Seguridad en Sistemas de Información: Prueba Computacional 2

Jorge E. Chávez Saab

CINVESTAV-IPN

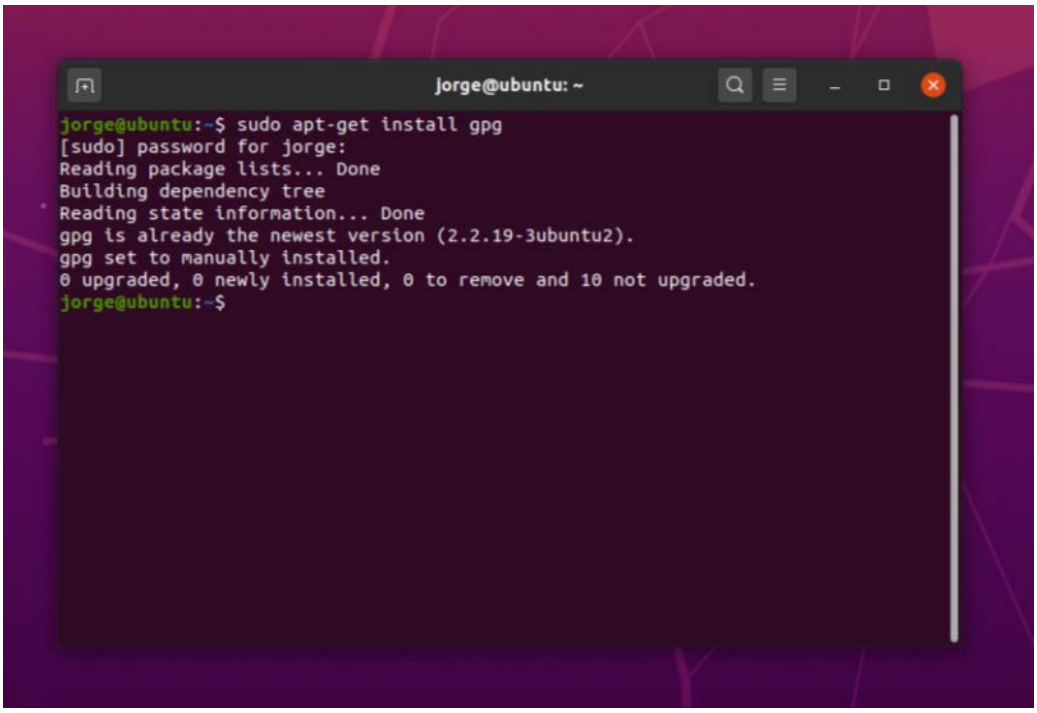
7 de Julio 2020

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.



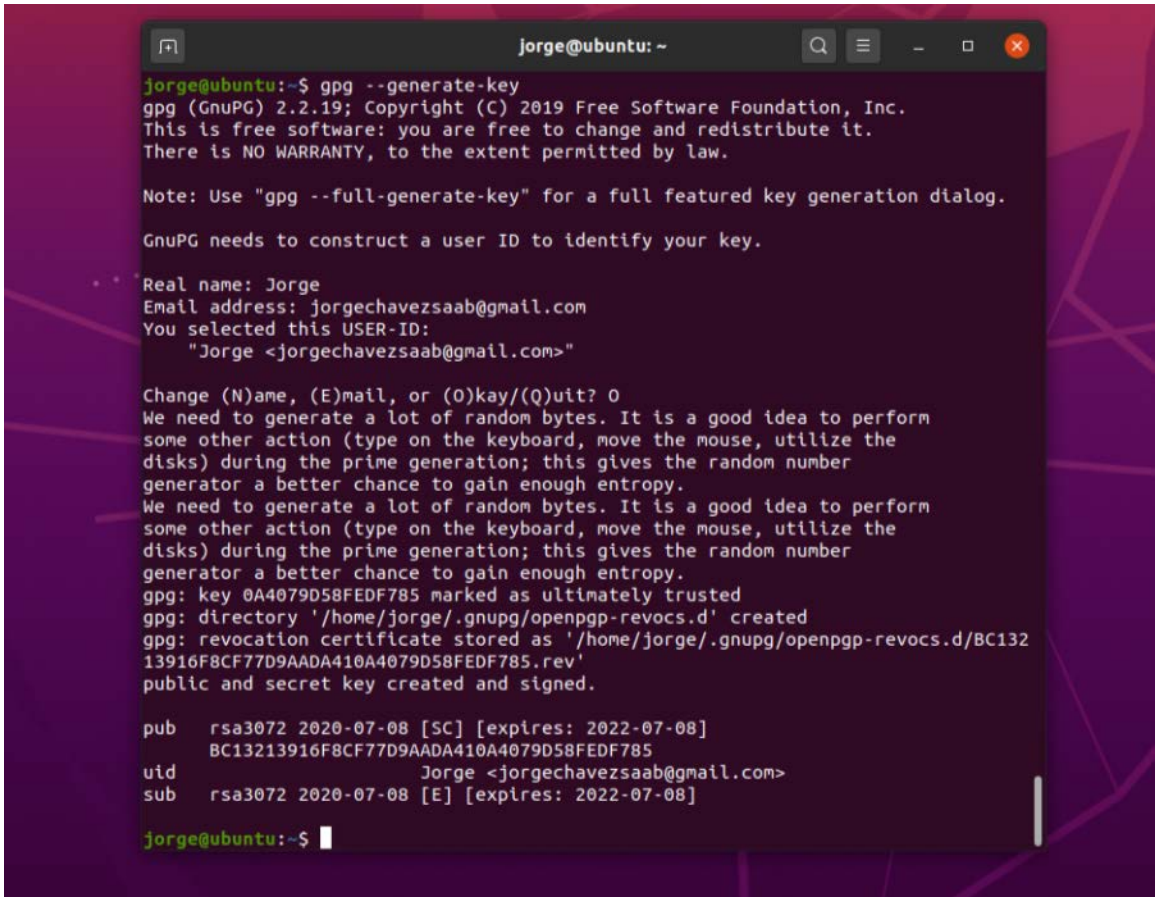
2. Pruebas en la línea de comandos

