МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ЭКОНОМИКИ И СЕРВИСА

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ОТЧЕТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

по дисциплине «Информационная безопасность и защита информации»

Реализация шифра гаммирования

Студент

гр. БИС–18–01 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Я.И. Илюшин

Старший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Е. Боршевников

Владивосток 2020

**Цель:**

Приобретение практических навыков в реализации простейших шифров.

**Задачи:**

1. Реализовать программно шифр Вернама для английского алфавита (26 символов);
2. Описать код в отчете, а также привести рабочую версию скомпилированной программы (или скрипта).

**Требование к реализации:**

1. Необходимо производить выбор между зашифрованием и расшифрованием.
2. Необходимо реализовать чтение текста и гаммы из файлов.
3. Необходимо, чтобы пробелы распознавались и не шифровались.

**Ход работы:**

**Задача 1 - 2.**

Зашифрование производится посимвольно по следующей формуле (1).

|  |  |
| --- | --- |
|  | (1) |

Где mod – операция взятия остатка от деления на число; |A| – мощность алфавита, количество букв в алфавите; c – номер символа шифртекста в алфавите; m – номер символа открытого текста в алфавите.

Расшифрование производится по следующей формуле (2).

|  |  |
| --- | --- |
|  | (2) |

Реализация выполнена на языке программирования Python в виде консольной программы, где нужно указать аргумент (“e” или “d”), чтобы программа начала выполняться в режиме шифрования или дешифрования, соответственно. Код, отвечающий за проверку и ввод аргумента, а также за считывания из консоли пользовательского ввода и проверку возможности открытия файла (рисунок 1).

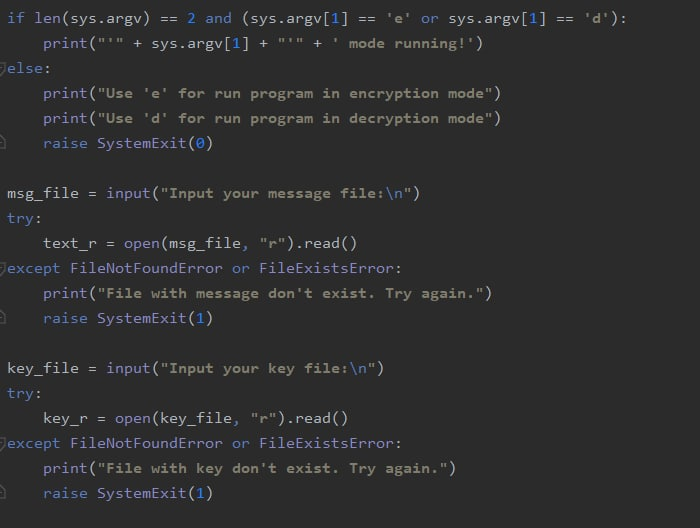


Рисунок 1 – Проверка на корректность ввода аргумента

Сразу после проверки на корректность аргумента выполняется условие, определяющее какой блок кода, соответствующий шифрованию или дешифрованию, соответственно, будет выполняться (рисунок 2).

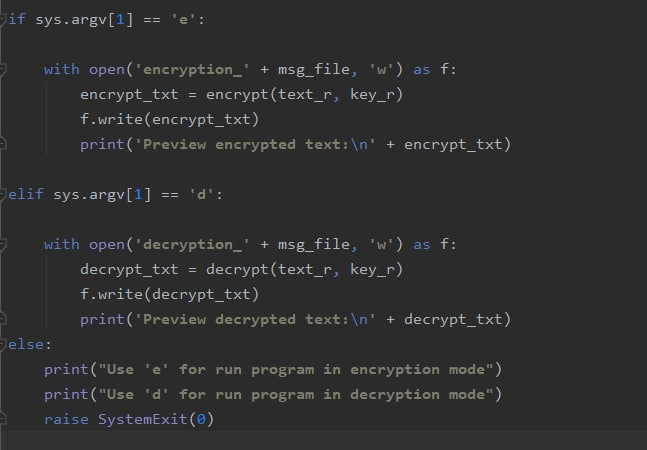


Рисунок 2 – Код, отвечающий за режим шифрования или дешифрования

Если пользователь выбрал режим шифрования или дешифрования, то откроется файл (или создастся, если его нет), в который программа запишет зашифрованный или дешифрованный текст, также выведет результат и в консоль для наглядности.

