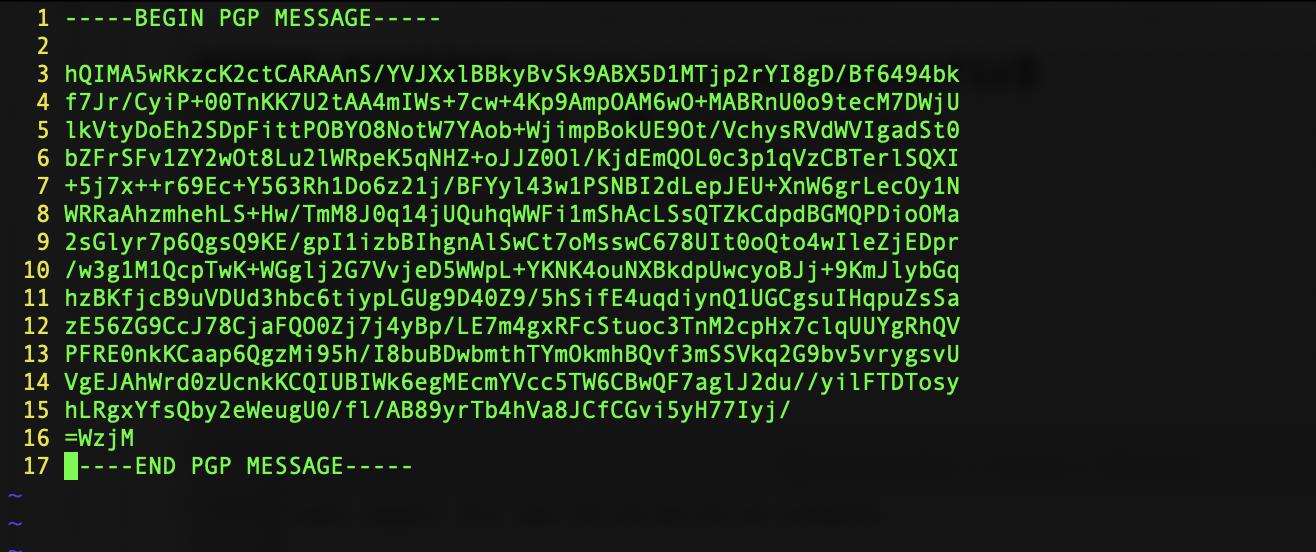
# **PGP key pair generation and encryption and decryption examples in Python 3**

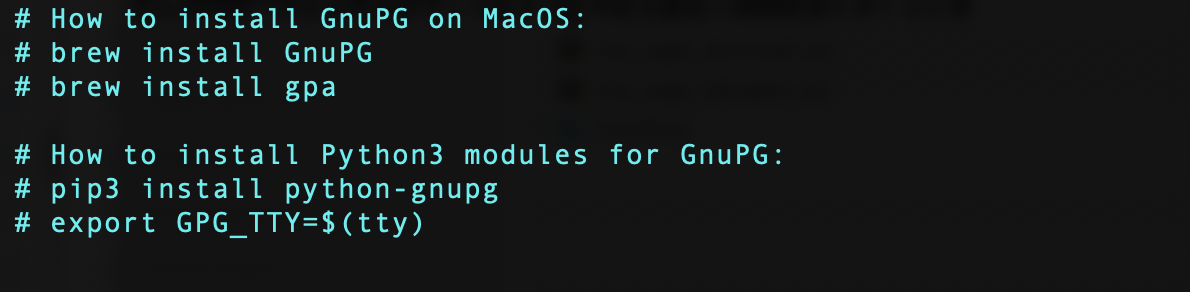
****

Let’s go through an example of PGP key pair generation and using the keys for encryption and decryption.

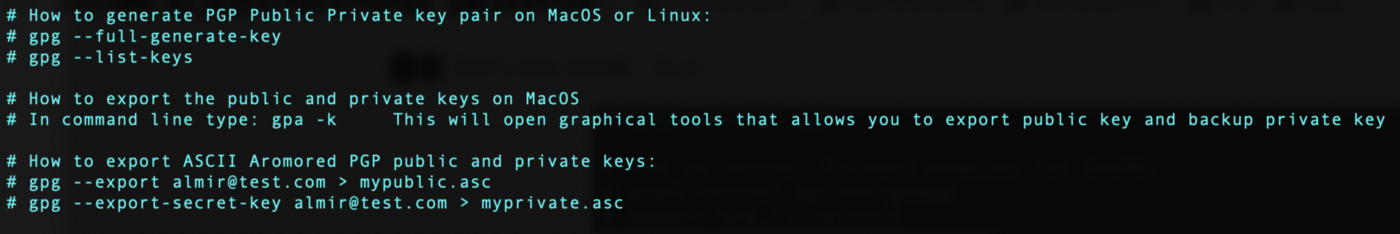
The PGP key generation is different on MacOS compared to Windows. In order to generate the PGP key pair on MacOS, you need to have the following installed:

