Министерство НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

Национальный исследовательский университет ИТМО

**факультет безопасности информационных технологий**

**Основы стеганографии**

Лабораторная работа №1

«Основы текстовой стеганографии»

Работу выполнил:

Студент группы №3352   
Александрович Д. А. \_\_\_\_\_\_\_\_\_

Работу проверил:

Давыдов В. В. \_\_\_\_\_\_\_\_\_



Санкт-Петербург, 2020

**Цель работы:** научиться работать с текстовыми алгоритмами стеганографии.

**Теоретическая часть:**

Стеганография [1] – способ передачи или хранения информации с учётом сохранения в тайне самого факта такой передачи (хранения). Текстовая стеганография [2] – это стеганография, использующая текстовые контейнеры для скрытия данных. В данной лабораторной работе были рассмотрены следующие 3 алгоритма:

1. Метод замены символов (метод знаков одинакового начертания);
2. Метод с использованием пробелов (метод хвостовых пробелов);
3. Метод с добавлением служебных символов.

Метод знаков одинакового начертания символы[3] предполагает подмену (бит 1) или ее отсутствие (бит 0) одной графемы другой, имеющей такой же вид начертания. Например, латинская “o” на кириллическую “о”. В моем случае была реализована замена букв о, а и р.

Метод хвостовых пробелов хвостовых пробелов [4] предполагает дописывание (бит 1) или его отсутствие (бит 0) или дописывание одного или двух пробелов в конце строки. Мною был реализован алгоритм дописывания/отсутствие.

Метод с добавление служебных символов предполагает добавление служебных символов или подмену одних служебных символов на другие. Кодирование осуществляется таким же образом, как описано выше.

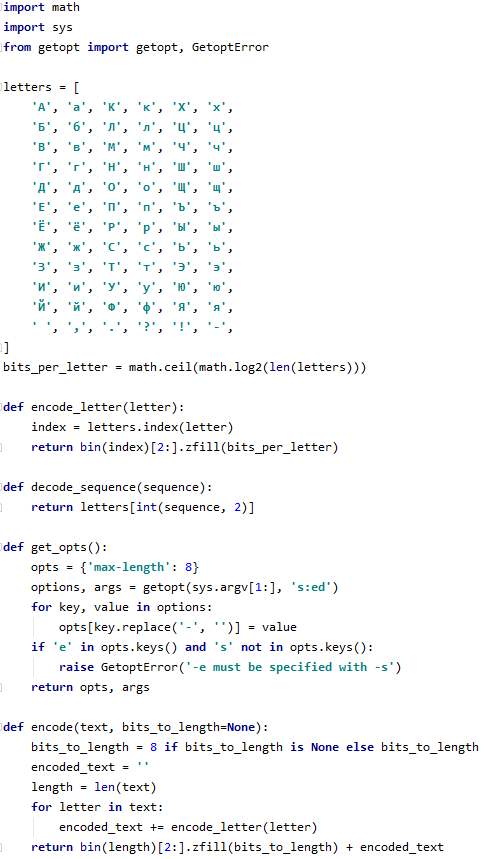
Для того, чтобы зашифровать исходное сообщение, оно преобразуется в двоичный код. Далее двоичная последовательность записывается одним из указанных выше методом. Извлечение информации происходит следующим образом: исходя из выбранного метода текстовый контейнер считывается и записывает полученную последовательность битов. Далее двоичный код преобразуется в зашифрованное сообщение.

