|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| **Институт комплексной безопасности и специального приборостроения** | | |
|  |  |  |

Кафедра КБ-4 «Интеллектуальные системы информационной безопасности »

**Отчет**

По проекту:  
Система перекрёстного шифрования информации CrosEnSys

Выполнил:

Студент 3 курса

Группа ББСО-02-19

Шифр 19Б1181

Кузьменко Александр Евгеньевич

Москва, 2021

Название данного проекта — CrosEnSys – означает Cross Encryption System и переводится как система перекрёстного шифрования.

Опишем принцип действия алгоритмов и принцип работы программы по пунктам  
1. Команды

Благодаря меню пользователь может получить список команд, а так же ознакомиться с информацией и инструкцией командами commands, information и instruction соответственно.

if command=="commands":

print("Список команд CrosEnSys:")

print("commands - пулучить список команд")

print("information - получить информацию о программе")

print("instruction - получить инструкцию к программе")

print("encrypt - зашифровать все файлы в указанной папке")

print("decrypt - расшифровать все файлы в указанной папке")

print("evaluate - выполненить однострочную команду в интерпретаторе Python")

print("system - выполненить однострочную команду в системной консоли")

print("getram - получить размер буфера")

print("setram - установить размер буфера")

print("isdir - проверка принадлежности указанного пути к папке")

print("isfile - проверка принадлежности указанного пути к файлу")

print("createfile - создать новый файл")

print("createdir - создать новую папку")

print("exists - проверка папки или файла на существование")

print("countfiles - посчитать, сколько файлов в указанной папке")

print("getfile - получить путь к файлу по его порядковому номеру в указанной папке (индексация от нуля)")

print("clear - очистить файл")

print("delete - удалить папку или файл")

print("size - определить размер папки или файла")

print("getabsolutepath - получить абсолютный путь к папке или файлу")

print("read - прочитать файл")

print("rewrite - перезаписать файл")

print("write - дозаписать в файл")

print("append - дозаписать в файл")

print("rename - переименовать папку или файл")

print("copy - скопировать папку или файл")

print("move - переместить папку или файл")

print("! - игнорировать следующее сообщение")

print("exit - выйти из программы")

print("stop - выйти из программы")

elif command=="information":

print("\*\*\*\*\*\*\* Информация \*\*\*\*\*\*\*")

print(" Версия: 1.0 Release")

print(" Почта: Sasha.kuzmenko.2001@mail.ru")

print(" Задача:")

print(" Основной задачей данного проекта является перекрёстное шифрование и дешифрование файлов в указанной папке.")

elif command=="instruction":

print("Инструкция:")

print(" Необходимо запустить программу и ознакомиться с данной инструкцией, информацией и списком команд, после чего можно работать с программой. Шифровать и дешифровать папку, в которой находится всего лишь один файл - недопустимо! Шифровать и дешифровать пустую папку - бессмысленно!")

2. Интерфейс

Для взаимодействия пользователя с данной программой был написан интерфейс:

def interface(): #здесь находится интерфейс

print("Добро пожаловать в программу перекрёстного шифрования данных CrosEnSys!")

print("Посмотреть список команд можно командой \"commands\"")

isRun = True

command = ""

while isRun:

command = input("\n---> ")

print()

if command=="commands":

3. Шифрование

Шифрование происходит так: в указанной папке берётся файл и к значению каждого его байта прибавляется значение байта из следующего файла, при том позиция считывания из следующего файла зависит от размера предыдущего файла. Если при считывании данных из следующего файла курсор достигает конца файла, то он просто перемещается в начало этого файла. Так же в программе логика для особых случаев следующая: для первого файла предыдущим считается последний файл, а для последнего файла следующим файлом считается первый файл.

