



Misha Sv

Sep 2, 2020 · 4 min read · Listen



# Encrypt and Decrypt Files using Python — Python Programming

In this article we will discuss how to encrypt and decrypt files using Python.



Photo by [Markus Spiske](#) on [Unsplash](#)

## Table of Contents

- Introduction
- Creating a key
- Loading a key
- Encrypting a file
- Decrypting a file
- Complete Object-Oriented Programming Example
- Conclusion

### How to Encrypt and Decrypt Files using Python

• • •

## Introduction

Today we observe data from all different sources, and ultimately the challenge becomes its security and privacy. That involves not only major datasets held by businesses, but also the AI/ML models and algorithms, and ultimately the data with forecasts.

The amount of data is growing as the number of industries adapting data science practices in their decision making grows. There have been several companies found in the field of encryption and security for AI/ML and cloud computing, which are now even allowing computations on encrypted data.

In the evolving world of data and information transfer, security of the file contents remain to be one of the greatest concerns for companies. Some information can be password protected (emails, logins) while other information being transferred via emails or FTP lacks efficiency if protected by some keyword. This is where file encryption plays a big role and provides security and convenience sought by parties engaged in file transfers.

So what is encryption? It is a process of converting information into some form of a code to hide its true content. The only way to access the file information then is to decrypt it. The process of encryption/decryption is called cryptography.

Let’s see how we can encrypt and decrypt some of our files using Python. We will follow symmetric encryption which means using the same key to encrypt and decrypt the files.

To continue following this tutorial we will need the following Python library: cryptography.

If you don’t have them installed, please open “Command Prompt” (on Windows) and install them using the following code:

```
pip install cryptography
```

And we will also need a sample file we will be working with. Below is the sample .csv file with some data on students’ grades:

<https://pyshark.com/wp-content/uploads/2020/08/grades.csv>

. . .

### Creating a Key

In our example we will be using symmetric equation:

```
1 from cryptography.fernet import Fernet
```

encrypt1.py hosted with ❤ by GitHub [view raw](#)

Fernet is authenticated cryptography which doesn’t allow to read and/or modify the file without a “key”.

Now, let’s create the key and save it in the same folder as our data file:

```
1 key = Fernet.generate_key()
2
3 with open('mykey.key', 'wb') as mykey:
4     mykey.write(key)
```

encrypt2.py hosted with ❤ by GitHub [view raw](#)

If you check the directory where you Python code is located, you should see the **mykey.key** file. You can open it with any text editor (in my case it shows up in the local directory because I use VS Code). The file should contain one line which is a string of some order of characters. For me it is “VlD8h2tEiJkQpKKnDNKnu8ya2fpIBMOo5oc7JKNasvk=”.


. . .

### Loading a Key

After we generated the encryption key, we would need to load it into our environment in order to encrypt/decrypt the files.

The following step is very simple, and requires to just open the mykey.key file and store it in local memory:

```
1 with open('mykey.key', 'rb') as mykey:
2     key = mykey.read()
3
4 print(key)
```

encrypt3.py hosted with  by GitHub

view raw

And just to verify, we will see the following output:

VlD8h2tEiJkQpKKnDNKnu8ya2fpIBMOo5oc7JKNasvk=


The encryption key is now stored locally as the **key** variable.

. . .

### Encrypting a File

Now that we have the file to encrypt and the encryption key, we will now write a function to utilize these and return the encrypted file:

```
1 f = Fernet(key)
2
3 with open('grades.csv', 'rb') as original_file:
4     original = original_file.read()
5
6 encrypted = f.encrypt(original)
7
8 with open ('enc_grades.csv', 'wb') as encrypted_file:
9     encrypted_file.write(encrypted)
```

encrypt3.py hosted with  by GitHub

view raw

Let’s discuss what we did here:

- We initialize the Fernet object as store is as a local variable **f**
- Next, we read our original data (grades.csv file) into **original**
- Then we encrypt the data using the Fernet object and store it as **encrypted**
- And finally, we write it into a new .csv file called “enc\_grades.csv”

You can take a look at the encrypted file here:

[https://pyshark.com/wp-content/uploads/2020/09/enc\\_grades.csv](https://pyshark.com/wp-content/uploads/2020/09/enc_grades.csv)


. . .