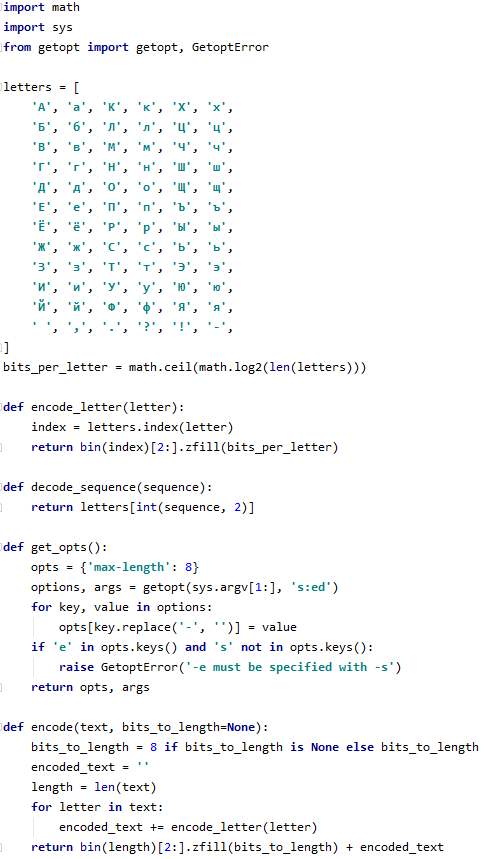
**Практическая часть:**

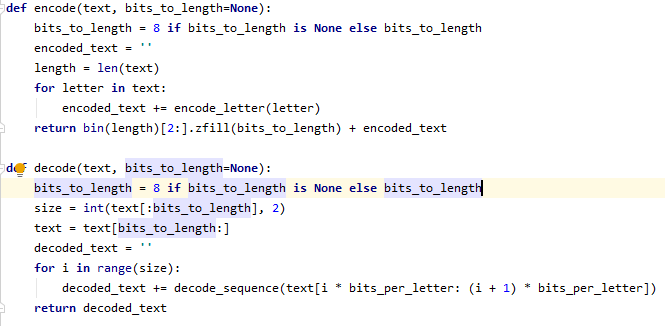
Исходным контейнером в моей работе является рассказ А. П. Чехова «Агафья» состоящий из 19271 символа в кодировке UTF-8.

Все алгоритмы были написаны на языке Python. Для каждого метода был написан отдельный файл, а также общий файл common.py, где хранятся функции, используемые в каждом из методов.

Файл common.py отвечает за функциональность каждого метода. В нем реализованы метод чтения ключей из строки терминала, метод перевода исходного сообщения в двоичный код и метод перевода из двоичного кода в исходное сообщение. Файл содержит массив letters, в котором хранятся символы, которые можно зашифровать. На каждый символ отводится бит (округление происходит в большую сторону).