* GnuPG
* GPA
* python-gnupg module if you plan to generate the keys using Python 3

Here is how you would install this on MacOS:



Now that you have your MacOS set up, here is how you would generate the keys using the GnuPG command line tool:

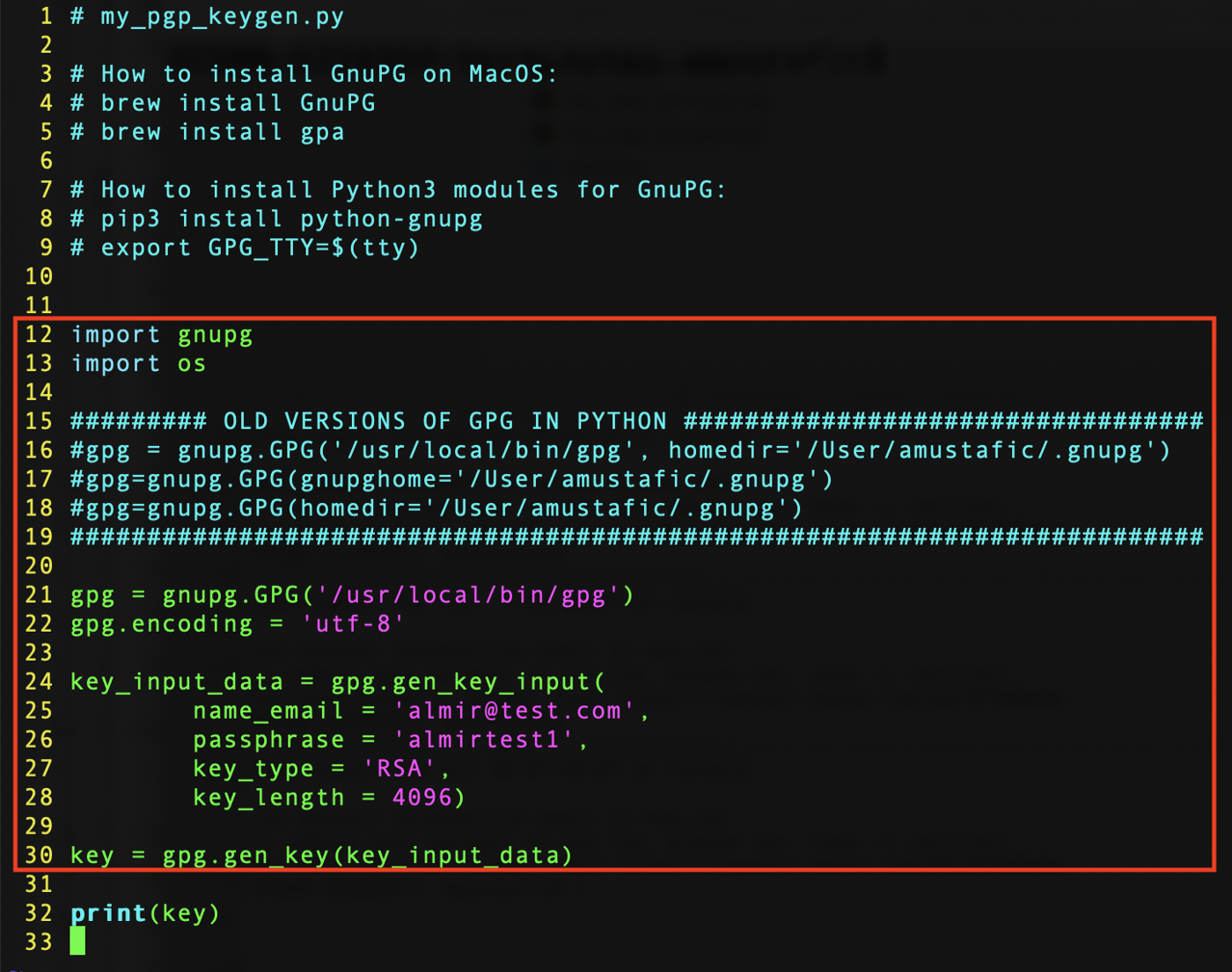


# **In Python 3:**

If you want to generate the PGP key into the key store using Python 3.9.3+, you would first need to pip install GnuPG module for Python 3:

pip3 install python-gnupg

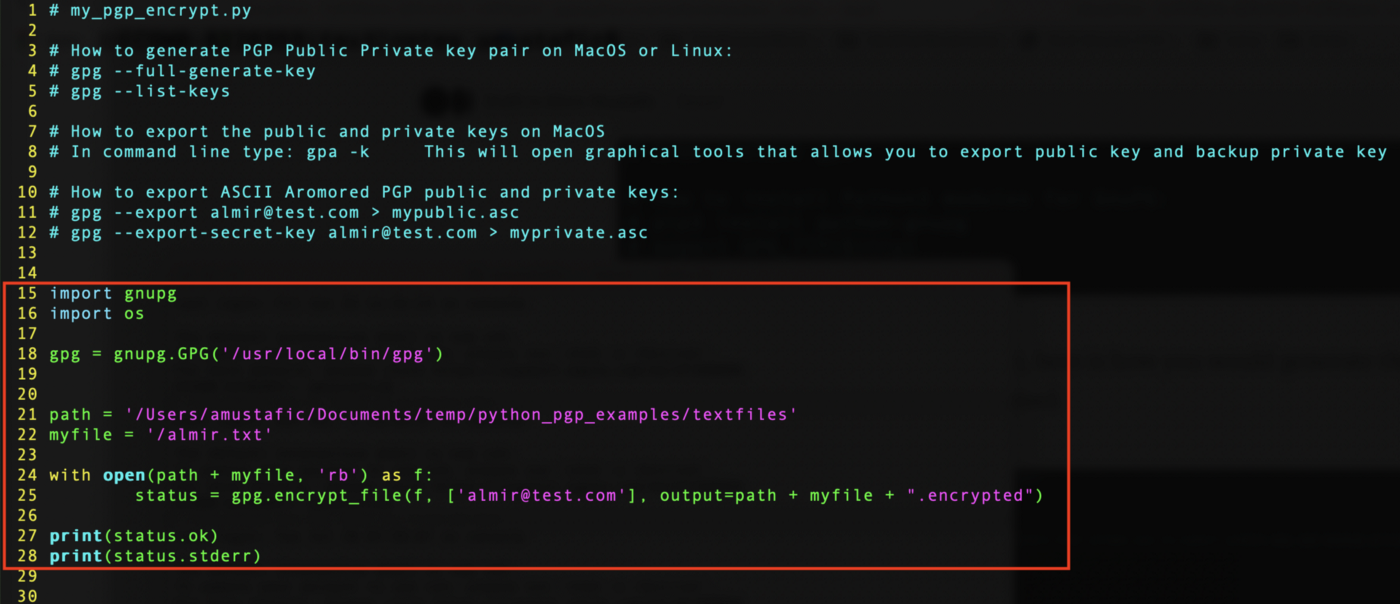
Here is the Python 3 code to generate the PGP key in the key store:



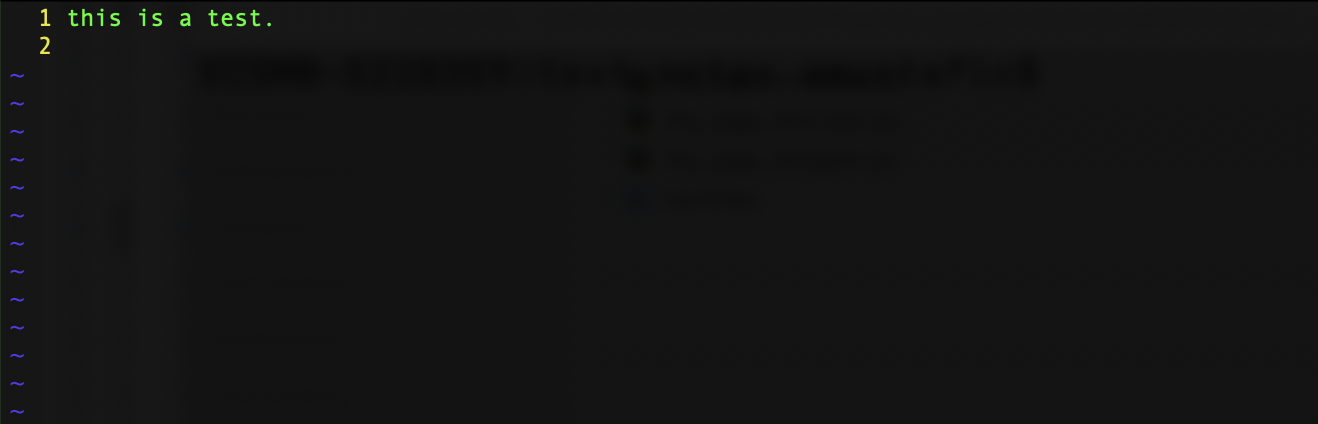
This code automatically puts the the PGP key into the key store. The email address is your unique identifier for accessing the key from the key store. As you will see in the examples below, in order to encrypt a file we will use the **public** key portion from the key store; however, in order to decrypt that same file, we will use the **private** key portion with the **passphrase.**

