Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Информационная безопасность**

Студент: Козека Е. М.

ФИТ 3 курс 4 группа

Преподаватель: Нистюк О. А.

Минск 2025

**Лабораторная работа №5. Исследование криптографических шифров на основе перестановки символов**

**Цель:** изучение и приобретение практических навыков разработки и использования приложений для реализации перестановочных шифров.

**Задачи:**

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости перестановочных шифров.
2. Ознакомиться с особенностями реализации и свойствами различных перестановочных шифров на основе готового программного средства (L\_LUX).
3. Разработать приложение для реализации указанных преподавателем методов перестановочного зашифрования/расшифрования.
4. Выполнить исследование криптостойкости шифров на основе статистических данных о частотах появления символов в исходном и зашифрованном сообщениях.
5. Оценить скорость зашифрования/расшифрования реализованных способов шифров.
6. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**Теоретические сведения**

*Сущность перестановочного шифрования*состоит в том, что исходный текст (М) и зашифрованный текст (С) основаны на использовании одного и того же алфавита, а тайной или ключевой информацией является алгоритм перестановки.

Шифры перестановки относятся к классу *симметричных.* Элементами текста могут быть отдельные символы (самый распространенный случай), пары, тройки букв и т. д.

Классическими примерами перестановочных шифров являются *анаграммы*. ***Анаграмма*** – литературный прием, состоящий в перестановке букв (или звуков), что в результате дает другое слово или словосочетание, например: проездной – подрезной, листовка – вокалист, апельсин – спаниель.

В классической криптографии шифры перестановки делятся на два подкласса:

• шифры *простой*, или *одинарной*, *перестановки* – при зашифровании символы открытого текста *Мi* перемещаются с исходных позиций в новые (в шифртексте *Сi*) один раз;

• шифры *сложной*, или *множественной*, *перестановки* – при зашифровании символы открытого текста *Мi* перемещаются с исходных позиций в новые (в шифртексте *Сi*) несколько раз.

В общем случае для использования ***шифров одинарной перестановки***используется таблица, состоящая из двух строк: в первой строке записываются буквы, во второй – цифры *J*. Строки состоят из *n* столбцов. Буквы составляют шифруемое сообщение. Цифры *J* = *j*1, *j*2, …, *jn*, где *j*1 – номер позиции в зашифрованном сообщении первого символа открытого текста, *j*2 – номер позиции в зашифрованном сообщении второго символа открытого текста и т. д. Таким образом, порядок следования цифр определяется используемым правилом (ключом) перестановки символов открытого текста для получения шифрограммы.

***Шифры простой блочной перестановки*** строятся по тем же правилам, что и шифры простой перестановки. Блок должен состоять из 2 или более символов. Если общее число таких символов в сообщении не кратно длине сообщения, то последний блок можно дополнить произвольными знаками.

***Шифры маршрутной перестановки***. Основой современных шифров рассматриваемого типа является геометрическая фигура, обычно прямоугольник или прямоугольная матрица. В ячейки этой фигуры по определенному маршруту (слево направо, сверху вниз или каким-либо иным образом) записывается открытый текст. Для получения шифрограммы нужно записать символы этого сообщения в иной последовательности, т. е. по иному маршруту.

***Шифр Скитала (Сцитала).*** Известно, что в V в. до н. э. в Спарте существовала хорошо отработанная система секретной военной связи. Для этого использовался специальный жезл «скитала» – первое, вероятно, простейшее криптографическое устройство, реализующее метод перестановки. Для зашифрования и расшифрования необходимо было иметь абсолютно одинаковые жезлы. На такой предмет наматывалась пергаментная лента. Далее на эту ленту построчно наносился текст. Для расшифрования ленту с передаваемым сообщением нужно было намотать так же, как и при нанесении открытого текста.

Следуя вышеприведенным рассуждениям, можно отождествить Скитала с таблицей размерами *k×s*, где *k* – количество столбцов, *s* – количество строк. Поскольку при регулярном обмене данными сообщения часто имеют разную длину, то оба этих параметра за неизменяющийся ключ взять неудобно. Поэтому обычно в качестве известного каждой стороне ключа выбирается один из них (часто это *s*), а второй вычисляется на основе известного и длины *n* сообщения *Мi*:

*k* = [(*n* - 1)/*s*] + 1.

При этом слагаемое в квадратных скобках должно быть целым числом.