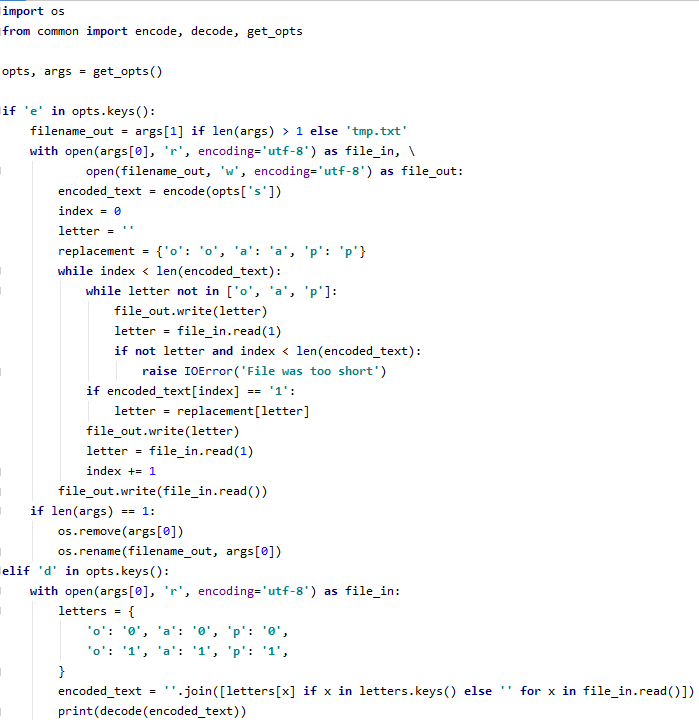


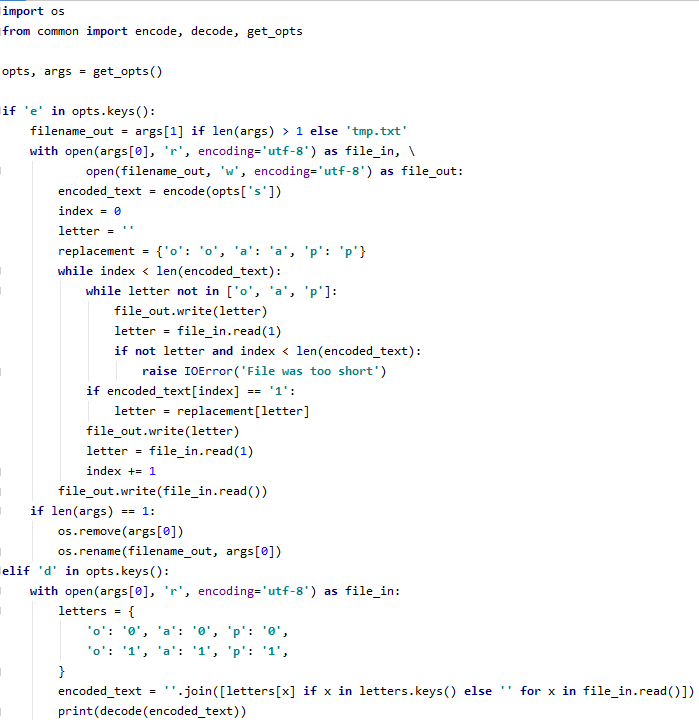




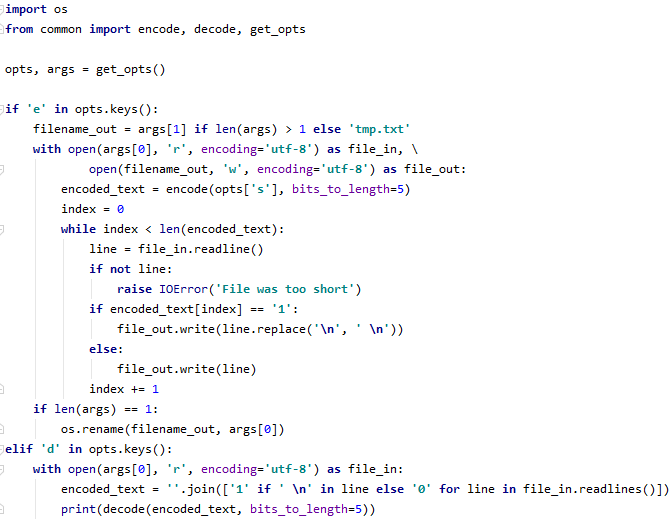
Далее следуют три файла first.py, second.py и third.py, отвечающие за первый, второй и третий методы соответственно. Каждый из скриптов принимает ключи и аргументы. При передаче ключа “-e” скрипт будет работать в режиме шифрования. Ключ “-e” должен подаваться совместно ключом “-s” и последующим строковым значением – сообщением, которое нужно зашифровать. В режиме шифрования нужно подавать также аргументы – название одного или двух файлов – случае, если файл один, происходит запись в этот же файл; если файлов два – один файл становится контейнером, а во второй происходит запись. Ключ “-d” нужен для режима дешифрования. Подается совместно с аргументом – названием файла, откуда нужно считывать информацию.

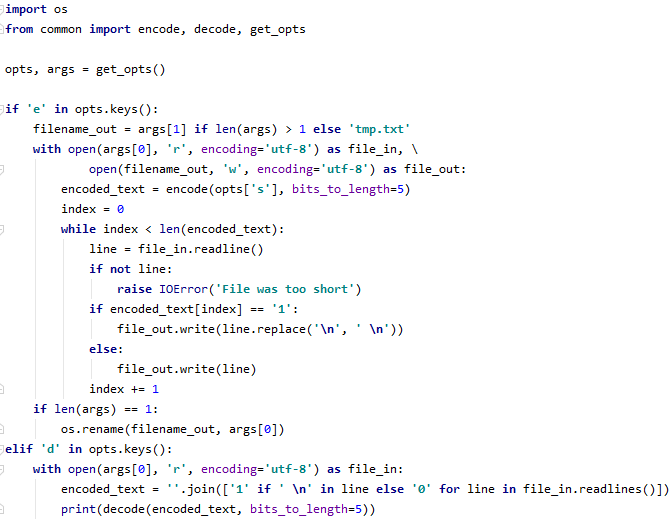
В файле first.py происходит посимвольное считывание исходного контейнера и посимвольная запись в файл. Шифрование и извлечение информации происходит описанным выше способом.