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

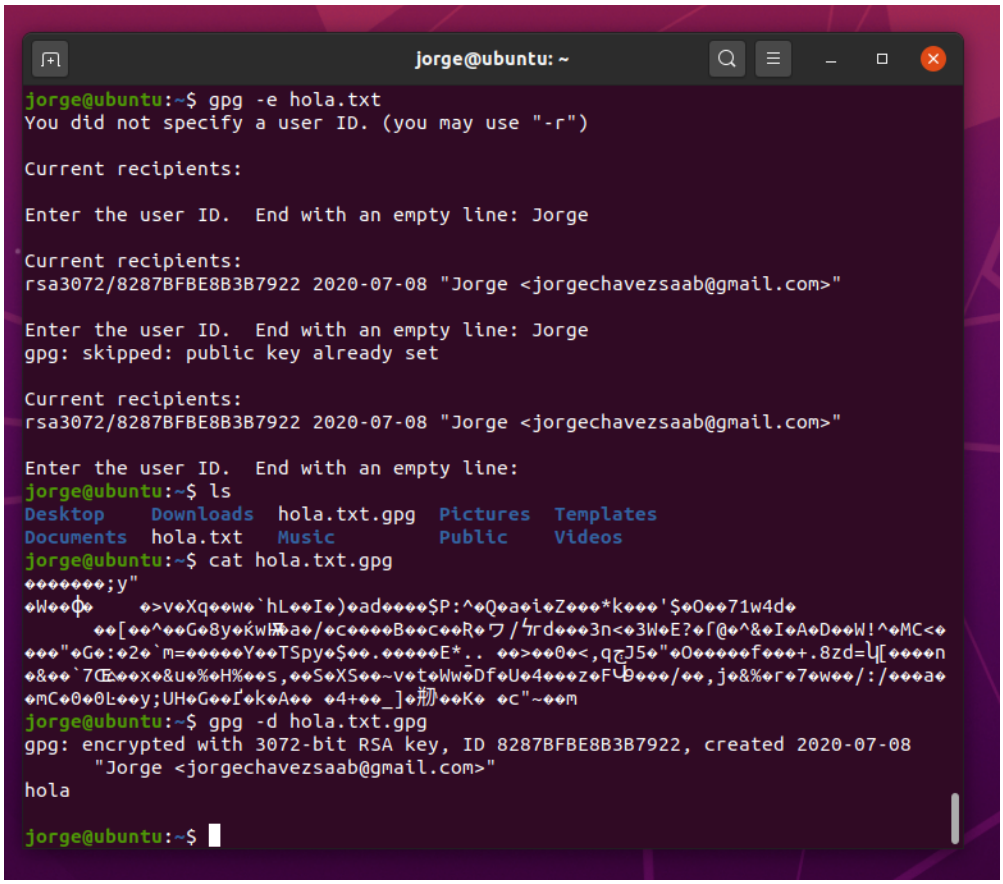
```
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2

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 guío para crear un nuevo usuario y una contraseña para mi llave, todo de forma muy intuitiva.