При использовании шифра Скитала для формирования шифртекста сначала выбирается 1-я буква открытого текста, затем (*k* + 1)-я буква, (2*k* + 1)-я буква и т. д. для некоторого *k*, равного числу букв в каждой строке Скиталы. Значение *k* является постоянной величиной для данной Скиталы.

***Организация маршрутной перестановки.*** Уже упоминавшаяся маршрутная перестановка (записываем сообщение по строкам, считываем – по столбцам матрицы) можно усложнить и считывать не по столбцам, а по спирали, зигзагом, змейкой или каким-то другим способом. Такие способы шифрования несколько усложняют процесс, однако усиливают криптостойкость шифра.

***Шифр вертикальной перестановки*** является разновидностью шифра маршрутной перестановки. К особенностям вертикального шифра можно отнести следующие:

• количество столбцов в таблице фиксируется и определяется длиной ключа;

• маршрут вписывания: слево направо, сверху вниз;

• шифрограмма выписывается по столбцам в соответствии с их нумерацией (ключом).

Особенностью ***шифров множественной перестановки*** является минимум двукратная перестановка символов шифруемого сообщения. В простейшем случае это может задаваться перемешиванием не только столбцов, но и строк. Таким образом, этот случай соответствует использованию двух основных ключей: длина одного из них равна числу столбцов, другого – числу строк. К ключевой информацию мы можем относить также способы вписывания сообщения и считывания отдельных символов из текущего столбца матрицы.

Шифры гаммирования рассматриваются как самостоятельный класс. Такие шифры схожи с перестановочными тем, что в обоих случаях можно использовать табличное представление выполняемых операций на основе ключей. Вместе с тем шифры гаммирования имеют много общего с подстановочными шифрами, поскольку на самом деле при зашифровании происходит подмена одних символов другими.

**Ход работы**

**Задание 1.** Разработать авторское приложение в соответствии с целью лабораторной работы. Приложение должно выполнять зашифрование и расшифрование текстовых документов (объемом не менее 500 знаков), созданных на основе алфавита языка в соответствии с нижеследующей таблицей вариантов задания; при этом следует использовать шифры подстановки из третьего столбца данной таблицы.

Для начала, был реализован алгоритм шифрования и расшифрования с помощью маршрутной перестановки. Маршрут – по зигзагу.

Пример работы алгоритма:

1. Допустим, исходное сообщение – «London is a capital of Great Britain».

2. Длина сообщения – 36 символов. Размер таблицы перестановки: 4x9.

Исходная таблица:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L | o | n | d | o | n |  | i | s |
|  | a |  | c | a | p | i | t | a |
| l |  | o | f |  | G | r | e | a |
| t |  | B | r | i | t | a | i | n |

3. Далее для зашифрования сообщения таблица считывается зигзагом по столбцам (первый столбец – по направлению вниз, второй – вверх, третий – снова вниз и т. д.). Зашифрованное сообщение: «L lt aon oBffcdoa itGpn iraietisaan».

4. При расшифровании строится новая таблица, в которой столбцы и строки меняются местами. Т.е. если таблица была размеров 4x9, то новая таблица будет размеров 9x4. Зашифрованное сообщение записывается по строкам (при переходе на новую строку направление записи меняется на противоположное) и считывается по столбцам.

Таблица для расшифрования:

|  |  |  |  |
| --- | --- | --- | --- |
| L |  | l | t |
| o | a |  |  |
| n |  | o | B |
| d | c | f | f |
| o | a |  | i |
| n | p | G | t |
|  | i | r | a |
| i | t | e | i |
| s | a | a | n |

Расшифрованное сообщение: «London is a capital of Great Britain».

Функция шифрования routeTranspositionEncode принимает на вход исходный текст и размер для таблицы. На основе исходного текста она формирует таблицу и начинает её обход по строкам.

const routeTranspositionEncode = (text, tableSize) => {

    let table = formEncryptionTable(text, tableSize);

    let result = "";

    for (let col = 0; col < table[0].length; col++) {

        for (let row = 0; row < table.length; row++) {

            const zigzagRow = col % 2 === 0

                ? row : table.length - 1 - row;

            result += table[zigzagRow][col];

        }

    }

    return result;

}

Листинг 5.1 – Функция routeTranspositionEncode

Функция расшифрования routeTranspositionDecode принимает на вход зашифрованный текст и размер таблицы. Она формирует таблицу и обходит её по столбцам.