### Decrypting a File

After you encrypted the file and, for example, successfully transferred the file to another location, you will want to access it. Now, that data is in the encrypted format. The next step is to decrypt it back to the original content.

The process we will follow now is the reverse of the encryption in the previous part. Exactly the same process, but now we will go from encrypted file to decrypted file:

```
1 f = Fernet(key)
2
3 with open('enc_grades.csv', 'rb') as encrypted_file:
4     encrypted = encrypted_file.read()
5
6 decrypted = f.decrypt(encrypted)
7
8 with open('dec_grades.csv', 'wb') as decrypted_file:
9     decrypted_file.write(decrypted)
```

encrypt4.py hosted with  by GitHub

view raw

Let’s discuss what we did here:

- We initialize the Fernet object as store is as a local variable **f**
- Next, we read our encrypted data (enc\_grades.csv file) into **encrypted**
- Then we decrypt the data using the Fernet object and store it as **decrypted**
- And finally, we write it into a new .csv file called “dec\_grades.csv”

You can take a look at the decrypted file here:

Get unlimited access

Search



Misha Sv

169 Followers

Data Scientist, Python Blogger,  
YouTuber,  
<https://www.linkedin.com/in/mikhail-sidyakov/>

Follow



#### More from Medium

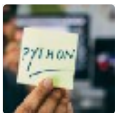
Mzohaib

Lambda  
Functions in  
Python



Fernando Barbo...

Python bits:  
dataclasses—  
Part 1



Kazi Mushfiquir ...

Python Functions  
Exercise



Alexander Doyle

Flying Tello EDU  
Drones in Python





[https://pyshark.com/wp-content/uploads/2020/09/dec\\_grades.csv](https://pyshark.com/wp-content/uploads/2020/09/dec_grades.csv)

Comparing “dec\_grades.csv” with the original “grades.csv”, you will see that in fact these two have identical contents. Our encryption/decryption process was successful.

. . .

### Complete Object-Oriented Programming Example

This is a bonus part where I organized everything in a more structured format:

```
1  class Encryptor():
2
3      def key_create(self):
4          key = Fernet.generate_key()
5          return key
6
7      def key_write(self, key, key_name):
8          with open(key_name, 'wb') as mykey:
9              mykey.write(key)
10
11     def key_load(self, key_name):
12         with open(key_name, 'rb') as mykey:
13             key = mykey.read()
14         return key
15
16
17     def file_encrypt(self, key, original_file, encrypted_file):
18
19         f = Fernet(key)
20
21         with open(original_file, 'rb') as file:
22             original = file.read()
23
24         encrypted = f.encrypt(original)
25
26         with open (encrypted_file, 'wb') as file:
27             file.write(encrypted)
28
29     def file_decrypt(self, key, encrypted_file, decrypted_file):
30
31         f = Fernet(key)
32
33         with open(encrypted_file, 'rb') as file:
34             encrypted = file.read()
35
36         decrypted = f.decrypt(encrypted)
37
38         with open(decrypted_file, 'wb') as file:
39             file.write(decrypted)
```

encrypt5.py hosted with ❤ by GitHub [view raw](#)

[Help](#) [Status](#) [Writers](#) [Blog](#) [Careers](#)  
[Privacy](#) [Terms](#) [About](#) [Knowable](#)

And this is an example of encryption/decryption using the above class:

```
1  encryptor=Encryptor()
2
3  mykey=encryptor.key_create()
4
5  encryptor.key_write(mykey, 'mykey.key')
6
7  loaded_key=encryptor.key_load('mykey.key')
8
9  encryptor.file_encrypt(loaded_key, 'grades.csv', 'enc_grades.csv')
10
11 encryptor.file_decrypt(loaded_key, 'enc_grades.csv', 'dec_grades.csv')
```

encrypt6.py hosted with ❤ by GitHub [view raw](#)

. . .

### Conclusion

This article introduces basic symmetric file encryption and decryption using Python. We have discussed some parts of cryptography library as well as created a full process example.

Feel free to leave comments below if you have any questions or have suggestions for some edits and check out more of my [Python Programming](#) articles.

. . .

Originally published at <https://pyshark.com> on September 1, 2020.

## Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. [Take a look.](#)

 Get this newsletter

Emails will be sent to [pankaj33199@gmail.com](mailto:pankaj33199@gmail.com).  
[Not you?](#)