	update the trust database
--print-md	print message digests
--server	run in server mode
--tofu-policy VALUE	set the TOFU policy for a key

Options:

-a, --armor	create ascii armored output
-r, --recipient USER-ID	encrypt for USER-ID
-u, --local-user USER-ID	use USER-ID to sign or decrypt
-z N	set compress level to N (0 disables)
--textmode	use canonical text mode
-o, --output FILE	write output to FILE
-v, --verbose	verbose
-n, --dry-run	do not make any changes
-i, --interactive	prompt before overwriting
--openpgp	use strict OpenPGP behavior

La documentación se me hizo bastante clara y el uso de los comandos muy intuitivo. Lo primero que hice fue usar el comando `gpg --generate-key` para crear una llave nueva que pudiera usar como prueba. Por medio de una serie de mensajes la herramienta me guió para crear un nuevo usuario y una contraseña para mi llave, todo de forma muy intuitiva.



```
jorge@ubuntu: ~  
jorge@ubuntu:~$ gpg --generate-key  
gpg (GnuPG) 2.2.19; Copyright (C) 2019 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.  
  
Note: Use "gpg --full-generate-key" for a full featured key generation dialog.  
  
GnuPG needs to construct a user ID to identify your key.  
  
Real name: Jorge  
Email address: jorgechavezsaab@gmail.com  
You selected this USER-ID:  
"Jorge <jorgechavezsaab@gmail.com>"  
  
Change (N)ame, (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: key 0A4079D58FEDF785 marked as ultimately trusted  
gpg: directory '/home/jorge/.gnupg/openpgp-revocs.d' created  
gpg: revocation certificate stored as '/home/jorge/.gnupg/openpgp-revocs.d/BC132  
13916F8CF77D9AADA410A4079D58FEDF785.rev'  
public and secret key created and signed.  
  
pub  rsa3072 2020-07-08 [SC] [expires: 2022-07-08]  
    BC13213916F8CF77D9AADA410A4079D58FEDF785  
uid  Jorge <jorgechavezsaab@gmail.com>  
sub  rsa3072 2020-07-08 [E] [expires: 2022-07-08]  
  
jorge@ubuntu:~$
```

Una vez que cree mi clave, cree un archivo de prueba y use el comando `gpg -e hola.txt` para cifrarlo. La interfaz de usuario me pidió mi nombre de usuario y quién va a recibirlo (lo dirigí a mí mismo). Esto creo un nuevo archivo `hola.txt.gpg` y, como es de esperarse el texto en él es irreconocible. No obstante, al usar `gpg -d hola.txt.gpg` para descifrar el mensaje, recuperamos el contenido original del archivo que es el texto "hola".

```

jorge@ubuntu: ~
jorge@ubuntu:~$ gpg -e hola.txt
You did not specify a user ID. (you may use "-r")

Current recipients:

Enter the user ID. End with an empty line: Jorge

Current recipients:
rsa3072/8287BFBE8B3B7922 2020-07-08 "Jorge <jorgechavezsaab@gmail.com>"

Enter the user ID. End with an empty line: Jorge
gpg: skipped: public key already set

Current recipients:
rsa3072/8287BFBE8B3B7922 2020-07-08 "Jorge <jorgechavezsaab@gmail.com>"

Enter the user ID. End with an empty line:
jorge@ubuntu:~$ ls
Desktop  Downloads  hola.txt.gpg  Pictures  Templates
Documents  hola.txt  Music        Public    Videos
jorge@ubuntu:~$ cat hola.txt.gpg
#####y"
#####
#####vXqwe`hL(I)ad#####P:~Qaaiz***k'!$071w4d
#####G8ykwHba/oc#####BccRw/4rd###3n<3WwE?[@^&IADw!^MC<
#####G:z`m=#####YeTSpysS.#####E*:. >##0<,qJ5"0#####f###.8zd=U[#####n
&#7E#x&U%#H%#s,##S#S#~vetoWwDfU4##zFUB###/##,j&#r7w##/:/###a
mC#0#L#y;UH#G#I#k#A# #4+#_#]##K# #c"~##m
jorge@ubuntu:~$ gpg -d hola.txt.gpg
gpg: encrypted with 3072-bit RSA key, ID 8287BFBE8B3B7922, created 2020-07-08
"Jorge <jorgechavezsaab@gmail.com>"
hola

jorge@ubuntu:~$