RAM = 1024\*1024 #размер буфера в байтах

def encrypt(path):

counter = 0

filecounter = fileio.countfiles(path)

while counter < filecounter:

print("Осталось зашифровать еще: " + str(filecounter-counter) + " файлов!")

lastfilesize = fileio.size(fileio.getFileByID(path, counter-1))

filesize = fileio.size(fileio.getFileByID(path, counter))

nextfilesize = fileio.size(fileio.getFileByID(path, counter+1))

currentfile = open(fileio.getFileByID(path, counter), "r+b")

seekpos = 0

tellpos = 0

nextfile = open(fileio.getFileByID(path, counter+1), "rb")

if lastfilesize>0 and nextfilesize>0:

nextfile.seek(lastfilesize%nextfilesize)

tempsize = filesize

while tempsize>0:

if tempsize>=fileio.RAM:

data1 = currentfile.read(fileio.RAM)

tempsize-=fileio.RAM

else:

data1 = currentfile.read(tempsize)

tempsize-=tempsize

arr = bytearray(data1)

i = 0

if nextfilesize>0:

while i < len(arr):

if (nextfile.tell()==nextfilesize):

nextfile.seek(0)

t = nextfile.read(1)

arr[i] = (arr[i]+t[0])%256

i+=1

tellpos = currentfile.tell()

currentfile.seek(seekpos)

currentfile.write(arr)

currentfile.seek(tellpos)

seekpos = tellpos

currentfile.flush()

currentfile.close()

counter+=1

4. Дешифрование

Дешифрование файлов происходит обратным образом. Вычисления там производятся не от первого файла к последнему, а от последнего файла к первому и значения, конечно же не прибавляются, а отнимаются.

def decrypt(path):

counter = fileio.countfiles(path) - 1

while counter >= 0:

print("Осталось расшифровать еще: " + str(counter+1) + " файлов!")

lastfilesize = fileio.size(fileio.getFileByID(path, counter-1))

filesize = fileio.size(fileio.getFileByID(path, counter))

nextfilesize = fileio.size(fileio.getFileByID(path, counter+1))

currentfile = open(fileio.getFileByID(path, counter), "r+b")

seekpos = 0

tellpos = 0

nextfile = open(fileio.getFileByID(path, counter+1), "rb")

if lastfilesize>0 and nextfilesize>0:

nextfile.seek(lastfilesize%nextfilesize)

tempsize = filesize

while tempsize>0:

if tempsize>=fileio.RAM:

data1 = currentfile.read(fileio.RAM)

tempsize-=fileio.RAM

else:

data1 = currentfile.read(tempsize)

tempsize-=tempsize

arr = bytearray(data1)

i = 0

if nextfilesize>0:

while i < len(arr):

if (nextfile.tell()>=nextfilesize):

nextfile.seek(0)

t = nextfile.read(1)

arr[i] = (arr[i]-t[0])%256

i+=1

tellpos = currentfile.tell()

currentfile.seek(seekpos)

currentfile.write(arr)

currentfile.seek(tellpos)

seekpos = tellpos

currentfile.flush()

currentfile.close()

counter-=1

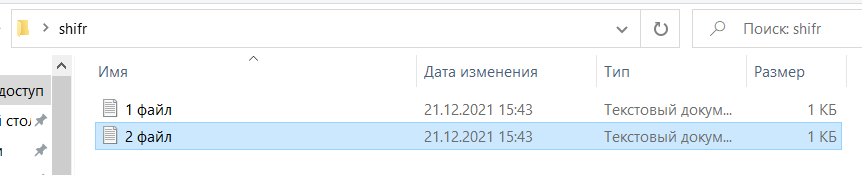
**Пример работы программы.**

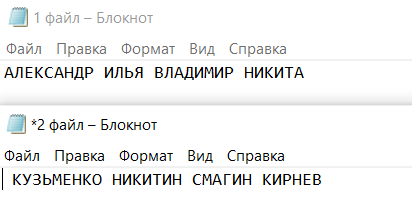
Для работы моей программы необходимо создать папку и в ней 2 файла с информацией.

**1.Создаем папку на рабочем столе и назовем ее «shifr»**



**2.В ней создаем 2 текстовых файла.**

**3.В файлы записываем информацию.**



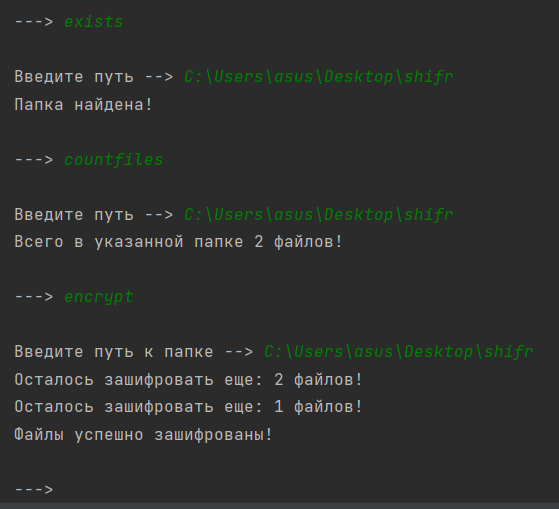
**4.Переходим в программу CrosEnSys и вводим команду ---> commands ---> exists ---> countfiles ---> encrypt.**

1.Commands- список команд.

