ШИФРОВАНИЕ ФАЙЛОВ И ДИРЕКТОРИЙ С ПОМОЩЬЮ МОДУЛЕЙ И БИБЛИОТЕК PYTHON

Ахапкина А. М.
Кафедра вычислительных методов и программирования, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: anastasia.akhapkina2018@gmail.com

В этой статье приводятся функции шифрования и дешифрования файлов и директорий компьютера, которые реализованы с помощью специализированных библиотек и содержащихся в них модулей на языке программирования Python. После каждой функции приведен результат выполнения операции в виде скриншотов "До-после".

Введение

В настоящее время большинство коммуникаций, которые мы проводим на личном и деловом уровнях, осуществляются онлайн с помощью смартфонов, планшетов и компьютеров. В связи с чем существует множество опасностей для пользователей данных, будь то взлом или потеря телефона, кража личных архивов, фотографий, номеров банковских карт или какой-либо другой интеллектуальной собственности. Шифрование файлов и жестких дисков помогает пользователям защитить данные и их конфиденциальность во время передачи сообщений или файлов, или при их хранении на устройстве. Шифрование дает уверенность в том, что данные не будут перехвачены при их передаче, а сохраненные файлы не попадут в чужие руки, даже если устройство, на котором они хранятся, потеряно или украдено.

I. Шифрование на Python

Python-высокоуровневый язык программирование. Большинство задач решается за счет встроенных модулей библиотек. Однако, если стоит задача, связанная с криптографическими методами, то Python предоставляет нам множество библиотек, которые в той или иной степени позволяют решать криптографические задачи. В нашем случаи, для шифрования файлов и директорий на языке программирования Python воспользуемся библиотеки pyAesCrypt. За счет встроенного в данную библиотеку метода pyAesCrypt.encryptFile() можно реализовать функцию, которая будет шифровать файлы:

```
def encryption(file, password):

    buffer_size=512*1024
    pyAesCrypt.encryptFile(
        str(file),
        str(file)+".crp",
        password,
        buffer_size
    )

    print("[Файл '"+str(os.path.splitext(file)[0])+"' зашифрован]")
    os.remove(file)
```

Рис. 1 – Реализация функции encryption

В качестве параметров функция encryption() принимает файл и пароль.

Для шифрования директорий необходимо реализовать функцию, которая будет сканировать все папки и файлы и шифровать их. Для написания данной функцию необходимо подключить модуль os(operationn system). Функция может иметь следующую реализацию:

```
def walking_by_dirs(dir, password):

    #перебираем все поддиректории в указанной директории
    for name in os.listdir(dir):
        path=os.path.join(dir,name)

        #если находим файл, то шифруем его
        if os.path.isfile(path):
            try:
                decryption(path,password)
            except Exception as ex:
                print(ex)
        #если находим директорию, то повторяем цикл в поисках файлов
        else:
            walking_by_dirs(path,password)
```

Рис. 2 – Реализация функции для сканирования всех директорий и папок

Результат шифрования:

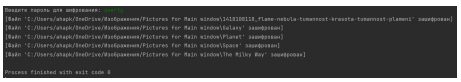

Рис. 3 – Сообщение об удачном шифровании файлов

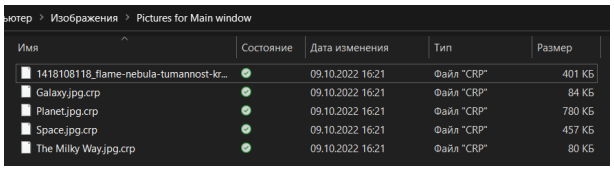

Рис. 4 – Результат шифрования файлов

Как видно из рисунков, изначально в одной папке размещалось 5 изображений. После компиляции нам вывелось сообщения об удачном шифровании каждой отдельной фотографии. Если внимательно посмотреть на рисунок 2, можно заметить, что у фотографий появилось расширение .crp, что указывает, что фотографии зашифрованы и мы не сможем их посмотреть. Стоит отметить, что это создались новые файлы, в то время как первоначальные изображения были удалены.

