Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Исследование криптографических шифров на основе**

**подстановки (замены) символов**

Студент: Гвоздовский К. В.

ФИТ 3 курс 6 группа

Преподаватель:

Нистюк Ольга Александровна

**Лабораторная работа № 4**

**Исследование криптографических шифров на основе подстановки символов**

1. **Цель работы**

Изучение и приобретение практических навыков разработки и использования приложений для реализации подстановочных шифров.

1. **Теоретическая часть:**

Сущность подстановочного шифрования состоит в том, что исходный текст (из множества М) и зашифрованный текст (из множества С) основаны на использовании одного и того же или разных алфавитов, а тайной или ключевой информацией является алгоритм подстановки.

В моноалфавитных шифрах операция замена производится только над каждым одиночным символом сообщения Мi. Для наглядной демонстрации шифра простой замены достаточно выписать под заданным алфавитом тот же алфавит, но в другом порядке или, например, со смещением. Записанный таким образом алфавит называют алфавитом замены.

Максимальное количество ключей для любого шифра этого вида не превышает N!, где N – количество символов в алфавите.

В шифре **Цезаря** шифрование:

y = x + k (mod N),

а расшифрование: х = у – k (mod N).

Диск **Альберти.** Он состоял из двух дисков – внешнего неподвижного и внутреннего подвижного дисков, на которые были нанесены буквы алфавита. Процесс шифрования заключался в нахождении буквы открытого текста на внешнем диске и замене ее на букву с внутреннего диска, стоящую под ней.

Таблица **Трисемуса**. При зашифровании первая буква открытого текста заменяется на букву, стоящую в первой строке, вторая – на букву, стоящую во второй строке, и т.д. После использования последней строки вновь возвращаются к первой.

Шифр **Виженера**. Основная идея заключается в следующем. Создается таблица размером N·N, в первой строке таблицы записывается весь используемый алфавит. Каждая последующая строка получается из предыдущего циклического сдвига последней на 1 символ влево.

1. **Задание**

Разработать авторское приложение в соответствии с целью лабораторной работы. Приложение должно реализовывать следующие операции:

1. выполнять зашифрование/расшифрование текстовых документов (объемом не менее 5 тысяч знаков) созданных на основе алфавита языка в соответствии с нижеследующей таблицей вариантов задания; при этом следует использовать шифры подстановки из третьего столбца данной таблицы;
2. формировать гистограммы частот появления символов для исходного и зашифрованного сообщений;
3. оценивать время выполнения операций зашифрования/расшифрования.
4. **Ход работы**

В соответствии с вариантом, алфавит выбран английский, для первого задания использовался шифр Цезаря с ключевым словом, ключевое слово – собственная фамилия, для второго задания использовалась таблица Трисемуса, ключевое слово – собственное имя.

Для обоих заданий следовало оценить время выполнения операций шифрования/дешифрования сообщений. Для этого был использован специальный класс Date и его метод now() на языке JavaScript, который предоставляет функциональность для измерения времени выполнения операций. Пример использования объекта данного класса представлен на рисунке 3.1.

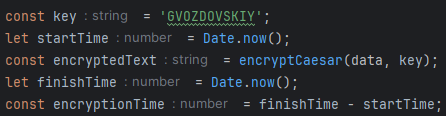


Рисунок 3.1 – Пример использования класса Stopwatch

Также, для этих заданий следовало сформировать гистограммы частот появления символов для исходного и зашифрованного сообщений. Для этой цели была разработана функция CountCharacterFrequency, представленная на рисунке 3.2.