```

También use el comando `gpg -s hola.txt` para firmar el mensaje y después

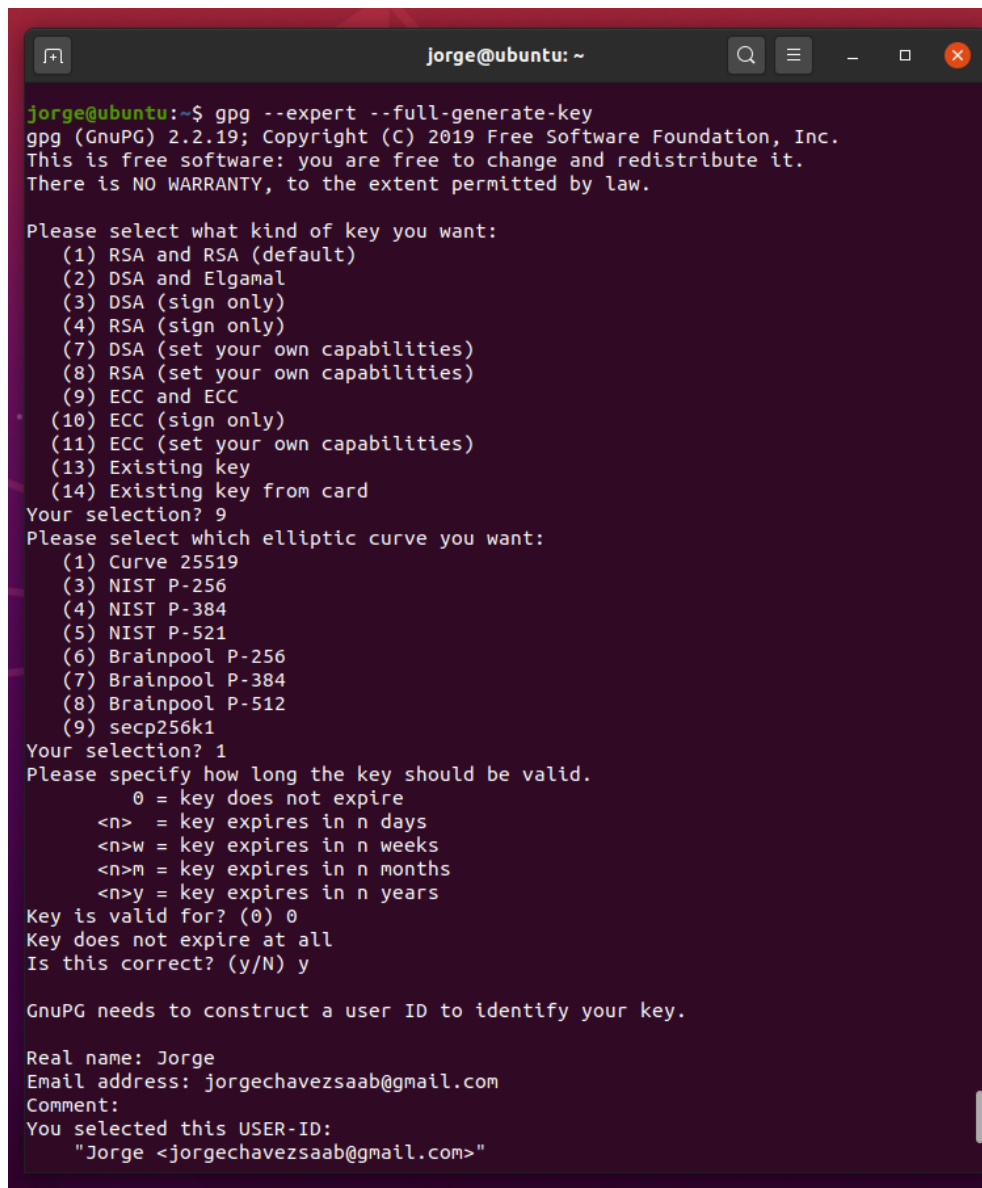
`gpg --verify hola.txt.gpg` para verificar la firma. Es interesante notar que en este caso el archivo `hola.txt.gpg` contiene datos adicionales correspondientes a la firma, pero en este caso el contenido original "hola" si es reconocible.

```

jorge@ubuntu: ~
jorge@ubuntu:~$ gpg -s hola.txt
File 'hola.txt.gpg' exists. Overwrite? (y/N) y
jorge@ubuntu:~$ gpg --verify hola.txt.gpg
gpg: Signature made Wed 08 Jul 2020 09:35:09 AM PDT
gpg: using RSA key BC13213916F8CF77D9AADA410A4079D58FEDF785
gpg: Good signature from "Jorge <jorgechavezsaab@gmail.com>" [ultimate]
jorge@ubuntu:~$ cat hola.txt.gpg
#####
@yS#####hola.txt_#hola
#####
!9w/A
@yS#_#
@yS#####
#####[<U#G#####Ct'Y*#####yMQ5月#3P#H###V
#)#h###D
Rs }g#|D#IU#L#MkX#z#DS*n#每Dm0s#:#^###h-###w{#x#?0B###r#hK#r#B###w#####V#
#IZ###fp&Sox###ex#鷄#.#,##+,#t7###HN#v#~##{###[###b#4`b#f#=# #wI###MD+
jorge@ubuntu:~$

