                     Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет ПИиКТ, кафедра ВТ



ОТЧЁТ

По лабораторной работе №1

“Основы шифрования данных”

По предмету: Информационная безопасность

Вариант 3

Студенты:

Степанов Михаил Алексеевич

Группа P34301

Преподаватель:

Маркина Татьяна Анатольевна

Санкт-Петербург

2023

Цель работы

Изучение основных принципов шифрования информации, знакомство с широко известными алгоритмами шифрования, приобретение навыков их программной реализации.

Вариант

Реализовать шифрование и дешифрацию файла с использованием метода биграмм. Ключевое слово вводится.

 Исходный код

import sys

# класс для хранения входной информации

class console\_param:

def \_\_init\_\_(self):

self.encrypt = True

self.in\_filename = ""

self.out\_filename = ""

self.key = ""

# считать из ключа уникальные символы

def reduce\_key(key):

unique = []

for char in key:

if char not in unique:

unique.append(char)

return unique

# прочитать аргументы из командной строки

def fill\_console(args):

console = console\_param()

console.out\_filename = args[-1]

console.in\_filename = args[-2]

console.key = reduce\_key(args[-3])

for param in args[:-3]:

if param == "-d":

console.encrypt = False

return console

def main(args):

console = fill\_console(args)

with open(console.out\_filename, 'w', encoding="utf-8") as fout:

with open(console.in\_filename, 'r', encoding="utf-8") as fin:

crypt(fin, fout, console.encrypt, console.key)

def crypt(fin, fout, flag, key):

for line in fin:

new\_line = ""

if flag:

new\_line = encrypt(line, key)

else:

new\_line = decrypt(line, key)

fout.write(new\_line + '\n')

# подсчитать номер из символа через таблицу

def order(char, key):

if char in key:

return key.index(char)

else:

less\_sum = 0

for c in key:

if ord(c) > ord(char):

less\_sum += 1

return ord(char) + less\_sum

# вычислить символ по номеру через таблицу

def char(code, key):

codes = list(map(ord, key))

if code in codes:

return key[codes.index(code)]

else:

less\_sum = 0

for c in codes:

if c > code:

less\_sum += 1

return chr(code - less\_sum)

def trim\_line(line):

if line[-1] == '\n':

line = line[:-1]

if len(line) % 2 == 1:

line += " "

return line

def make\_pair(line):

pairs = []

for pair in [line[i:i+ 2] for i in range(0, len(line), 2)]:

if pair[0] == pair[1]:

pairs.append(pair[0] + " ")

pairs.append(" " + pair[1])

else:

pairs.append(pair)

return pairs

# зашифровать строку

def encrypt(line, key):

line = trim\_line(line)

pairs = make\_pair(line)

new\_pairs = []

for pair in pairs:

index = [order(char, key) for char in pair]

rows = [int(i / 16) for i in index]

column = [i % 16 for i in index]

if rows[0] == rows[1]:

column = [(column[0] + 1) % 16, (column[1] + 1) % 16]

elif column[0] == column[1]:

rows = [rows[0] + 1, rows[1] + 1]

else:

column = [column[1], column[0]]

new\_pairs.append("".join([char(rows[0] \* 16 + column[0], key), char(rows[1] \* 16 + column[1], key)]))

return "".join(new\_pairs)

# расшифровать строку

def decrypt(line, key):

line = trim\_line(line)

pairs = make\_pair(line)

new\_pairs = []

for pair in pairs:

index = [order(char, key) for char in pair]

rows = [int(i / 16) for i in index]

column = [i % 16 for i in index]

if rows[0] == rows[1]:

column = [(column[0] - 1) % 16, (column[1] - 1) % 16]

elif column[0] == column[1]:

rows = [rows[0] - 1, rows[1] - 1]

else:

column = [column[1], column[0]]

new\_pairs.append("".join([char(rows[0] \* 16 + column[0], key), char(rows[1] \* 16 + column[1], key)]))

return "".join(new\_pairs)

if \_\_name\_\_ == "\_\_main\_\_":

if len(sys.argv) < 4:

print("недостаточно аргументов")

else:

main(sys.argv[1:])

Результат работы программы

Изначальный текст:



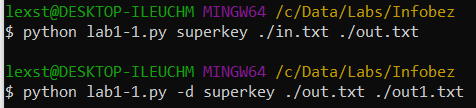
Зашифрованный текст



Расшифрованный текст



Последовательность команд для зашифровки и расшифровки:



Здесь superkey – ключ, а последующие аргументы соответственно входной и выходной файлы. Ключ “-d” используется для расшифровки.