const routeTranspositionDecode = (text, tableSize) => {

    let table = formDecryptionTable(text, tableSize);

    let result = "";

    let index = 0;

    for (let col = 0; col < table[0].length; col++) {

        for (let row = 0; row < table.length; row++) {

            const zigzagRow = col % 2 === 0

                ? row : table.length - 1 - row;

            if (index < text.length) {

                table[zigzagRow][col] = text[index]; index++;

            }

        }

    }

    for (let row = 0; row < table.length; row++) {

        for (let col = 0; col < table[row].length; col++) {

            result += table[row][col];

        }

    }

    return result;

}

Листинг 5.2 – Функция routeTranspositionDecode

Результат работы функций приведен на рисунке 5.1.

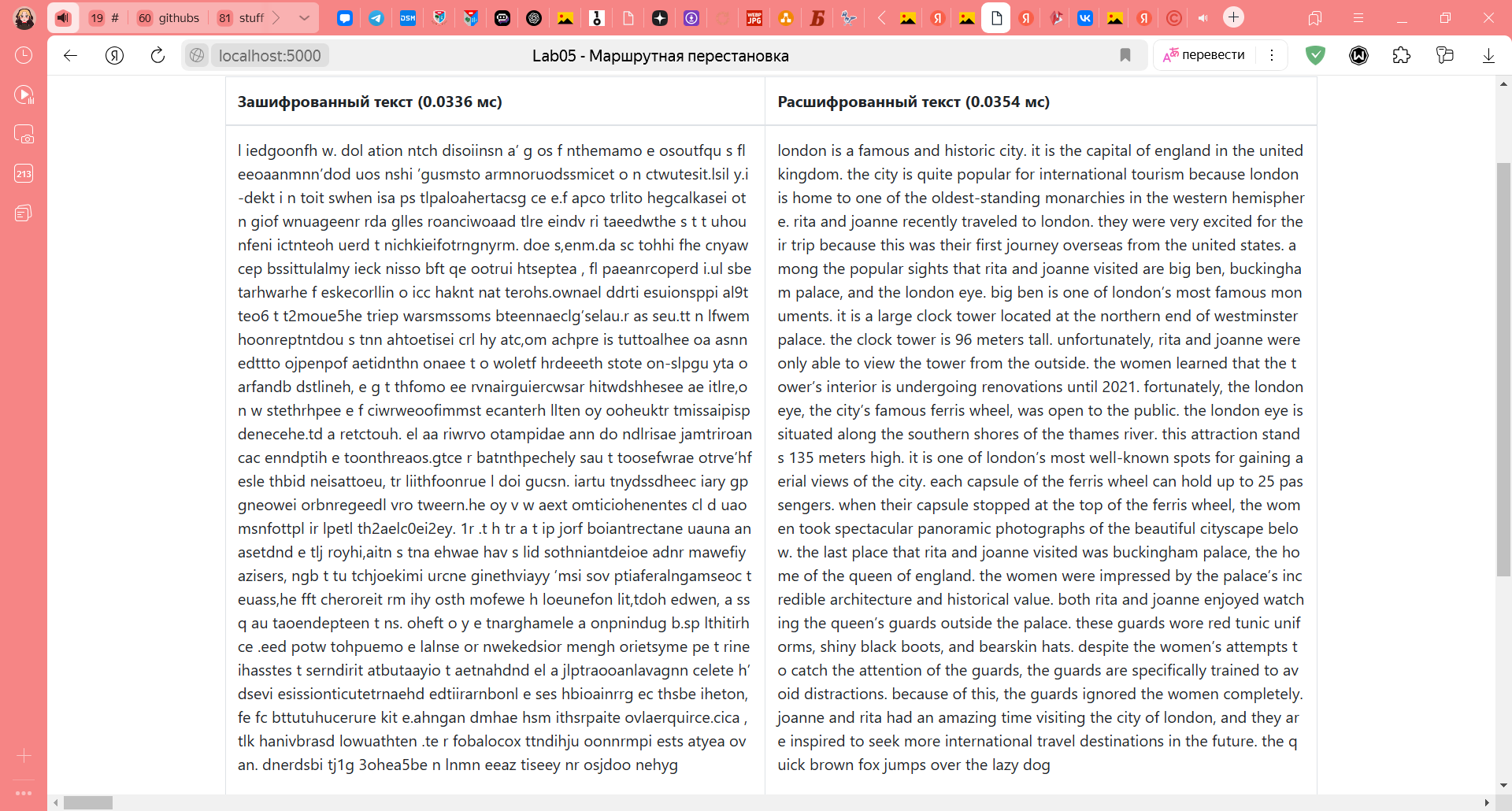


Рисунок 5.1 – Результат шифрования и дешифрования текста с помощью маршрутной перестановки

Затем были разработаны несколько функций, выполняющих шифрование и дешифрование текста с использованием множественной перестановки, где ключевые слова – собственное имя и фамилия.