```

Una duda que tuve a lo largo de todo este proceso fue que en nunca especifico qué protocolo quería usar: ni para la creación de llaves, ni para cifrar, ni para firmar. Obviamente la herramienta escogió protocolos por defecto pero me pareció inconveniente que la documentación de `gpg --help` incluye los algoritmos que se admiten pero nunca dice como especificar el uso de uno. Tuve que hacer un poco más de investigación para averiguar que el comando `gpg --expert --full-generate-key` es el que permite escoger que tipo de firma queremos generar y da muchas más opciones para el tipo de protocolo.



```
jorge@ubuntu:~$ gpg --expert --full-generate-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 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)
  (7) DSA (set your own capabilities)
  (8) RSA (set your own capabilities)
  (9) ECC and ECC
  (10) ECC (sign only)
  (11) ECC (set your own capabilities)
  (13) Existing key
  (14) Existing key from card
Your selection? 9
Please select which elliptic curve you want:
  (1) Curve 25519
  (3) NIST P-256
  (4) NIST P-384
  (5) NIST P-521
  (6) Brainpool P-256
  (7) Brainpool P-384
  (8) Brainpool P-512
  (9) secp256k1
Your selection? 1
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) 0
Key does not expire at all
Is this correct? (y/N) y

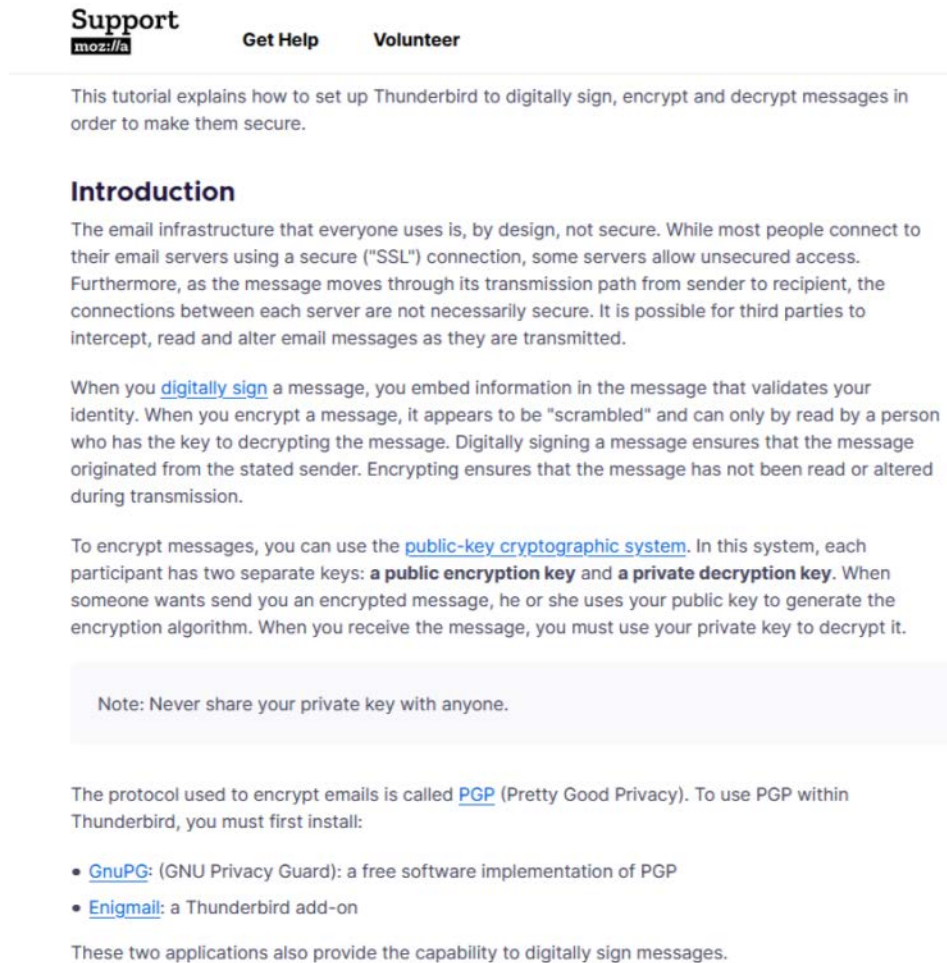
GnuPG needs to construct a user ID to identify your key.

Real name: Jorge
Email address: jorgechavezsaab@gmail.com
Comment:
You selected this USER-ID:
  "Jorge <jorgechavezsaab@gmail.com>"
```