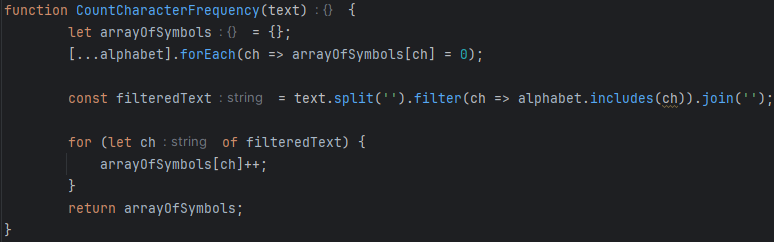


Рисунок 3.2 – Функция для определения частот появления символов

Для выполнения первого задания были разработаны функции EncryptCaesar и DecryptCaesar, которые осуществляют шифрование и дешифрование текста с использованием алгоритма Цезаря с ключевым словом. Они представлены на рисунках 3.3 и 3.4.

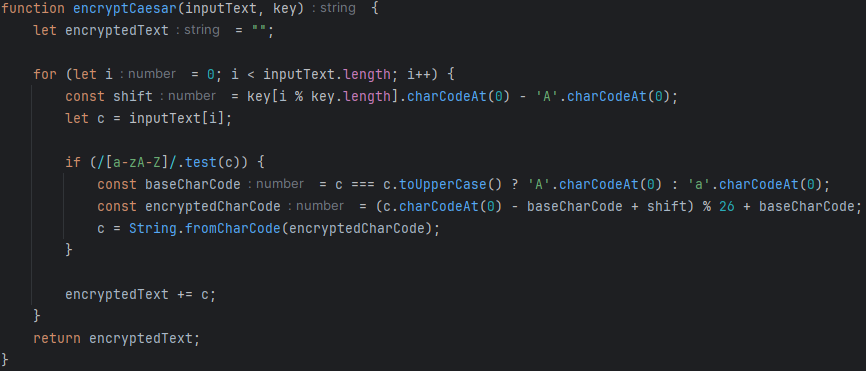


Рисунок 3.3 – Функция для шифрования текста с помощью алгоритма Цезаря с ключевым словом



Рисунок 3.4 – Функция для дешифрования текста с помощью алгоритма Цезаря с ключевым словом

Результаты выполнения данных функций представлены на рисунках 3.5 – 3.8.