Сначала была разработана функция formSortedTable, которая принимает размер таблицы и таблицу с исходным порядком символов в сообщении. Далее, с помощью ключевых слов, она сортирует таблицу сначала по строкам, а затем по столбцам.

const formSortedTable = (tableSize, table) => {

    let sortedTable = [];

    const { nameKeywordIndexes, lastnameKeywordIndexes } = transformKeywords(tableSize);

    for (let i = 0; i < nameKeywordIndexes.length; i++) {

        sortedTable[nameKeywordIndexes[i][1]] = table[i];

    }

    table = structuredClone(sortedTable);

    for (let i = 0; i < nameKeywordIndexes.length; i++) {

        for (let j = 0; j < lastnameKeywordIndexes.length; j++) {

            sortedTable[i][lastnameKeywordIndexes[j][1]] = table[i][j];

        }

    }

    return sortedTable;

}

Листинг 5.3 – Функция formSortedTable

Функция шифрования multipleTranspositionEncode принимает на вход исходный текст и таблицу перестановок.

const multipleTranspositionEncode = (text, table) => {

    let result = "";

    for (let i = 0; i < table[0].length; i++) {

        for (let j = 0; j < table.length; j++) {

            result += table[j][i];

        }

    }

    return result;

}

Листинг 5.4 – Функция multipleTranspositionEncode

Функция расшифрования multipleTranspositionDecode принимает на вход зашифрованный текст и таблицу перестановок.

const multipleTranspositionDecode = (tableSize, table) => {

    let unsortedTable = [];

    let { nameKeywordIndexes, lastnameKeywordIndexes } = transformKeywords(tableSize);

    for (let i = 0; i < nameKeywordIndexes.length; i++) {

        unsortedTable[i] = [];

        for (let j = 0; j < lastnameKeywordIndexes.length; j++) {

            unsortedTable[i][j] = table[i][lastnameKeywordIndexes[j][1]];

        }

    }

    table = structuredClone(unsortedTable);

    for (let i = 0; i < nameKeywordIndexes.length; i++) {

        unsortedTable[i] = table[nameKeywordIndexes[i][1]];

    }

    let result = "";

    for (let i = 0; i < unsortedTable.length; i++) {

        for (let j = 0; j < unsortedTable[0].length; j++) {

            result += unsortedTable[i][j];

        }

    }

    return result;

}

Листинг 5.5 – Функция multipleTranspositionDecode

Результат работы функций приведен на рисунке 5.2.