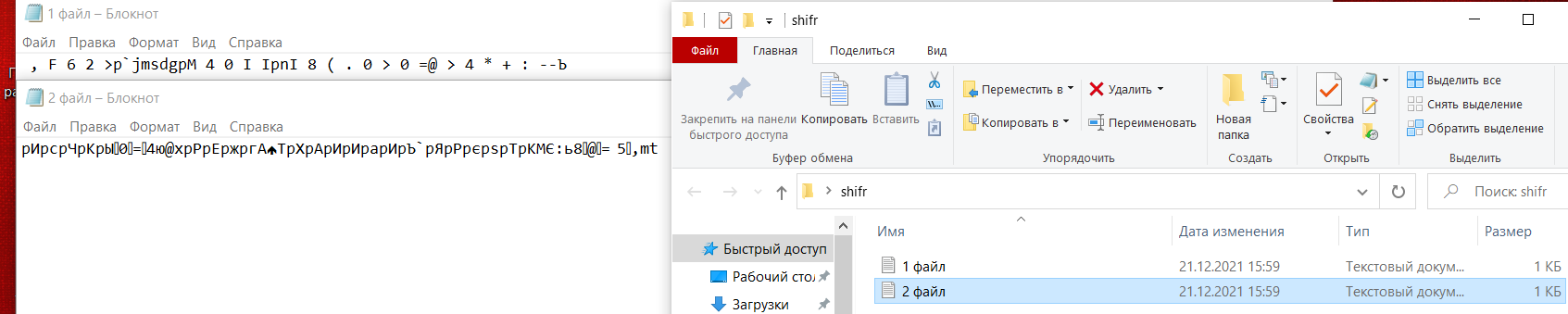
2.exists - проверка папки или файла на существование

3. countfiles - посчитать, сколько файлов в указанной папке

4. encrypt - зашифровать все файлы в указанной папке



**5.Проверям файлы и убеждаемся в их шифровке**.



**6.Дешифруем файлы в первоначальный вид**.

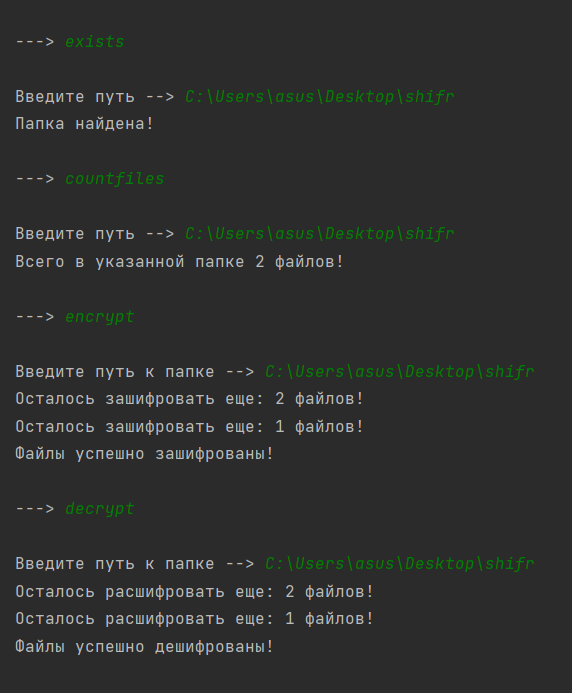
Переходим в программу CrosEnSys и вводим команду ---> commands ---> exists ---> countfiles ---> encrypt.

1.Commands- список команд.

2.exists - проверка папки или файла на существование

3. countfiles - посчитать, сколько файлов в указанной папке

4. decrypt - расшифровать все файлы в указанной папке



**7.Проверям файлы и убеждаемся в их дешифровке**.