Также следует упомянуть, что новый файл создастся с конкатенацией «encryption\_» или «decryption\_» в зависимости от режима запущенной программы.

Функция шифрования принимает сообщение и ключ, представляет собой посимвольный перебор текста с использованием алгоритма шифрования. Исключением являются символы конца строки (\n) и пробел, данные символы не шифруются (рисунок 3).

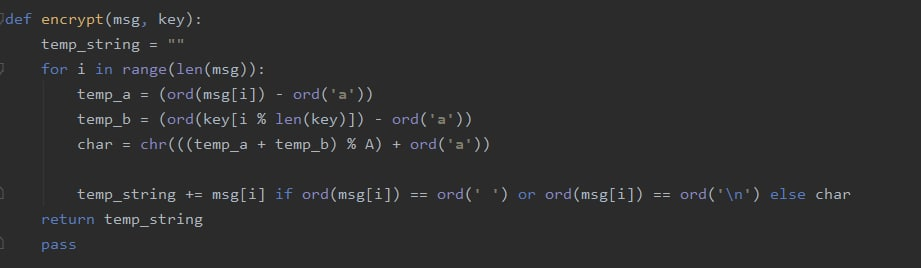


Рисунок 3 – Функция шифрования

Часть алгоритма была разбита на переменные temp\_a и temp\_b для удобства восприятия.

Функция дешифрования принимает сообщение и ключ, представляет собой также посимвольный перебор текста с использованием алгоритма дешифрования. Исключением являются символы конца строки (\n) и пробел, данные символы не дешифруется (рисунок 4). Также алгоритм разбит на временные переменные, как и в случае функции шифрования.

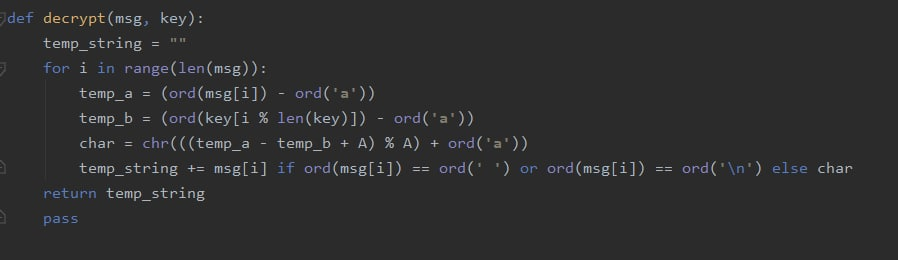


Рисунок 4 – Функция дешифрования

Буквой «А» обозначена переменная, хранящая в себе мощность алфавита (рисунок 5).



Рисунок 5 – Мощность алфавита

Проверим функционал программы. Создадим файл “message.txt”, в котором будет содержаться некоторый текст (рисунок 6).

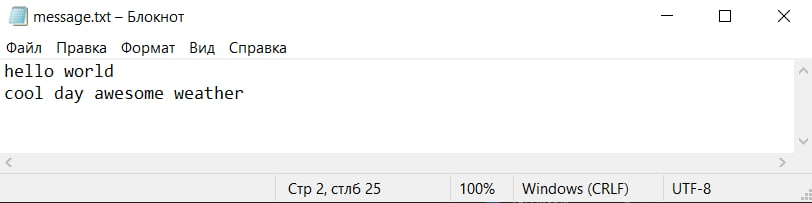


Рисунок 6 – Файл с сообщением

Создадим файл “otp.txt”, который будет содержать текст, который будет являться ключом (рисунок 7).

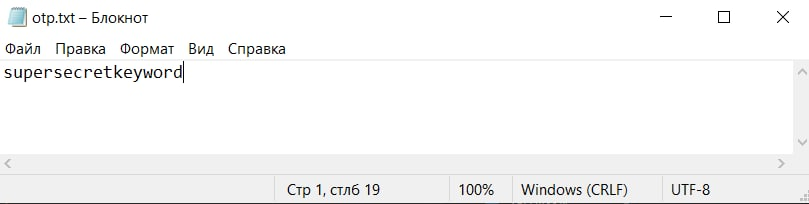


Рисунок 7 – Файл с ключом

Далее, запустим сам скрипт с помощью cmd. Находясь в исходной папке с скриптом и двумя текстовыми файлами пропишем «cmd» в каталоге, чтобы открыть cmd по данному пути в системе (рисунок 8).