Me pareció algo molesto que estas opciones estuvieran tan escondidas, aunque en retrospectiva creo que es adecuado para que los usuarios que busquen privacidad sin ser expertos en criptografía puedan usar la herramienta sin preocupación de todos estos detalles.

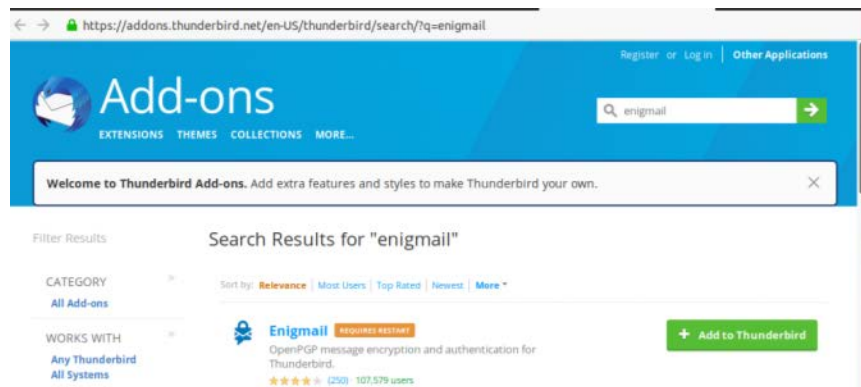
3. Integración a Thunderbird

Tuve que hacer una búsqueda en internet sobre cómo integrar GnuPG a Thunderbird, pero fácilmente llegué a una página en la sección de ayuda de Mozilla donde explica que esto se hace a través de una extensión llamada Enigmail y describe cómo instalarla.



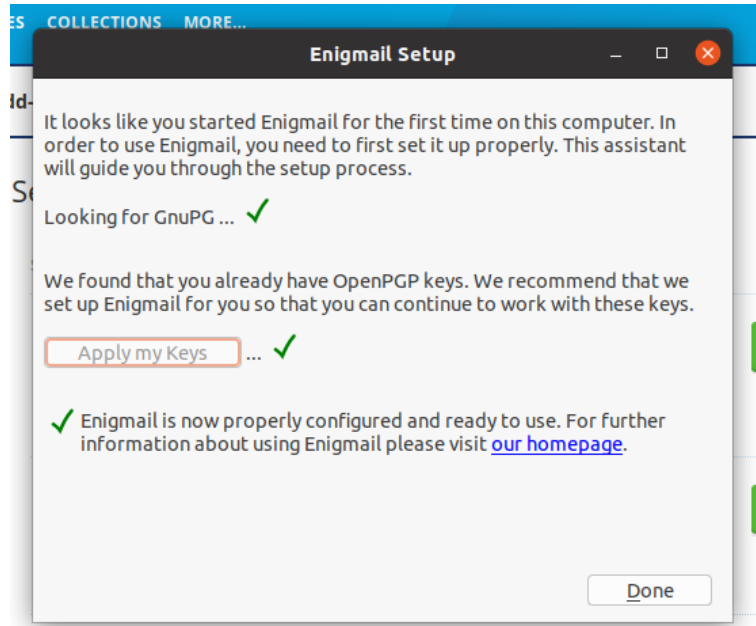
The screenshot shows the Mozilla Support page for Enigmail. At the top, there are navigation links for "Support", "Get Help", and "Volunteer". The main heading is "Support" with the Mozilla logo. Below this, a paragraph states: "This tutorial explains how to set up Thunderbird to digitally sign, encrypt and decrypt messages in order to make them secure." The section is titled "Introduction" and explains that email infrastructure is not secure by design. It describes how digital signing and encryption work. A note in a light blue box says: "Note: Never share your private key with anyone." Below this, it mentions that the protocol used is PGP (Pretty Good Privacy) and lists two applications: GnuPG (GNU Privacy Guard) and Enigmail (a Thunderbird add-on). It concludes that these two applications provide the capability to digitally sign messages.

