careful, lots of gpg instead of gpg2 (use gpg2)

https://www.digitalocean.com/community/tutorials/how-to-use-gpg-to-encrypt-and-sign-messages

https://fedoraproject.org/wiki/Creating\_GPG\_Keys#:~:text=To%20create%20a%20key%2C%20go,are%20(e.g.%3A%20John%20C.

# To generate a key:

> gpg2 --full-gen-key

# To publish a key:

to publish the key,

> gpg2 --send-key Key-ID|fingerprint

> gpg2 --keyserver hkp://pgp.mit.edu --send-key Key-ID|fingerprint

The first command did not work, it gave errors, the second one did.

# Revoking Key

They say this should be setup in advance and held on a usb locked up somewhere … so that the key owner can cancel the key if something happens to the private key.

> gpg2 --output ~/revocation.crt --gen-revoke your\_email@address.com

> chmod 600 ~/revocation.crt

# Importing a key

To install a key from a file:

> gpg2 --import name\_of\_pub\_key\_file

To install a key from the mit server:

> gpg2 --keyserver pgp.mit.edu --search-keys <search\_parameters>  
 for example, put the email address for the <search\_parameters> e.g. example@company.com

> gpg2 --keyserver pgp.mit.edu --recv-keys <key-id>

for example, gpg2 --keyserver pgp.mit.edu --recv-keys 00760419BC8C7B9B

# Encrypt File

Once a remote users key has been installed, one may encrypt a message for the remote user:

> gpg2 --encrypt --sign --armor -r remoteuser@email.com file  
  
This produces name\_of\_file.asc. Use multiple -r options so that any one of the corresponding private keys may be used to decrypt the file.

Be sure to add your own key so that you can read the file also:

> gpg2 --encrypt --sign --armor -r remoteuser@email.com -r me@email.com file

# Decrypt File

To decrypt the name\_of\_file.asc:

> gpg2 --output name\_of\_file --decrypt name\_of\_file.asc

the output option must be given first

or

> gpg2 --decrypt encrypted\_file.gpg > output\_file\_name

# Sign the Key

After you, the receiver of the public key, import a key, check the fingerprint and sign it. Do this or gpg will plague you forever with warning messages.

Get a copy of the fingerprint from the user via email and have them read it to you on the phone to check what you received matches what you generate locally. Generally, to get a fingerprint:

> gpg2 --fingerprint <email@address.com for key>

After checking the fingerprint, sign the key

> gpg2 --sign-key email@example.com

Share the love, this creates the temporary file ~/signed.key, then exports it

> gpg2 --output ~/signed.key --export --armor email@example.com

The key issuer should then import the signed key

# Nautilus Support

do not do this, as of 2020-05-12 it does not really work:

X dnf install seahorse-nautilus

do not do this or emacs will not open gpg/pgp anymore

X gio mime application/pgp-encrypted seahorse-pgp-encrypted.desktop

Do this:

Add script ~/.local/share/nautilus/scripts/Decrypt and ~/.local/share/nautilus/scripts/Encrypt

see the versions in my user directory. I should make a project for Fedora setup scripts and put it up on github ..

Note this fine command for getting the mime type of a file:

> file -b --mime-type note.txt.pgp

Note the scripts must be executable for the user.