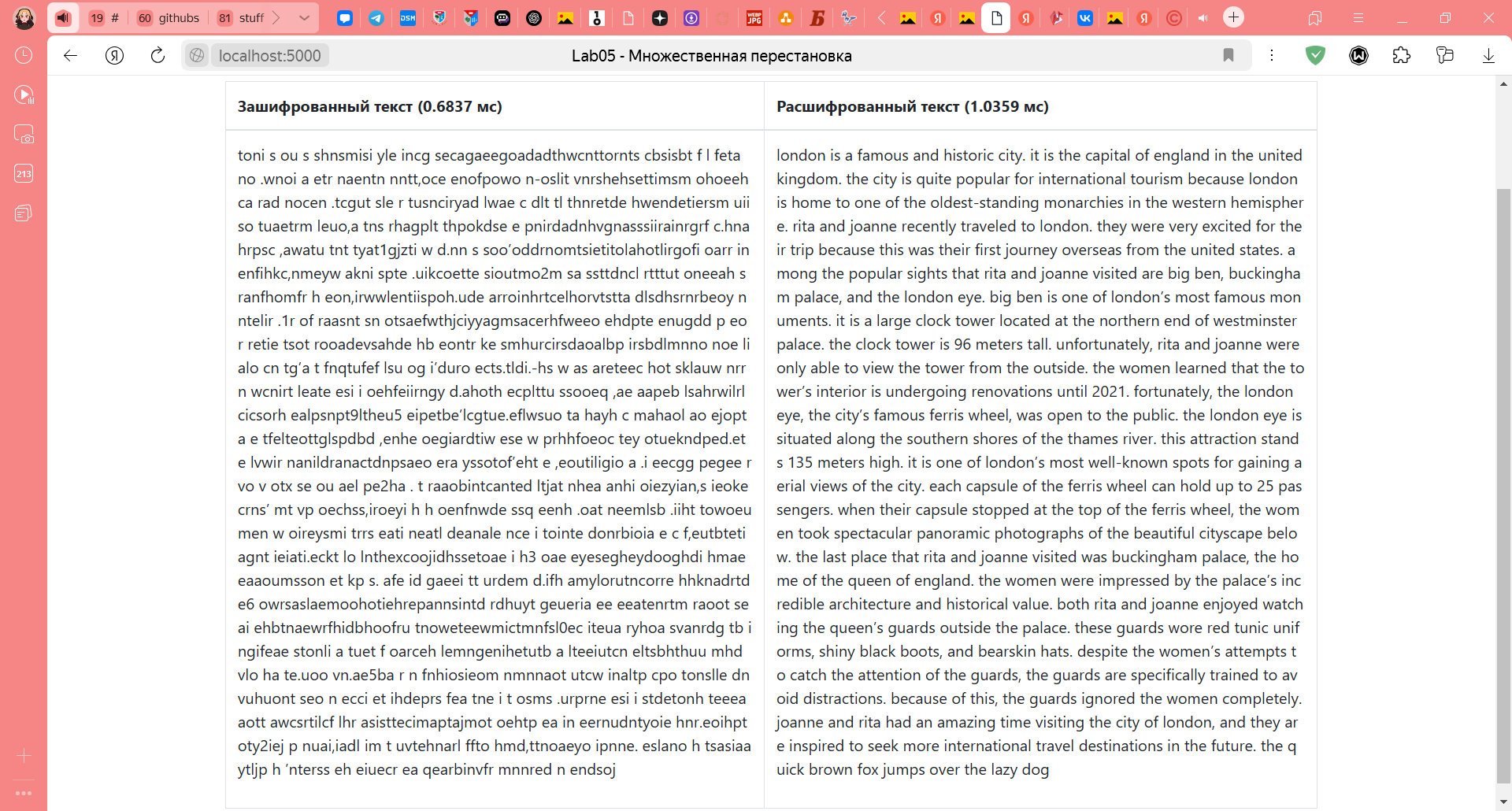


Рисунок 5.2 – Результат шифрования и дешифрования текста с помощью множественных перестановок с ключевыми словами «liza» и «kozeka»

**Задание 2.** Сформировать гистограммы частот появления символов для исходного и зашифрованного сообщений. Оценить время выполнения операций зашифрования/расшифрования.

***Маршрутная перестановка:***

Гистограмма частот появления символов исходного сообщения приведена на рисунке 5.3.

Рисунок 5.3 – Гистограмма частот появления символов исходного сообщения

Гистограмма частот появления символов зашифрованного сообщения приведена на рисунке 5.4.

Рисунок 5.4 – Гистограмма частот появления символов зашифрованного сообщения

Также было определено, что как зашифрование, так и расшифрование текста с помощью маршрутной перестановки выполняется менее чем за 1 миллисекунду.

***Множественная перестановка:***

Гистограмма частот появления символов исходного сообщения приведена на рисунке 5.5.

Рисунок 5.5 – Гистограмма частот появления символов исходного сообщения

Гистограмма частот появления символов зашифрованного сообщения приведена на рисунке 4.6.

Рисунок 4.6 – Гистограмма частот появления символов зашифрованного сообщения

Также было определено, что как зашифрование, так и расшифрование текста с помощью множественной перестановки выполняется примерно за 1-3 миллисекунды, что уступает по сравнению с шифрованием и расшифрованием с помощью маршрутной перестановки.

**Вывод:** в ходе выполнения лабораторной работы были изучены основных принципов работы перестановочных шифров, методов их зашифрования и расшифрования. Также было разработано приложение для реализации методов маршрутной и множественной перестановок. Было выполнено исследование криптостойкости шифров на основе статистических данных о частотах появления символов в исходном и зашифрованном сообщениях. Была проведена оценка скорости зашифрования и расшифрования реализованных способов шифров. Это позволило сравнить эффективность различных алгоритмов.