Basta con navegar a la página de extensiones para Thunderbird y descargarla directamente:

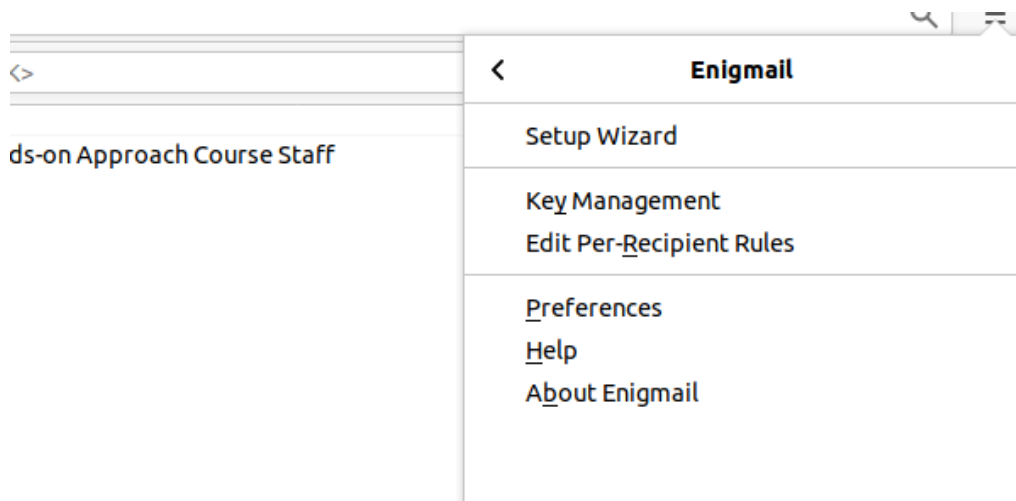


The screenshot shows the Thunderbird Add-ons page. The URL is "https://addons.thunderbird.net/en-US/thunderbird/search?q=enigmail". The page features a search bar with "enigmail" entered. Below the search bar, there is a "Welcome to Thunderbird Add-ons" message. The search results for "enigmail" are displayed, showing the extension's name, a "REQUIRES RESTART" badge, a description: "OpenPGP message encryption and authentication for Thunderbird.", a star rating of 4.5, and 107,579 users. A green "Add to Thunderbird" button is visible next to the extension details.

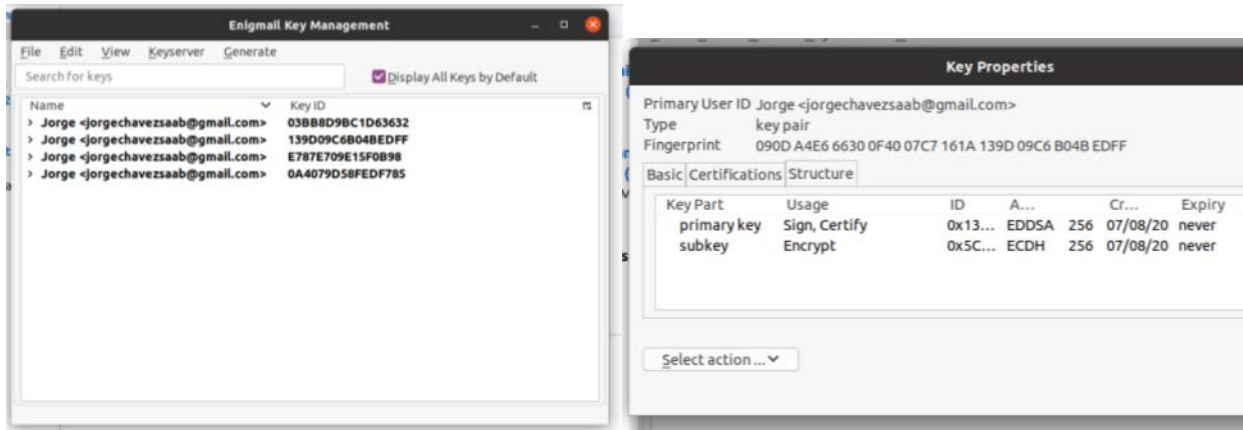
Con solo hacer click en el botón al extensión se agrega y detecta que ya tenemos GnuPG instalado y que ya hemos creado algunas llaves, las cuales podemos agregar automáticamente.