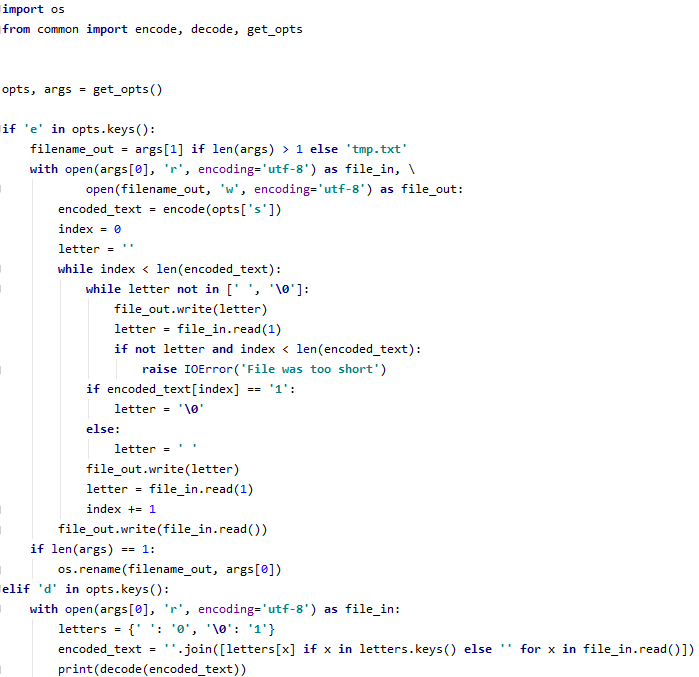


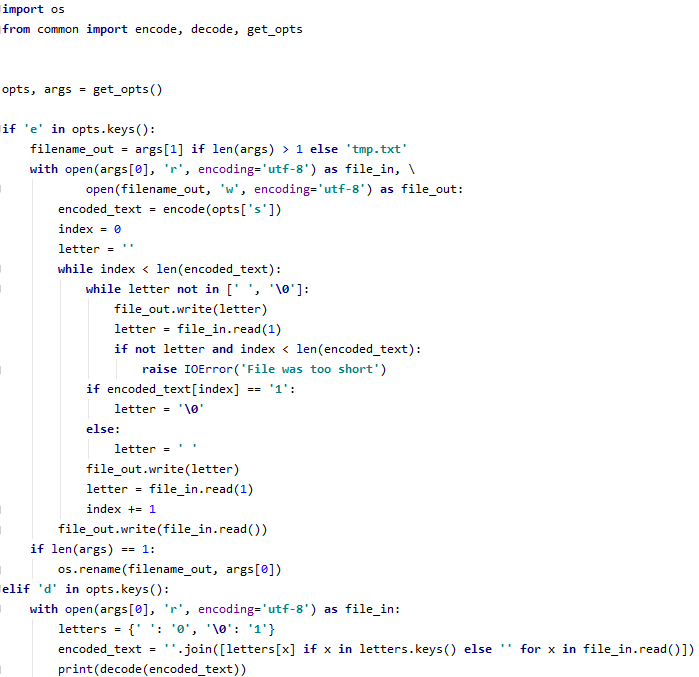
В файле second.py происходит построковое считывание исходного файла и последующая запись с добавлением или без добавления пробела в конце строки.



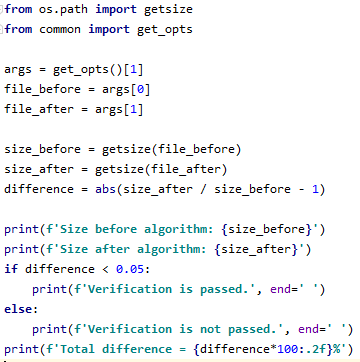


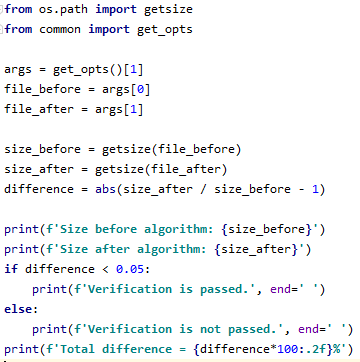
В файле third.py также реализовано посимвольное чтение и последующее посимвольная запись с заменой пробелов на символ ‘\0’, или ее отсутствием в зависимости от битовой информационной последовательности.



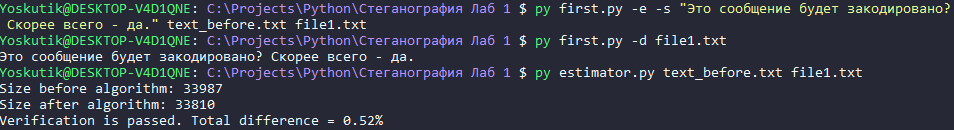


Также мною был написан файл оценщик estimator.py. Этот скрипт сравнивает два файла до вставки сообщения и после – выводит размер каждого файла, а также отличие размера одного файла от другого в процентах. Если разница составляет не более 5%, скрипт выводит сообщение о том, что проверка пройдена.