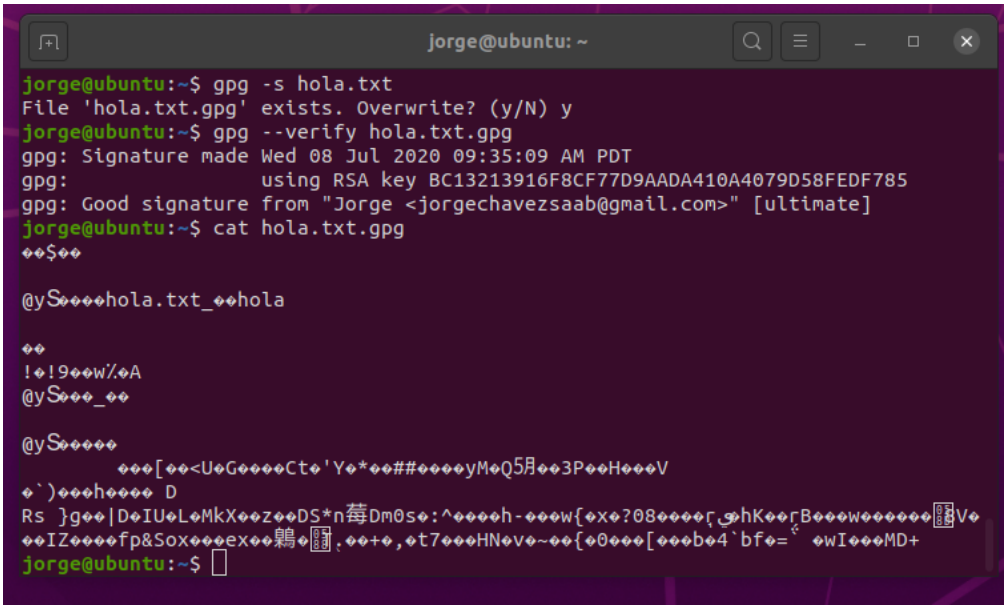


Una vez que cree mi clave, cree un archivo de prueba y use el comando `gpg -e hola.txt` para cifrarlo. La interfaz de usuario me pidió mi nombre de usuario y quién va a recibirlo (lo dirigí a mí mismo). Esto creo un nuevo archivo `hola.txt.gpg` y, como es de esperarse el texto en él es irreconocible.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

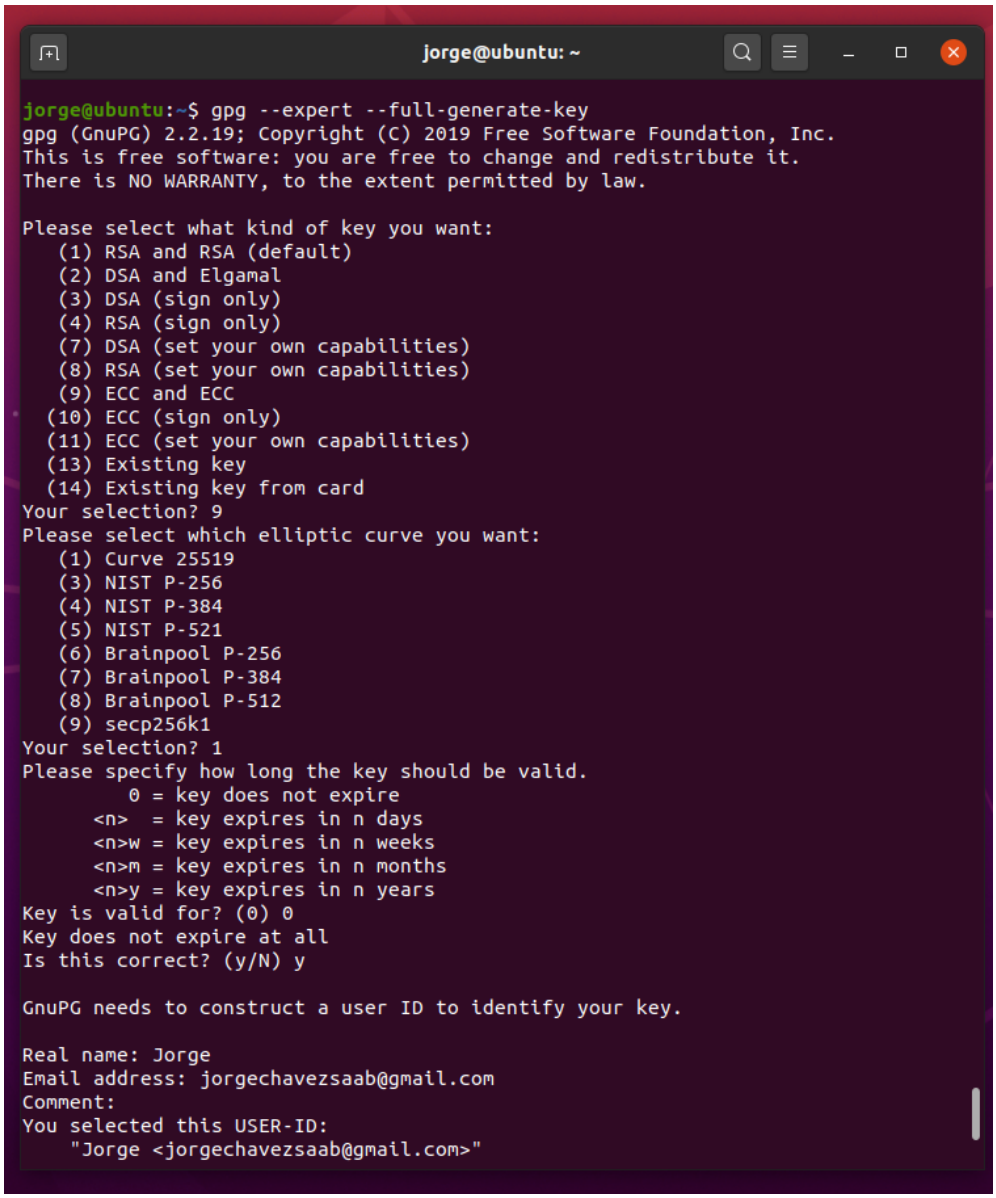


También use el comando `gpg -s hola.txt` para firmar el mensaje y después

`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.



