Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Лабораторная работа № 5**

«ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ ШИФРОВ НА ОСНОВЕ ПЕРЕСТАНОВКИ СИМВОЛОВ»

Выполнил:

Студент: Белицкий В.Д.

ФИТ 2 курс 5 группа

Преподаватель: Савельева М.Г.

Минск 2023

1. **Описание приложения**

Приложение написано на языке программирования C# и позволяет:

* Зашифровать и расшифровать сообщение используя алгоритм маршрутной перестановки (запись – по строкам, считывание по столбцам)
* Зашифровать и расшифровать сообщение с помощью шифра множественной перестановки используя два ключевых слова (имя и фамилия)
* Оценить время выполнения каждого шифра

**Шифр маршрутной перестановки** (также называемый шифром обходного пути) — это метод шифрования, который использует перестановку символов в сообщении для создания шифруемого текста. При этом перестановка символов осуществляется с помощью специального «маршрута», который определяет порядок перестановки.

В шифре маршрутной перестановки для шифрования сообщения используется квадратная матрица, в которую последовательно записываются символы сообщения в строках (или столбцах) согласно маршруту. Затем, для получения шифруемого текста, символы матрицы считываются по определенному порядку (например, сверху вниз, слева направо) и записываются в виде строки.

Для расшифровки сообщения необходимо знать маршрут, по которому были переставлены символы, и использовать обратную перестановку для восстановления исходного текста.

**Шифр множественной перестановки** — это метод шифрования, который использует несколько перестановок символов в сообщении для создания шифруемого текста. При этом каждая перестановка производится независимо друг от друга и по отдельности, что обеспечивает более высокую стойкость шифра.

Для шифрования сообщения в шифре множественной перестановки используются ключевые слова, которые определяют порядок перестановки символов. Обычно ключевые слова используются для определения порядка перестановки символов в строках и столбцах, которые затем записываются в матрицу. Количество строк и столбцов матрицы определяется длиной ключевых слов.

1. **Методика выполнения расчетов**

В данной лабораторной работе была поставлена цель создания приложения, позволяющее произвести зашифрование и расшифрование на основе маршрутной и множественной перестановок. Добавлена функция оценки времени выполнения метода шифрования.

public static string Encrypt(string input, int rows, int columns)

{

char[,] table = new char[rows, columns]; // создание таблицы

// заполнение таблицы

int index = 0;

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < columns; j++)

{

if (index < input.Length)

{

table[i, j] = input[index];

index++;

}

else

{

table[i, j] = ' ';

}

}

}

// считывание таблицы по столбцам и запись зашифрованного сообщения

string output = "";

for (int j = 0; j < columns; j++)

{

for (int i = 0; i < rows; i++)

{

output += table[i, j];

}

}

return output;

}

Листинг 2.1 – Шифрование для маршрутной перестановки

public static string Decrypt(string input, int rows, int columns)

{

char[,] table = new char[rows, columns]; // создание таблицы

// заполнение таблицы

int index = 0;

for (int j = 0; j < columns; j++)

{

for (int i = 0; i < rows; i++)

{

if (index < input.Length)

{

table[i, j] = input[index];

index++;

}

else

{

table[i, j] = ' ';

}

}

}

// запись расшифрованного сообщения

string output = "";

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < columns; j++)

{

output += table[i, j];

}

}

return output;

}

Листинг 2.2 – Расшифрование для маршрутной перестановки

public static string Encrypt2(string plaintext, string rowKey, string columnKey)

{

int rows = rowKey.Length;

int columns = columnKey.Length;

int length = rows \* columns;

char[,] matrix = new char[rows, columns];

int index = 0;

for (int j = 0; j < columns; j++)

{

int k = columnKey.IndexOf(columnKey[j]);

for (int i = 0; i < rows; i++)

{

int l = rowKey.IndexOf(rowKey[i]);

if (index < plaintext.Length)

{

matrix[l, k] = plaintext[index];

index++;

}

else

{

matrix[l, k] = ' ';

}

}

}

string ciphertext = "";

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < columns; j++)

{

ciphertext += matrix[i, j];

}

}

return ciphertext;

}

Листинг 2.3 – Шифрование для множественной перестановки

public static string Decrypt2(string ciphertext, string rowKey, string columnKey)

{

int rows = rowKey.Length;

int columns = columnKey.Length;

int length = rows \* columns;

char[,] matrix = new char[rows, columns];

// заполнение матрицы по столбцам

int index = 0;

for (int j = 0; j < columns; j++)

{

int k = columnKey.IndexOf(columnKey[j]);

for (int i = 0; i < rows; i++)

{

int l = rowKey.IndexOf(rowKey[i]);

matrix[l, k] = ciphertext[index];

index++;

}

}

// считывание зашифрованного сообщения по строкам

string plaintext = "";

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < columns; j++)

{

plaintext += matrix[i, j];

}

}

return plaintext.TrimEnd(); // удаляем лишние пробелы в конце строки

}

Листинг 2.4 – Расшифрование для множественной перестановки

**Рис 2.1 – График оценки времени выполнения множественной перестановкой**

**Рис 2.2 – График оценки времени выполнения зашифрования множественной перестановкой**

**Рис 2.3 – Частота появления символов в исходном сообщении (маршрутная перестановка)**

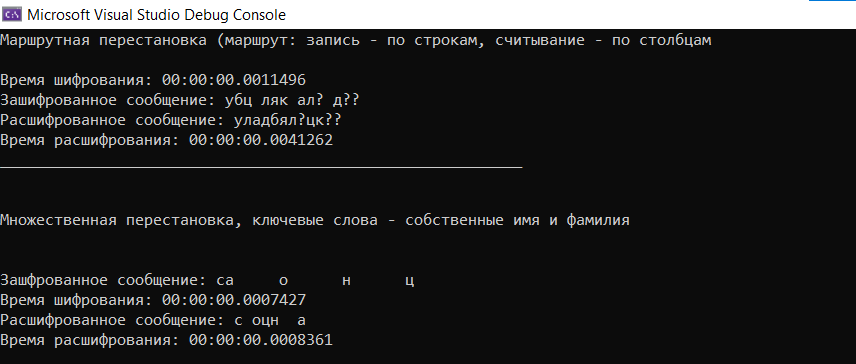
**Рис 2.4 – Частота появления символов в зашифрованном сообщении (маршрутная перестановка)**

**Рис 2.5 – Частота появления символов в исходном сообщении (множественная перестановка)**

**Рис 2.6 – Частота появления символов в зашифрованном сообщении (множественная перестановка)**

**3. Результаты работы приложения**

Для выполнения расчетов достаточно необходимо запустить приложение. Следующий рисунок показывает необходимые расчеты и вызовы методов, требуемые в данной лабораторной работе.



**Рис 3.1 – Результат работы приложения**

1. **Вывод по гистограмме**

При зашифровании алгоритмом маршрутной перестановки были допущены неточности, связанные с белорусской буквой *i* она не корректно распознается программой, также были добавлены пробелы, впрочем, они не влияют на прочтение результата расшифрования.

При зашифровании алгоритмом множественной перестановки, может возникнуть таже проблема из-за выбранного языка по варианту (белорусский). Количество символов осталось неизменным, не считаю лишние пробелы.

**Вывод**

В ходе изучения теоретических материалов лабораторной работы и выполнения её практической части было разработано авторское приложение для шифрования и расшифрования заданного текста. Использовались:

1. Шифр маршрутной перестановки
2. Шифр множественной перестановки

Также приложение оценивает время всех произведенных операций с текстом.