# **Encryption:**

Let’s now zoom in on how to encrypt the contents of a text file using Python 3 and PGP public key portion from the key store:



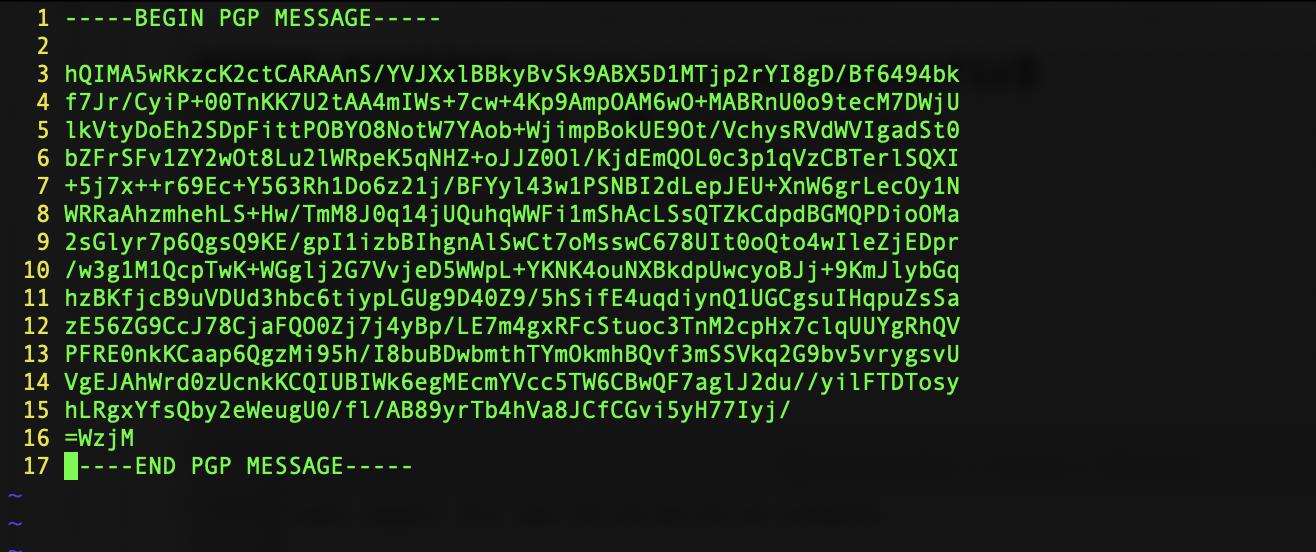
Here are the contents of ./textfiles/almir.txt file:



After we execute the Python script:

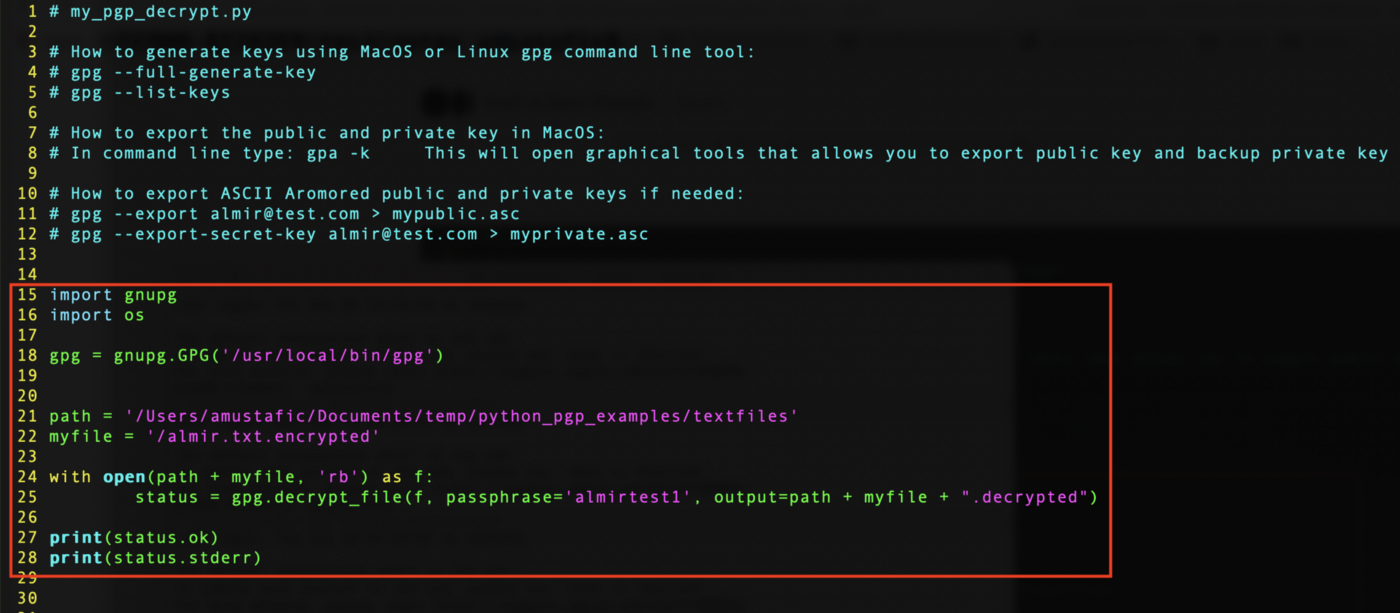
$ python3 my\_pgp\_encrypt.py

The encrypted contents of the input file get stored into ./textfiles/almir.txt.encrypted file:



# **Decryption:**

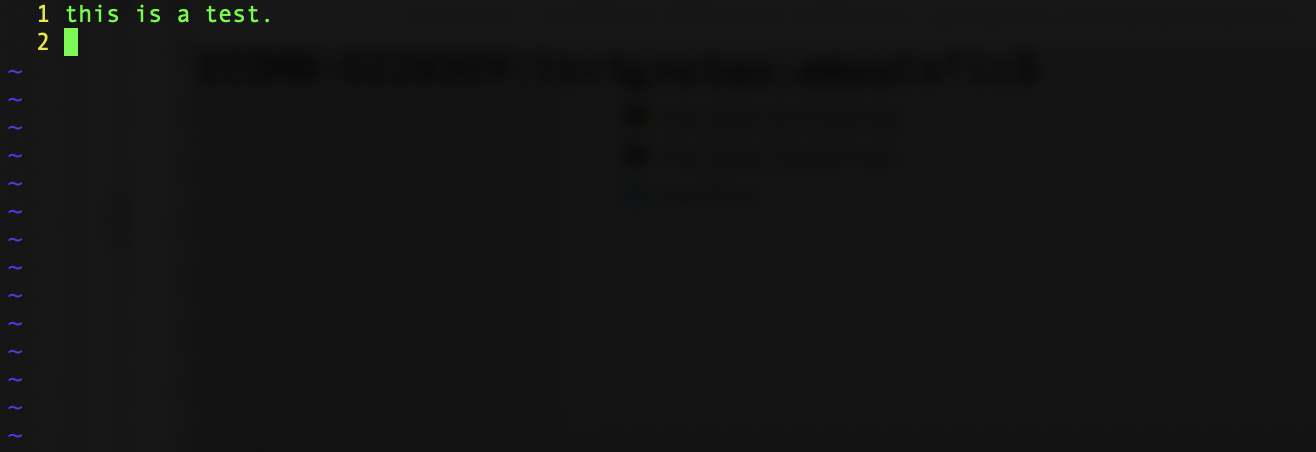
In order to decrypt the contents of this file, we need to use the PGP **private** key from the key store and the **passphrase** is needed when using the private key. Here is the Python code for that:



Let’s execute the decryption code:

$ python3 my\_pgp\_decrypt.py

As per our Python code, the file with the decrypted content is: ./textfiles/almir.txt.encrytped.decrypted



Now it’s time to modularize this code and wrap it into a common library that I can easily use in the future.

# **On Windows?**

I have not tried the Python code on Windows 10, but I did play around with the PGP key pair generations and the exporting of the public and private keys:

* Download Kleopatra tool: <https://www.gpg4win.org/thanks-for-download.html>
* Then use the Kleopatra graphical interface for generating key pair and exporting public and private keys.

Thank you for reading this article. Keep geeking out !!

Almir Mustafic