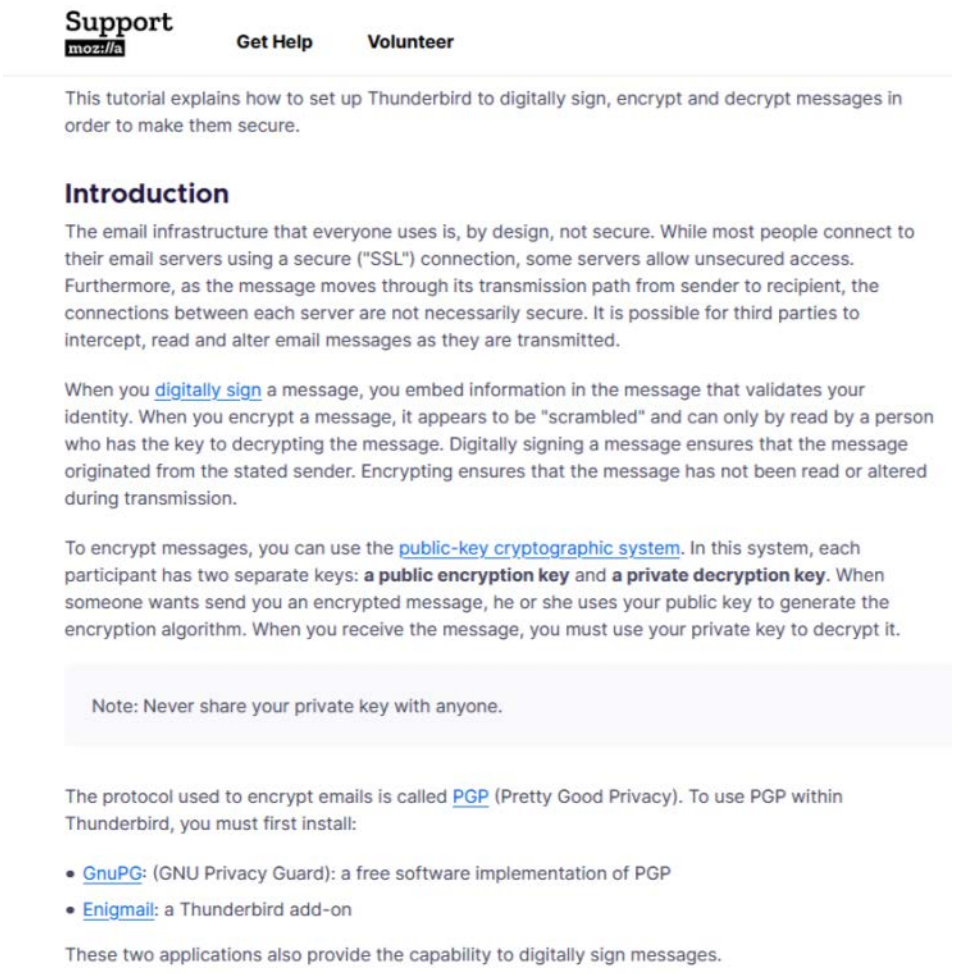
Una duda que tuve a lo largo de todo este proceso fue que en nunca especifique qué protocolo quería usar: ni para la creación de llaves, ni para cifrar, ni para firmar. Obviamente la herramienta escogió protocolos por defecto pero me pareció inconveniente que la documentación de `gpg --help` incluye 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.



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.



Support
moz://a
Get Help Volunteer

This tutorial explains how to set up Thunderbird to digitally sign, encrypt and decrypt messages in order to make them secure.

Introduction

The email infrastructure that everyone uses is, by design, not secure. While most people connect to their email servers using a secure ("SSL") connection, some servers allow unsecured access. Furthermore, as the message moves through its transmission path from sender to recipient, the connections between each server are not necessarily secure. It is possible for third parties to intercept, read and alter email messages as they are transmitted.

When you digitally sign a message, you embed information in the message that validates your identity. When you encrypt a message, it appears to be "scrambled" and can only by read by a person who has the key to decrypting the message. Digitally signing a message ensures that the message originated from the stated sender. Encrypting ensures that the message has not been read or altered during transmission.