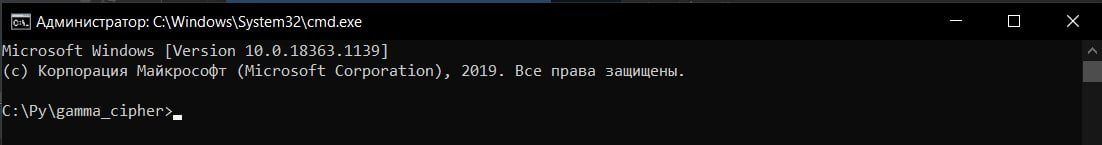


Рисунок 8 – Терминал

Затем, воспользуемся командой:

«python main.py e» (рисунок 9).

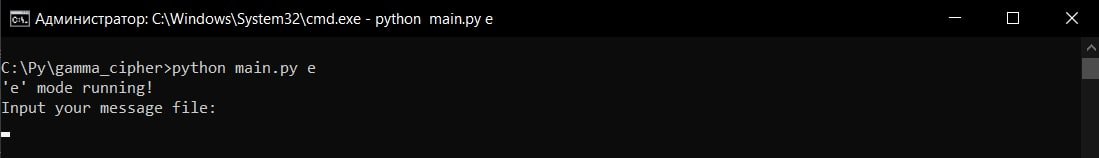


Рисунок 9 – Программа, запущенная в режиме шифрования

Впишем: «message.txt», на следующий вопрос от программы впишем: «otp.txt», после чего программа успешно создаст новый файл с зашифрованным сообщением и выведет его на консоль (рисунок 10).

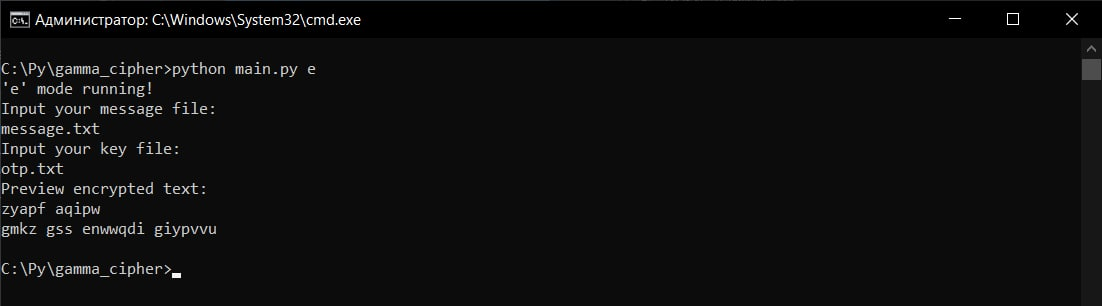


Рисунок 10 – Лог выполнения программы

Теперь расшифруем зашифрованный файл. Для этого запустим программу еще раз, введя аргумент «d». Впишем: «message.txt», на следующий вопрос от программы впишем: «otp.txt», после чего программа успешно создаст новый файл с расшифрованным сообщением и выведет его на консоль (рисунок 11).



Рисунок 11 – Лог выполнения программы

Содержимое полностью совпадает с тем, что было зашифровано, следовательно, программа работает корректно.

Также вывод содержится и в файловом виде (рисунок 12).

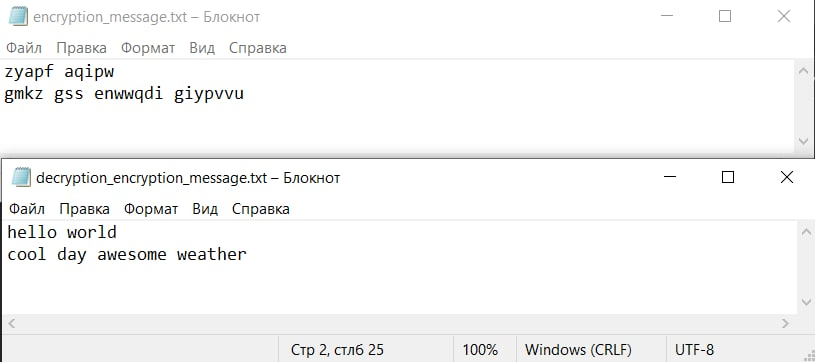


Рисунок 12 – Результат выполнения программы в двух режимах в файловом виде