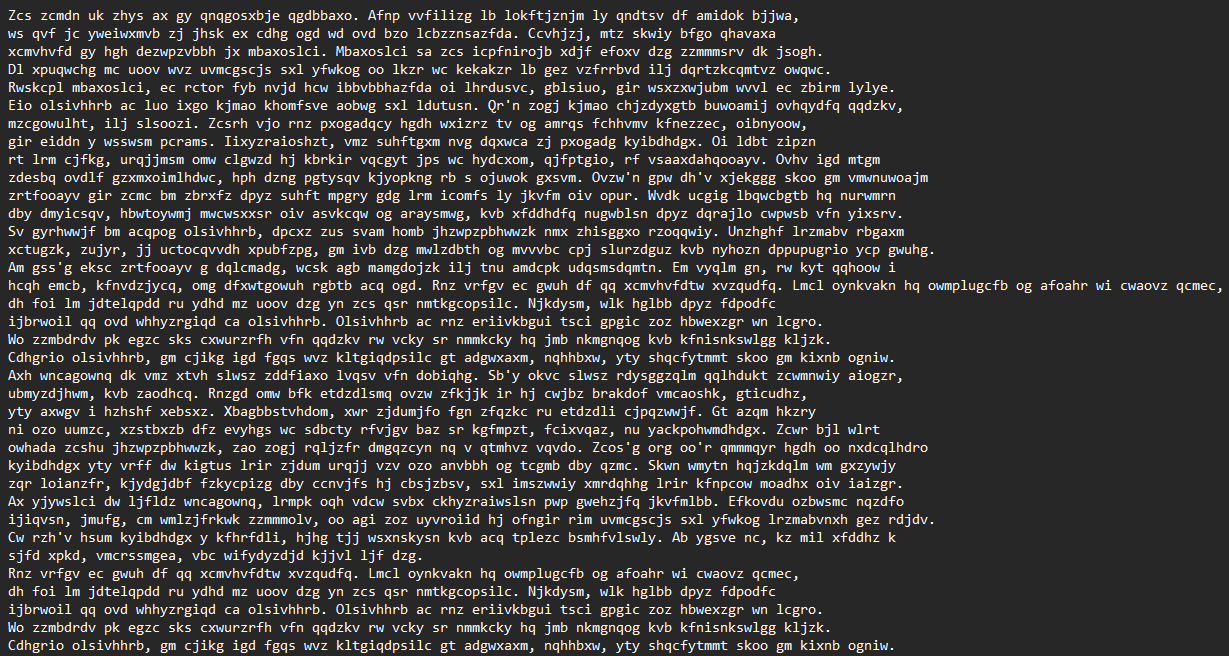


Рисунок 3.5 – Результаты выполнения шифрования текста с помощью алгоритма Цезаря с ключевым словом

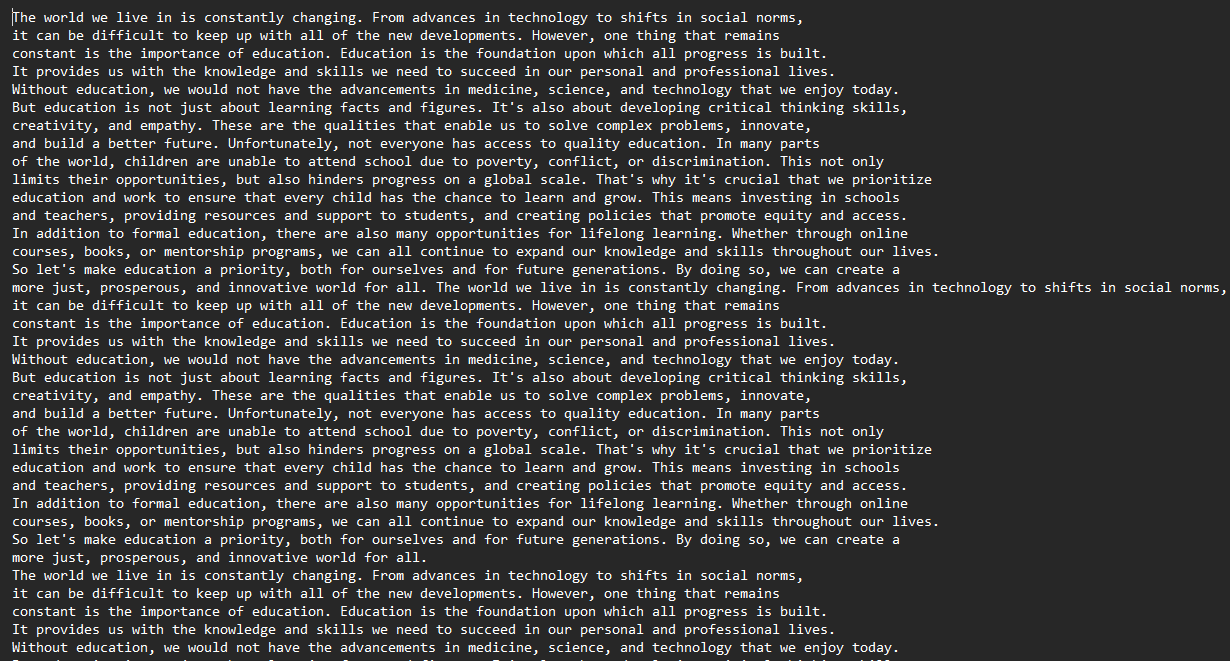


Рисунок 3.6 – Результаты выполнения расшифрования текста с помощью алгоритма Цезаря с ключевым словом

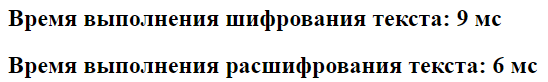


Рисунок 3.7 – Оценка скорости выполнения шифрования/дешифрования текста с помощью алгоритма Цезаря с ключевым словом

Частота появления символов в этих текстовых файлах представлена на рисунках 3.8 и 3.9 соответственно.