To encrypt messages, you can use the public-key cryptographic system. In this system, each participant has two separate keys: **a public encryption key** and **a private decryption key**. When someone wants send you an encrypted message, he or she uses your public key to generate the encryption algorithm. When you receive the message, you must use your private key to decrypt it.

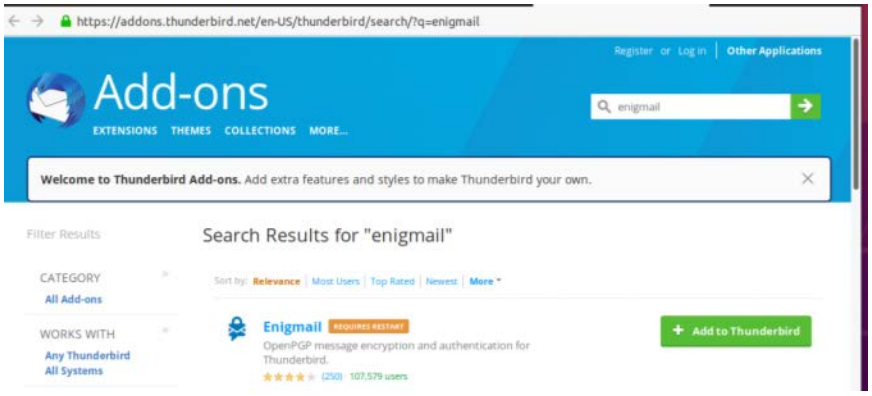
Note: Never share your private key with anyone.

The protocol used to encrypt emails is called PGP (Pretty Good Privacy). To use PGP within Thunderbird, you must first install:

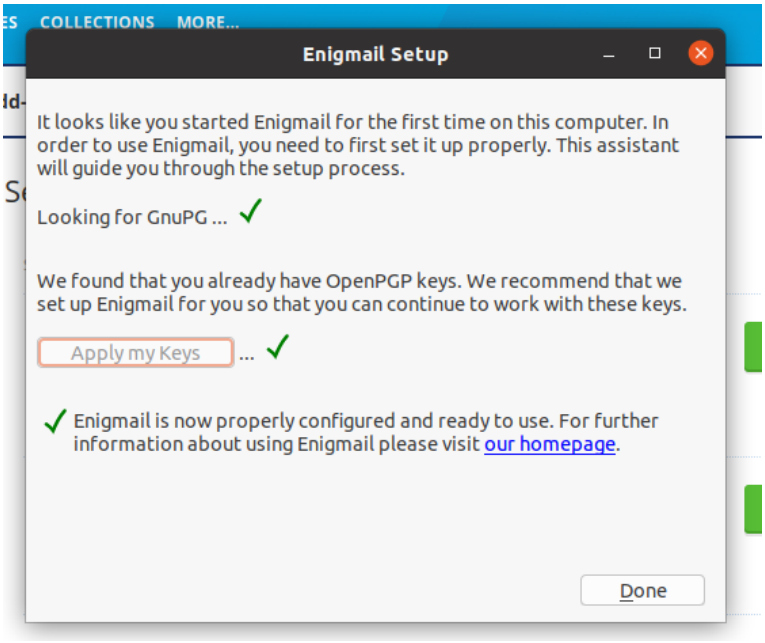
- GnuPG: (GNU Privacy Guard): a free software implementation of PGP
- Enigmail: a Thunderbird add-on

These two applications also provide the capability to digitally sign messages.