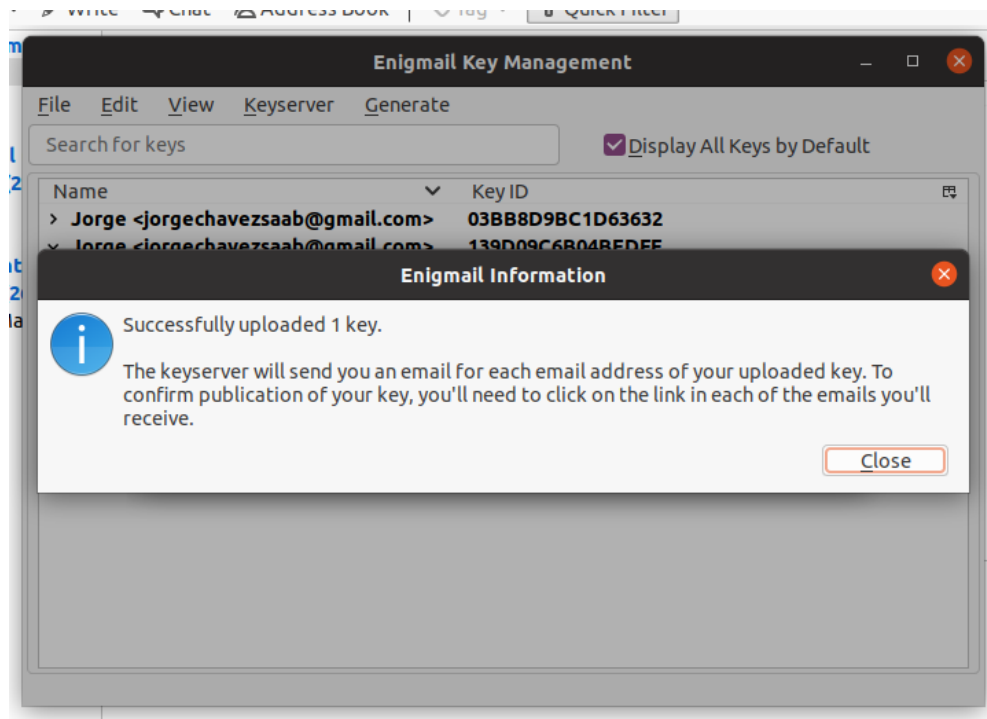
Ahora el menú de opciones de Thunderbird tiene una sección nueva para Enigmail, donde me tomé un tiempo para explorar los diferentes submenús.



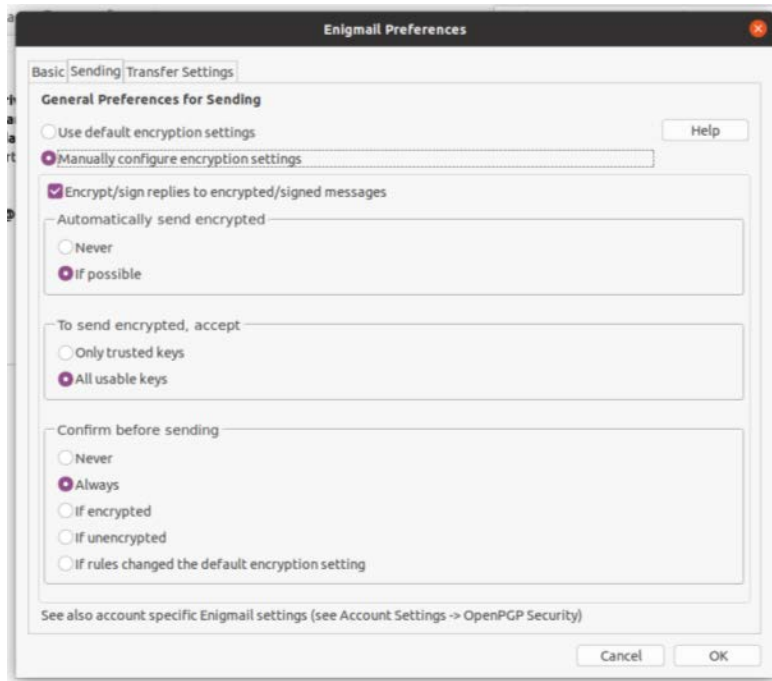
En Key Management encontré listadas todas las llaves que había creado, y al abrirlas puedo ver para qué algoritmo están diseñadas:



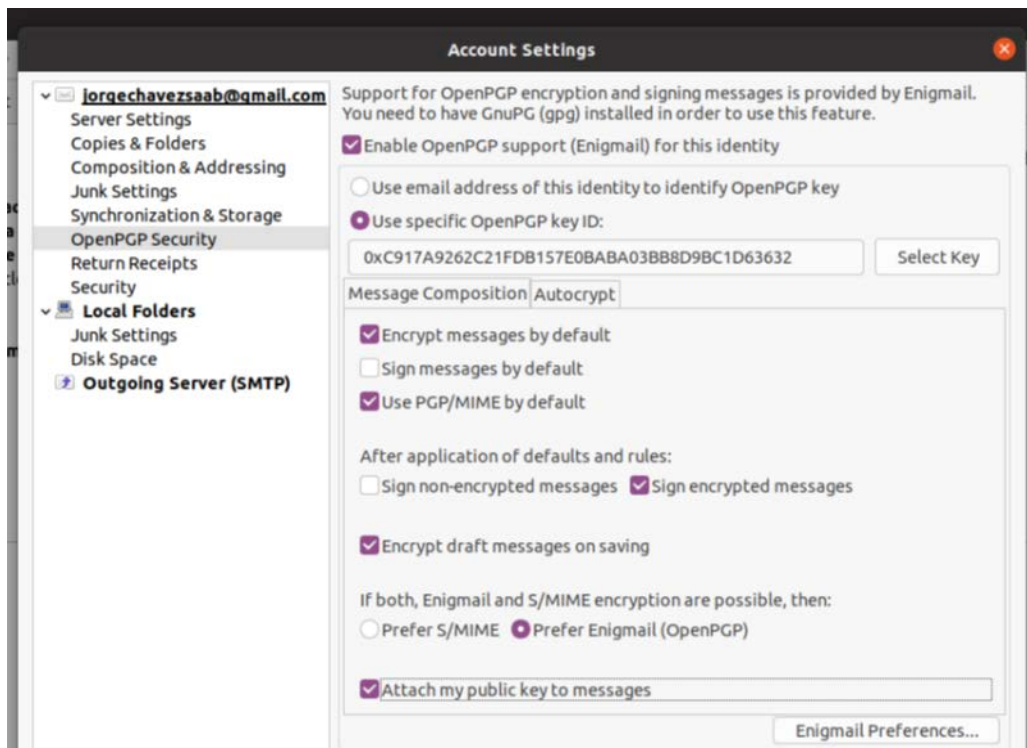
También hay una pestaña llamada Keyserver donde podemos buscar las llaves de otros usuarios por correo electrónico. Así mismo podemos subir nuestra llave al servidor para que otros usuarios la encuentren, para lo cual se nos envía un correo electrónico para confirmar propiedad de él.



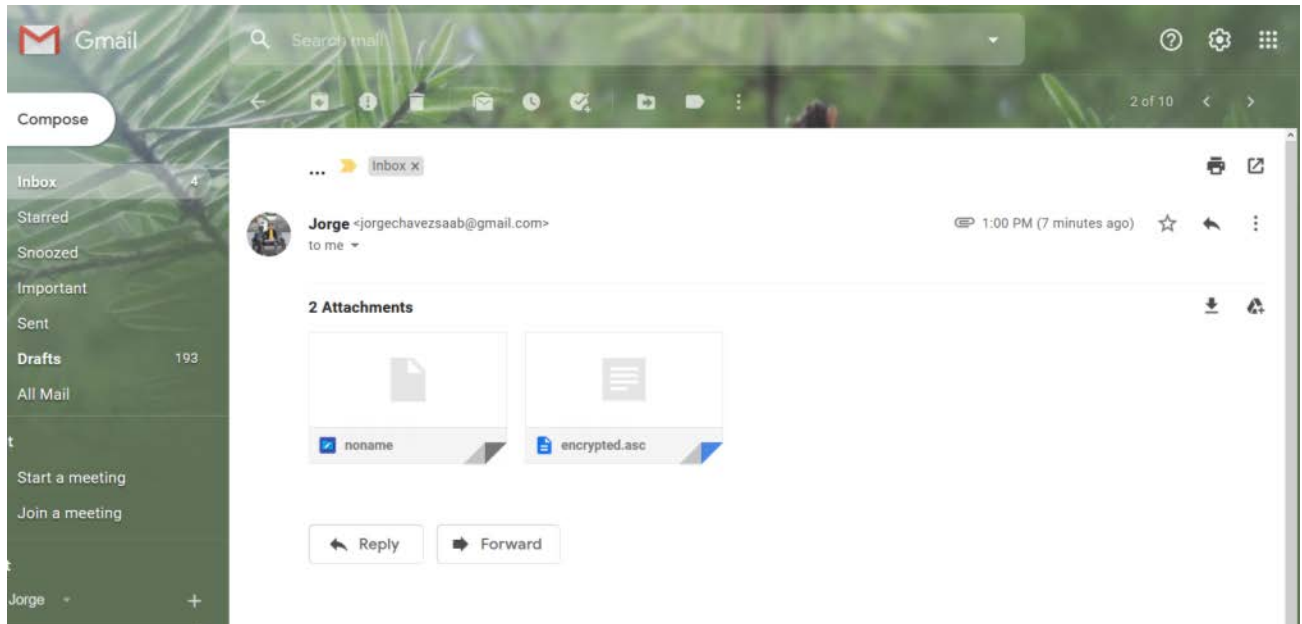
Yendo ahora al menú de Preferencias, sólo encontré las opciones para habilitar la firma y cifrado de mensajes al responder a mensajes que estaban firmados o cifrados.