Рисунок 3.8– Гистограмма частот появления символов для зашифрованного текста

Рисунок 3.9 – Гистограмма частот появления символов для расшифрованного текста

Для выполнения второго задания были разработаны функции EncryptTrisemus, DecryptTrisemus, которые реализуют алгоритм шифрования текста с помощью таблицы Трисемуса. Код реализации этих функций представлен на рисунках 3.10 и 3.11.



Рисунок 3.10 – Функция для шифрования текста с помощью таблицы Трисемуса



Рисунок 3.11 – Функция для дешифрования текста с помощью таблицы Трисемуса

В результате получаем следующие файлы с зашифрованным и дешифрованным текстами и время выполнения шифрования/дешифрования, представленные на рисунках 3.12 – 3.14.

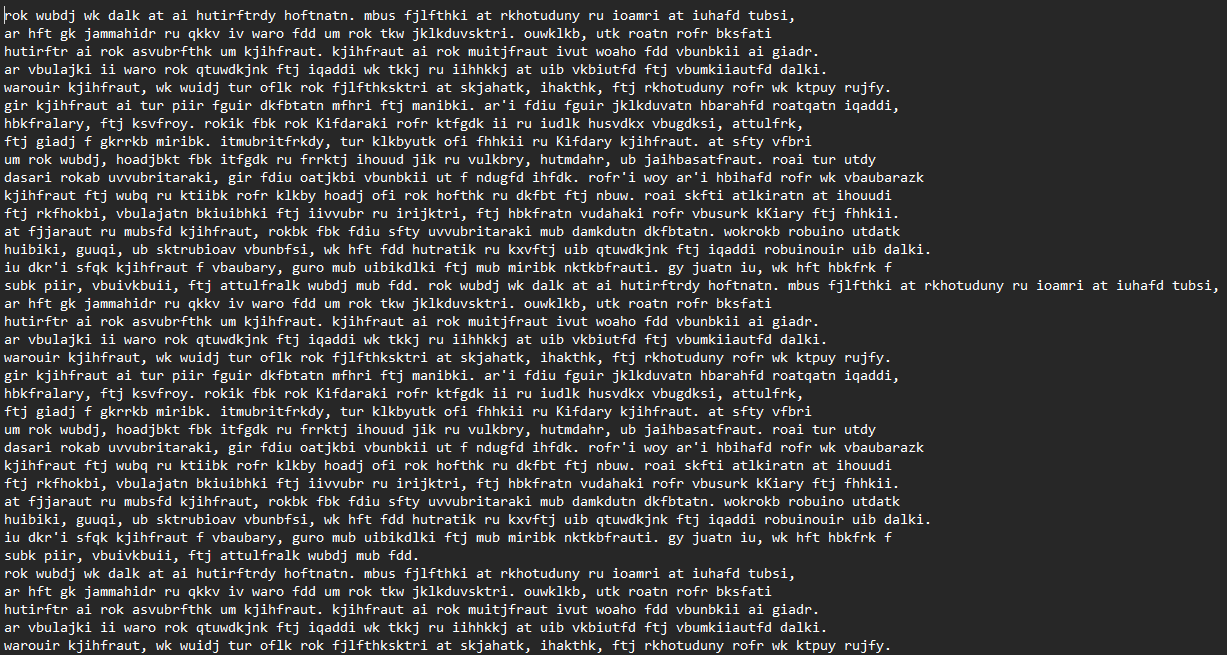


Рисунок 3.12 – Текст, зашифрованный с помощью функции EncryptTrisemus

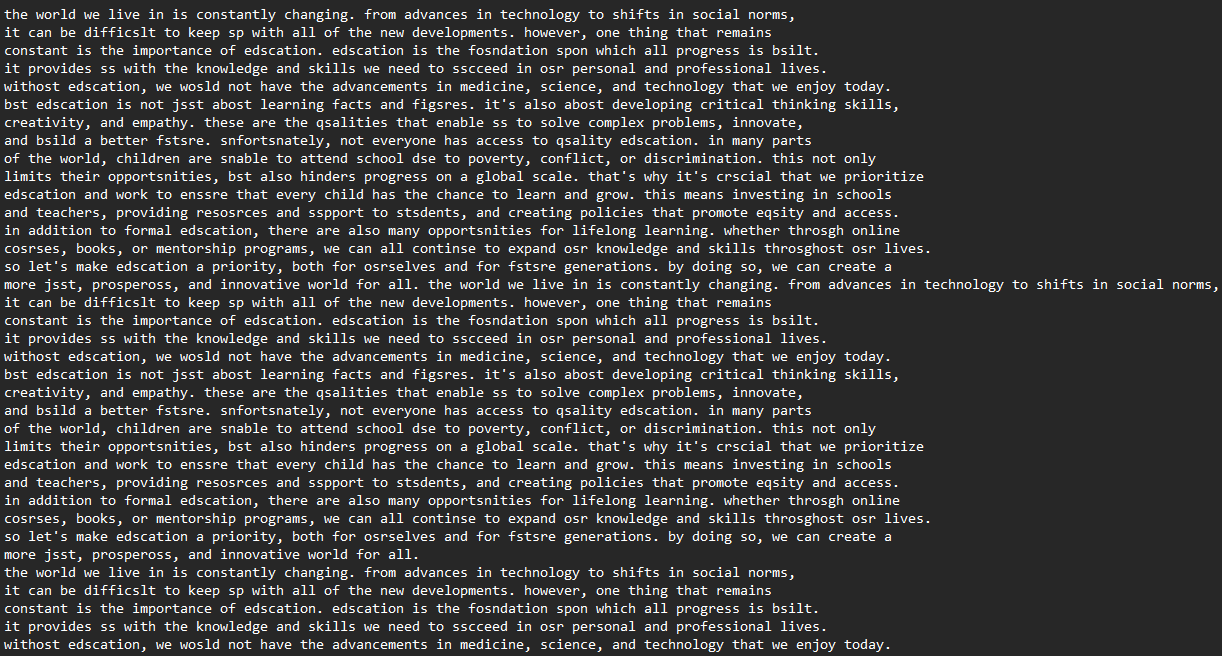


Рисунок 3.13 – Текст, расшифрованный с помощью функции DecryptTrisemus

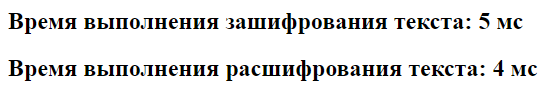


Рисунок 3.14 – Скорость выполнения шифрования/дешифрования текста

Частота появления символов в этих текстовых файлах представлена на рисунках 3.15 и 3.16 соответственно.

Рисунок 3.15– Гистограмма частот появления символов для зашифрованного текста

Рисунок 3.16– Гистограмма частот появления символов для расшифрованного текста

**Вывод**

В ходе лабораторной работы были изучены и приобретены практические навыки разработки и использования приложений для реализации подстановочных шифров.

Кроме того, были сформированы гистограммы появления частот на основе зашифрованных/расшифрованных сообщений. Из графиков видно, что частота символов в зашифрованном тексте существенно отличается от частоты символов в исходном тексте. Это обусловлено тем, что подстановочные шифры заменяют каждый символ открытого текста на другой символ или последовательность символов в зашифрованном тексте. Так как подстановка была случайной и не учитывала частоту появления символов в открытом тексте, то частотный анализ зашифрованного текста сильно отличается от анализа открытого текста.

Также было разработано авторское приложение в соответствии с целью лабораторной работы.