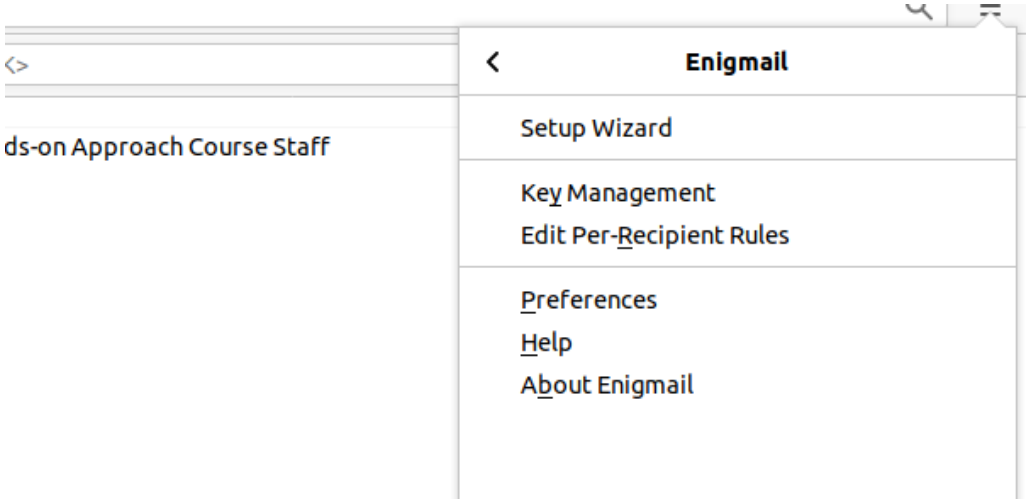
Basta con navegar a la página de extensiones para Thunderbird y descargarla directamente:



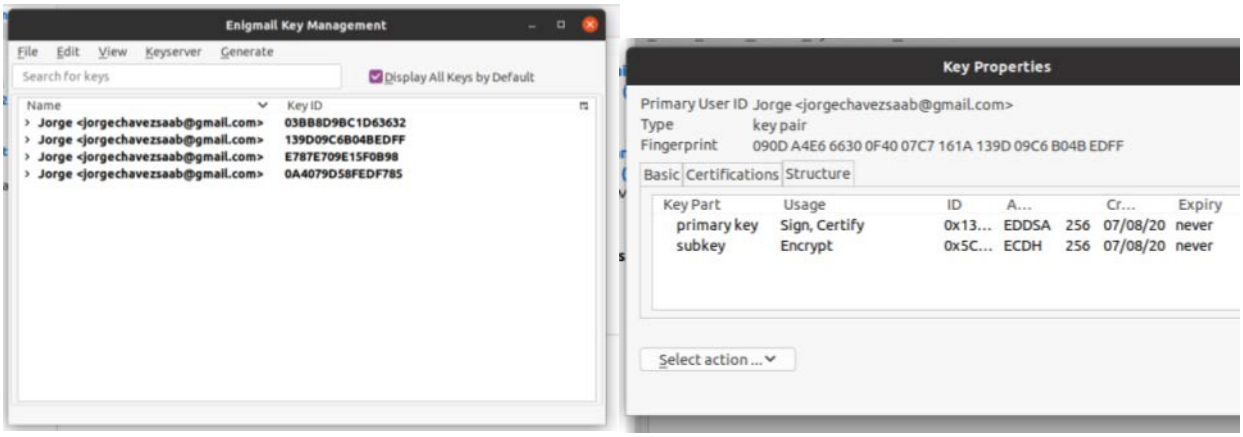
Con solo hacer click en el botón al extensión se agrega y detecta que ya tenemos GnuPG instalado y que ya hemos creado algunas llaves, las cuales podemos agregar automáticamente.



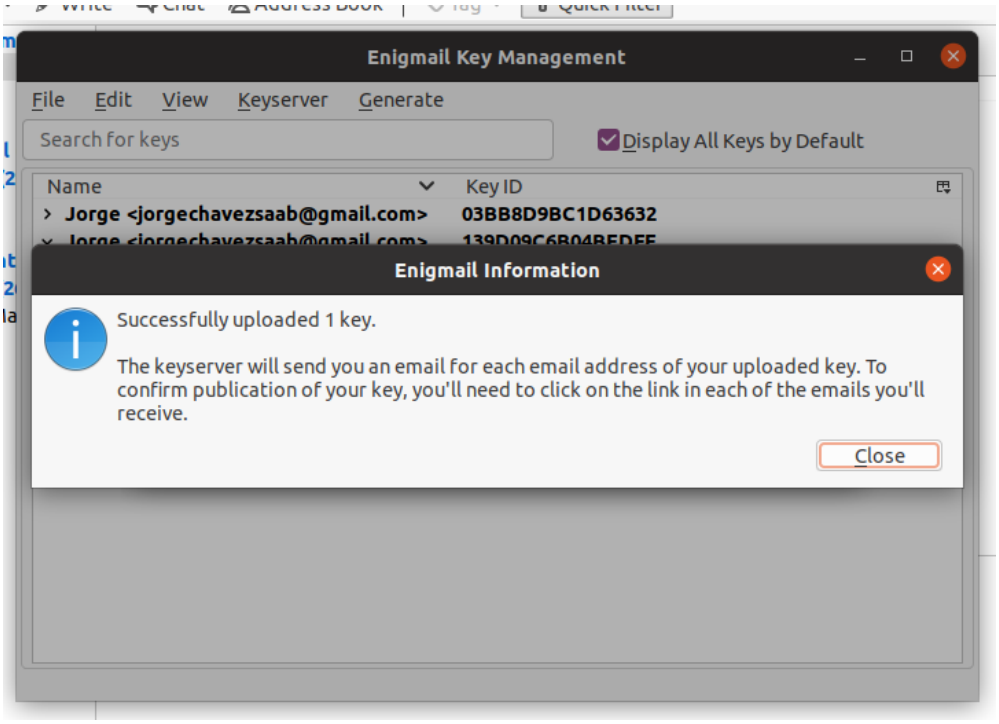
Ahora el menú de opciones de Thunderbird tiene una sección nueva para Enigmail, donde me tomé un tiempo para explorar los diferentes submenús.