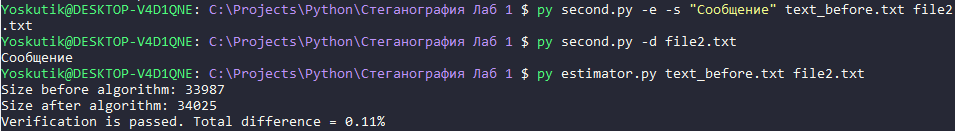




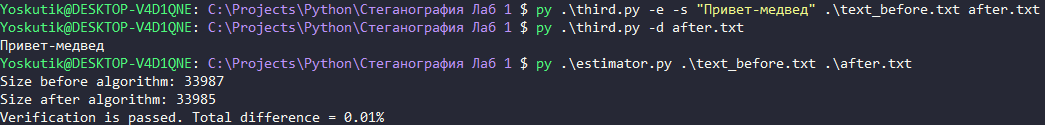
Метод замены символов:



Метод с использованием пробелов:

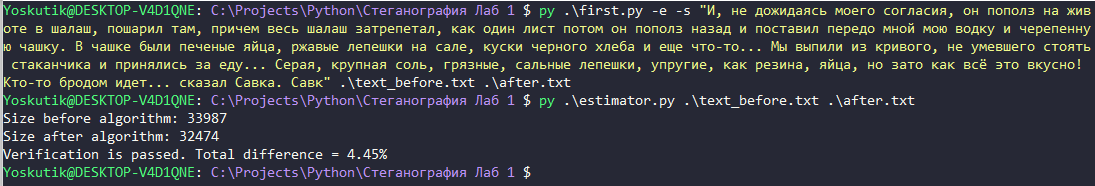


Метод с использованием служебных символов:

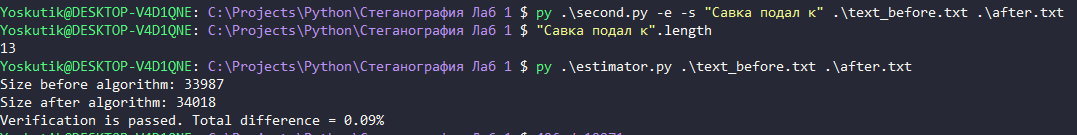


Проведение экспертной оценки:

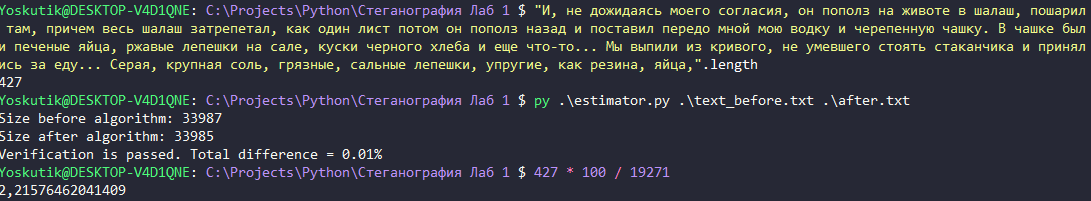
Проведя анализ первого метода, было выявлено, что для выбранного контейнера максимальный объём сообщения, которое можно вписать – 496 символов, что составляет 2.57% от длины исходного текста. И при максимально длинном сообщении файлы до и после вставки сообщения отличаются не более, чем на 4.5%. Следовательно, можно сделать вывод, что данный метод целесообразен.



Второй алгоритм исходя из того, что в контейнере количество строк меньше, чем слов, способен зашифровать меньшее количество символов – всего 13, что составляет 0.07% от длинны исходного контейнера. При максимально длинном сообщении разница между файлами до и после вставки составляет менее 0.1%. Несмотря на небольшое количество символов, которое можно вписать, метод хвостовых пробелов выглядит более надежным, так что можно сделать вывод, что для коротких сообщений этот метод целесообразен.



В третьем методе максимально количество символов – 427, что составляет 2.22% от длины контейнера. На мой взгляд этот метод является самым целесообразным, так как длина сообщения, которое можно вписать достаточно велика, и при этом зрительно отличить пробел от символа ‘\0’ невозможно в большинстве редакторов.



**Выводы:** В ходе данной лабораторной работы мною были изучены основы текстовой стеганографии, а также реализованы 3 её метода: замены символов, с использованием пробелов и с использованием служебных символов. В данной реализации можно кодировать буквы русского алфавита, пробелы и знаки препинания.

**Список использованной литературы:**

1. Стеганография [Электронный ресурс]: 2011 г. URL: <https://habr.com/ru/post/114597/>
2. Текстовая стеганография [Электронный ресурс]: 2004 г. URL: <https://www.osp.ru/pcworld/2004/11/169154/>
3. Текстовая стеганография: Метод знаков одинакового начертания [Электронный ресурс]: 2012 г. URL: <http://www.nestego.ru/2012/05/blog-post_05.html>
4. Текстовая стеганография: Метод хвостовых пробелов [Электронный ресурс]: 2012 г. URL: <http://www.nestego.ru/2012/05/blog-post_03.html> z