Если файлы для шифрования находятся на виртуальной машине, то после шифрования надо удалить за собой скрипты. Это можно сделать импортировав модуль sys и вызвать метод remove():

os.remove(str(sys.argv[o]))

II. Дешифрование на Python

При шифровании файлов и директорий рано или поздно настанет момент, когда нам понадобиться их расшифровать. Поэтому при написании программы по шифрованию файлов, надо сразу же позаботиться об их дешифровании. Для шифрования необходимо пользоваться методами библиотеки, с помощью которой мы шифровали файлы, так как у разных библиотек методы и ключи шифрования отличаются.

Процесс дешифрования схож с шифрованием, отличается лишь методом decryption():

```
def decryption(file, password):

    buffer_size=512*1024

    pyAesCrypt.decryptFile(
        str(file),
        str(os.path.splitext(file)[0]),
        password,
        buffer_size
    )

    print("[Файл '"+str(os.path.splitext(file)[0])+"' дешифрован]")
    os.remove(file)
```

Рис. 5 – Реализация функции decryption()

Результат дешифрования файлов:

```
Введите пароль: ******
[Файл 'C:/Users/ahapk/OneDrive/Изображения/Pictures for Main window\1418108118_flame-nebula-tumannost-krasota-tumannost-plameni.jpg' дешифрован]
[Файл 'C:/Users/ahapk/OneDrive/Изображения/Pictures for Main window\Galaxy.jpg' дешифрован]
[Файл 'C:/Users/ahapk/OneDrive/Изображения/Pictures for Main window\Planet.jpg' дешифрован]
[Файл 'C:/Users/ahapk/OneDrive/Изображения/Pictures for Main window\Space.jpg' дешифрован]
[Файл 'C:/Users/ahapk/OneDrive/Изображения/Pictures for Main window\The Milky Way.jpg' дешифрован]

Process finished with exit code 0
```

Рис. 6 – Сообщение об удачной дешифровании файлов

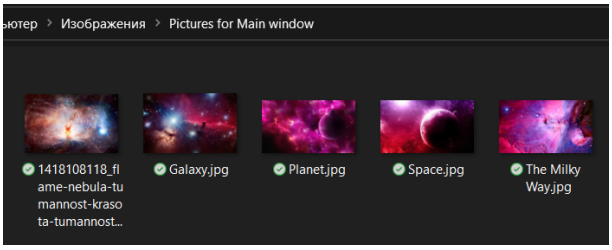

Рис. 7 – Результат дешифрования файлов

Как видно из рисунков, после компиляции мы получили сообщение об удачной дешифровании фотографий. У всех 5-ти фотографий отсутствует расширение .crp, что даем нам возможность их открыть и просмотреть.

Если бы в данной директории находились одна или несколько других папок или файлы с другим расширением, то компилятор бы зашифровал/дешифровал как сами папки, так и файлы, находящиеся в них. В нашем случаи был рассмотрен пример с шифрованием изображений, располагающихся в одной папке.

III. Заключение

Python- высокоуровневый язык программирования, который позволяет осуществлять методы криптографического преобразования информации. Для чего достаточно установить специализированные минформации осуществляется это в строенной IDE Python. В нашем случаи для шифрования файлов и директорий компьютера достаточно было подключить библиотеку pyAesCry, с помощью модулей и методов которой были реализовани шифрование и дешифрование изображений.

IV. Список литературы

1. Криптография и взлом шифров на Python / Эл Свейгарт
2. Реализация и дешифрования шифров Виженера и Цезаря в Python / Ф. В.Черджиев. – Мск.: Инкарт, 2017. – 50 с.
3. Ефипов, И. М. Таинственные страницы занимательной криптографии // Вест. аритмол. – 2015. – С. 5–13.
4. Национальный Интернет-портал Республики Беларусь [Электронный ресурс]