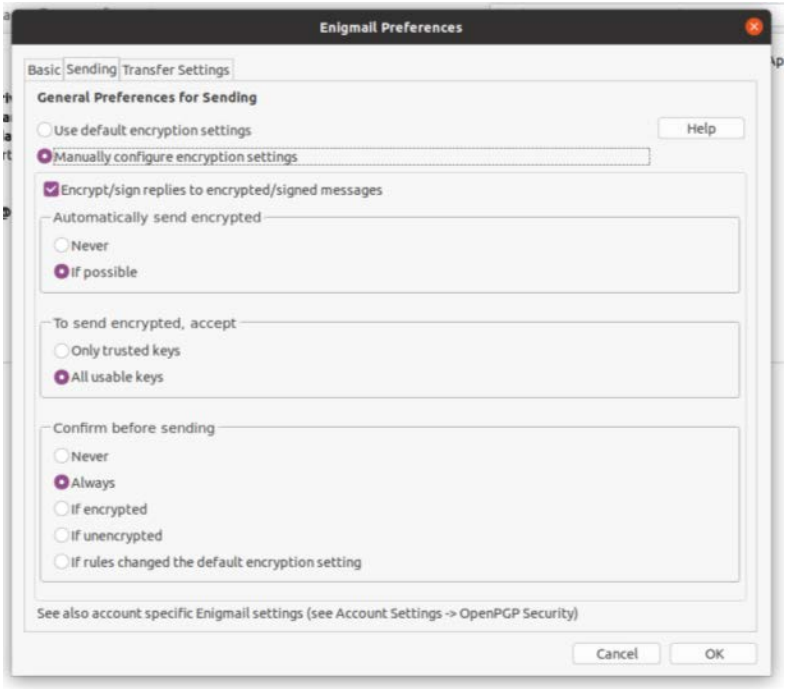
En Key Management encontré listadas todas las llaves que había creado, y al abrirlas puedo ver para qué algoritmo están diseñadas:



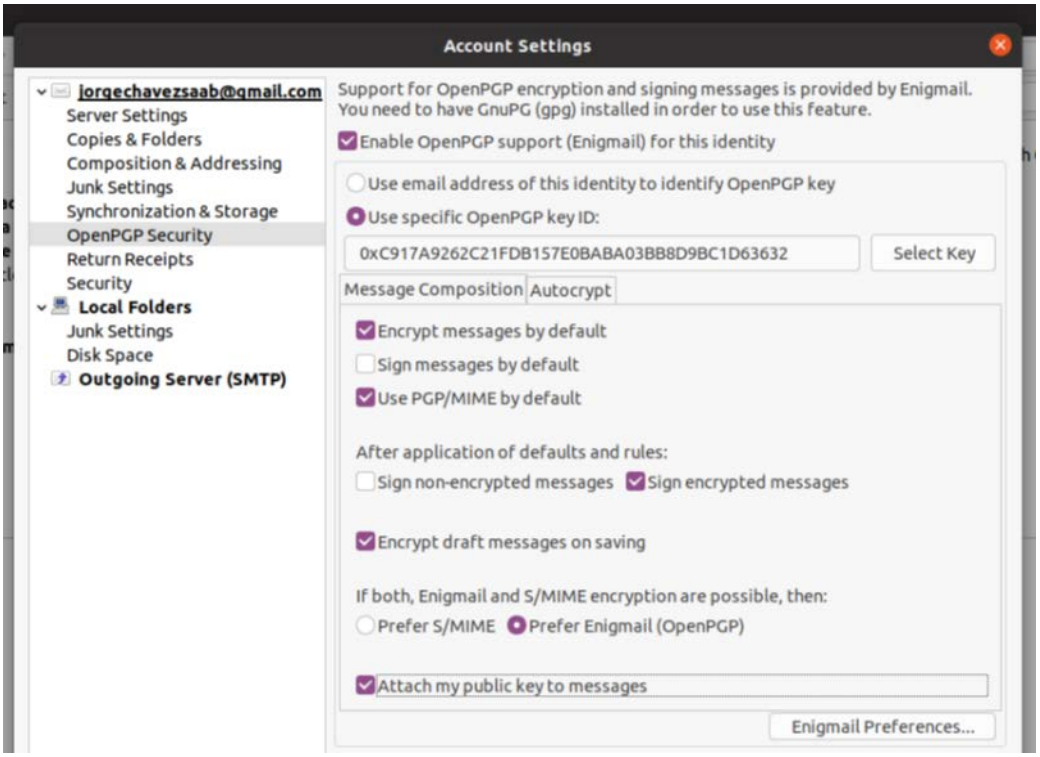
También hay una pestaña llamada Keyserver donde podemos buscar las llaves de otros usuarios por correo electrónico. Asi mismo podemos subir nuestra llave al servidor para que otros usuarios la encuentren, para lo cual se nos envía un correo electrónico para confirmar propiedad de él.