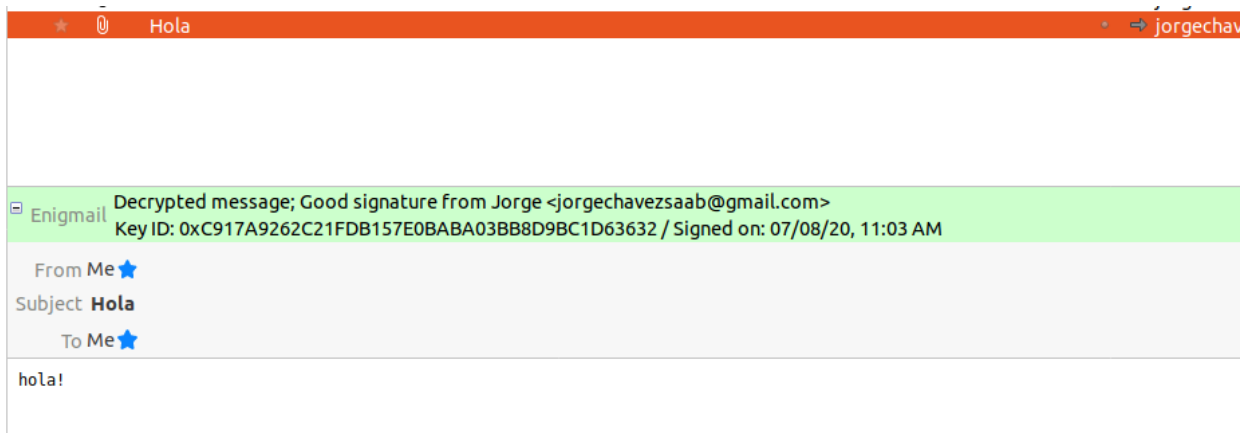
El diálogo también sugiere revisar las opciones en Account Settings > OpenPGP Security, que es donde encontré las opciones para habitar el cifrado y firmado en todos los mensajes salientes las cuales tuve que habilitar.



Habiendo completado la configuración, envié un mensaje de prueba a mí mismo. Al abrirlo desde la aplicación web de Gmail, el correo no tiene título y viene en forma del archivo cifrado adjunto, el cual podría descargar y extraer usando la línea de comandos de gpg.



Sin embargo, en Thunderbird la extensión se encarga en automático de verificar la firma y mostrar el contenido del mensaje, ya que la llave está en mi llavero.



Al tratarse de una aplicación comúnmente usada, la integración de GnuPG a Thunderbird es muy limpia y sencilla de usar, desde luego es mucho más intuitivo que usar la línea de comandos, y las interfaces gráficas para las preferencias me parecieron muy convenientes. Note por ejemplo la opción de cifrar/firmar mensajes sólo cuando se escribe a ciertas personas o dominios, lo cual me parece conveniente para aplicar medidas de seguridad al manejar información sensible o en ambientes profesionales pero no al tener conversaciones cotidianas con gente fuera de este ámbito que solo podría confundirse al recibir un correo así. En general tengo una impresión muy buena pues todo está automatizado de manera muy cómoda.