МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Информационных систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Направление специальности 1-40 01 01 10 Программирование интернет-приложений

**ОТЧЁТ К ЛАБОРАТОРНОЙ РАБОТЕ №4:**

по дисциплине «Криптографические методы защиты информации»

Тема «ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ ШИФРОВ НА ОСНОВЕ ПОДСТАНОВКИ (ЗАМЕНЫ) СИМВОЛОВ»

Исполнитель

студент 3 курса группы 5 Круглик Алексей Викторович

(Ф.И.О.)

Руководитель работы преподаватель Савельева М. Г.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Минск 2023

Ход работы

Приложение было разработано на С#. В своем приложении согласно варианту 9 я использовала алфавит немецкого языка, а так же два способа шифрования: шифр на основе соотношений при *k*=7 и таблица Трисемуса, где ключевое слово – enigma.

Здание №1

В данном задании я разработал шифр на основе соотношений при *k*=7. При данном методе у нас происходит сдвиг на 7 позиций вправо.

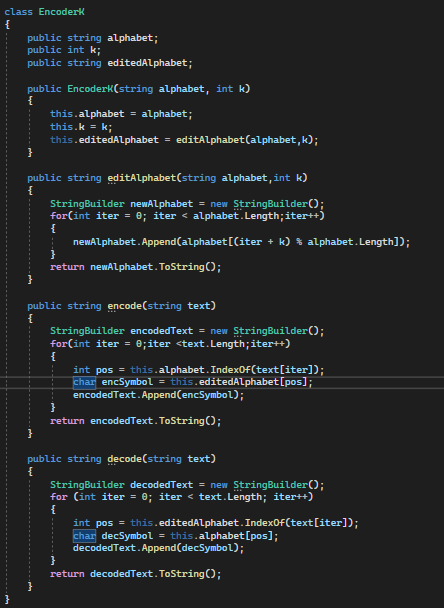


Рисунок 1 — реализация шифра соотношения при *k* = 7

Задание №2

В данном задании я разработал шифр на основе таблицы Трисемуса, ключевое слово – enigma. Для реализации данного алгоритма надо было разработать код для создания таблицы Трисемуса с ключевым словом enigma. Данный код приведен на рисунке 2.

public char[,] makeTable(string alphabet,string keyWord)

{

this.height = 5;

this.width = 6;

int k = 0;

int heightA = 0,widthA = 0;

char[,] finalTable = new char[height, width];

for(int iterY = 0; iterY < height;iterY++)

{

if (k >= keyWord.Length)

{

heightA = iterY;

break;

}

for (int iterX = 0; iterX < width;iterX++)

{

if(k < keyWord.Length)

{

if(!containSymbol(finalTable, keyWord[k],height,width))

{

finalTable[iterY, iterX] = keyWord[k++];

}

}

else

{

widthA = iterX;

break;

}

}

}

for(int iter = 0;iter < alphabet.Length; iter++)

{

if(widthA >= width)

{

heightA++;

widthA = 0;

}

if(heightA < height)

{

if (!containSymbol(finalTable, alphabet[iter], height, width))

{

finalTable[heightA, widthA++] = alphabet[iter];

}

}

else

{

break;

}

}

return finalTable;

}

private bool containSymbol(char[,] table, char symbol,int height, int width)

{

for (int iterY = 0; iterY < height; iterY++)

{

for (int iterX = 0; iterX < width; iterX++)

{

if (table[iterY, iterX] == symbol)

{

return true;

}

else

{

continue;

}

}

}

return false;

}

public string encode(string text)

{

StringBuilder stringFinal = new StringBuilder();

foreach(char x in text)

{

for(int iterY = 0; iterY < height;iterY++)

{

for (int iterX = 0; iterX < width; iterX++)

{

if(x == tableAlphabet[iterY, iterX])

{

if(iterY + 1 < height)

{

if(tableAlphabet[iterY + 1, iterX] != '\0')

stringFinal.Append(tableAlphabet[iterY + 1, iterX]);

else

stringFinal.Append(tableAlphabet[0, iterX]);

}

else

{

stringFinal.Append(tableAlphabet[0, iterX]);

}

}

}

}

}

return stringFinal.ToString();

}

public string decode(string text)

{

StringBuilder stringFinal = new StringBuilder();

foreach (char x in text)

{

for (int iterY = 0; iterY < height; iterY++)

{

for (int iterX = 0; iterX < width; iterX++)

{

if (x == tableAlphabet[iterY, iterX])

{

if (iterY - 1 >= 0)

{

if (tableAlphabet[iterY - 1, iterX] != '\0')

stringFinal.Append(tableAlphabet[iterY - 1, iterX]);

else

stringFinal.Append(tableAlphabet[height - 1, iterX]);

}

else

{

stringFinal.Append(tableAlphabet[height - 1, iterX]);

}

}

}

}

}

return stringFinal.ToString();

}

public void printMatrix(char[,] input)

{

for (int w = 0; w < height ; w++)

{

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

{

Console.Write($"{(input[w, i] == ' ' ? '\*' : input[w, i])} ");

}

Console.Write($"\n");

}

Console.Write($"\n");

}

Приложение 1 — код для создания таблицы Трисемуса с ключевым словом enigma.