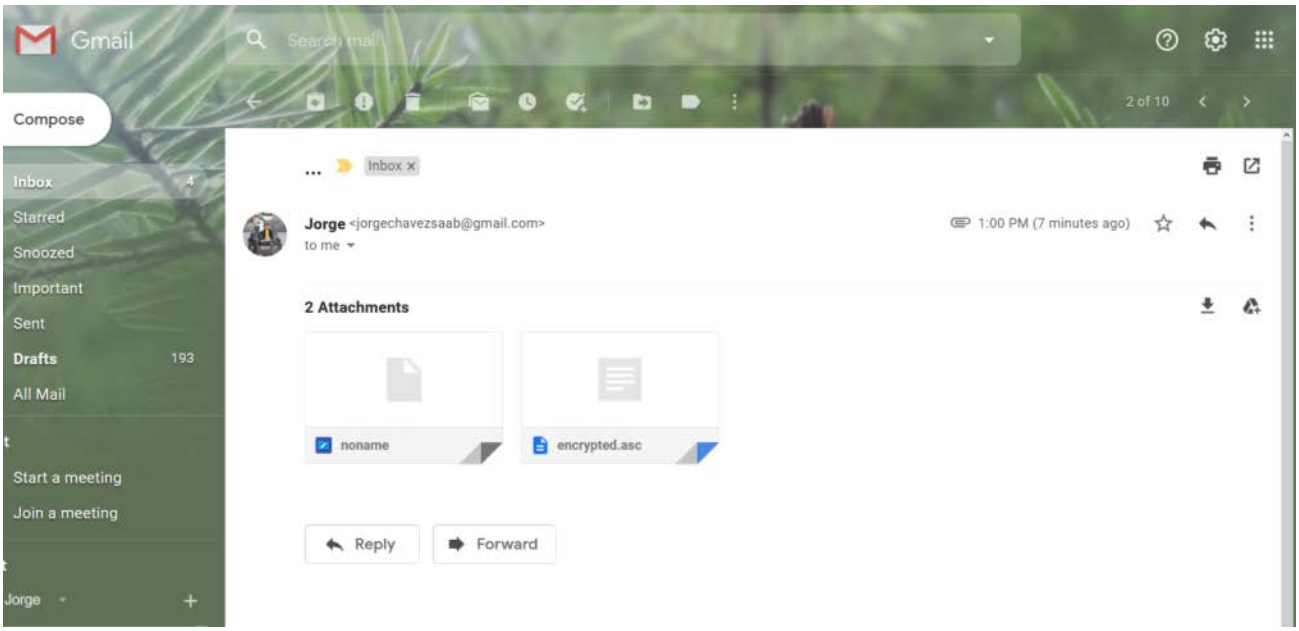
Yendo ahora al menú de `Preferencias`, sólo encontré las opciones para habilitar la firma y cifrado de mensajes al responder a mensajes que estaban firmados o cifrados.



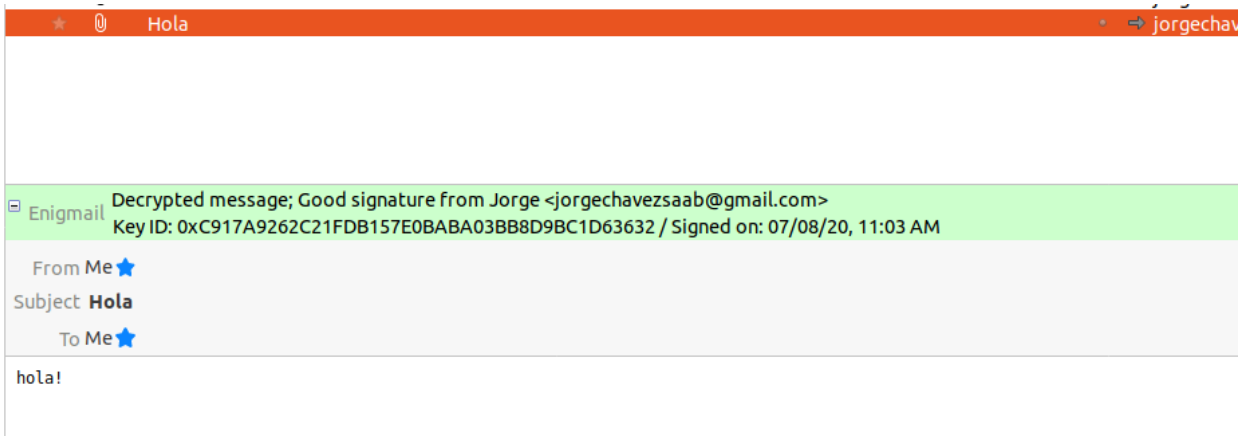
El diálogo también sugiere revisar las opciones en Account Settings > OpenPGP Security, que es donde encontré las opciones para habitar el cifrado y firmado en todos los mensajes salientes las cuales tuve que habilitar.



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.