Полученная таблица представлена на рисунке 2.

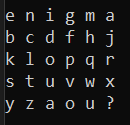


Рисунок 2 — таблица Трисемуса с ключевым словом enigma

Далее был произведен анализ гистограмм. Сами гистограммы представлены на рисунках 3-4. Гистограммы соответствующие частоте появления букв при шифровании и дешифровании текста с таблицей Трисемуса, ключевое слово – enigma представлены на рисунках 3, 4 соответственно.

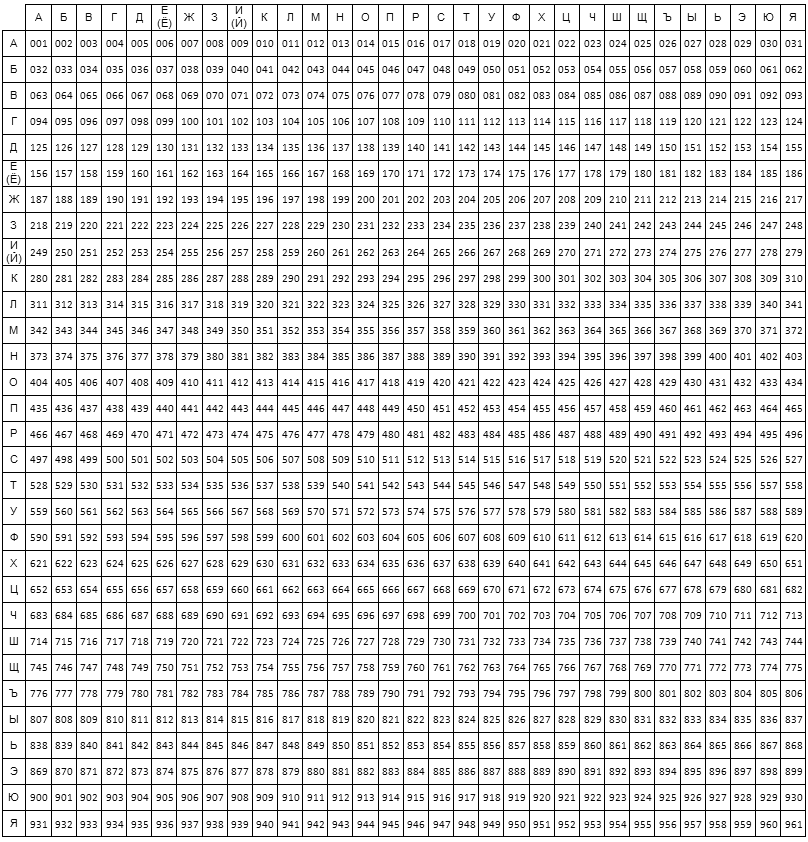


Рисунок 3 — Частота появления букв при шифровании



Рисунок 4 — Частота появления букв при дешифровании

Шифр Порты:

Рисунок 5 – шифр Порты

Далее будет приведен график зависимостей количества символов от времени при разном способе шифрования и дешифрования. График представлен на рисунке 6.

В данном графике линия оранжевого и серого цвета соответствуют способу шифрования и дешифрования с помощью шифра на основе аффинной системы подстановок Цезаря. А линии синего и желтого цвета – способу шифрования и дешифрования с помощью таблицы Трисемуса соответственно.

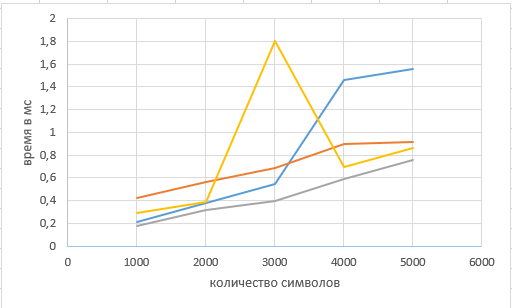


Рисунок 6 — График зависимостей:

Оранжевый – шифрование шифра на основе соотношений при *k* = 7;

Синий – шифрование с помощью таблицы Трисемуса;

Серый – дешифрование на основе соотношений при *k* = 7;

Желтый – дешифрование с помощью таблицы Трисемуса.

**Вывод**: Была произведена работа с разными шифрами, такими как: шифр Трисемуса и шифр соотношения при *k* = 7.