Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление 09.03.01 – «Информатика и вычислительная техника»

Дисциплина: «Защита информации»

Профиль: «Автоматизированные системы обработки информации и управления»

Семестр 6

ОТЧЕТ

по лабораторной работе №1

Тема: «Шифры перестановки и замены»

Вариант 16

Выполнил: студент группы АСУ-19-1б

Шеретов М.А. \_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ИТАС

Шереметьев В. Г. \_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_

Пермь, 2022

**ЦЕЛЬ РАБОТЫ**

Получить практические навыки по применению шифров перестановки и шифров простой замены.

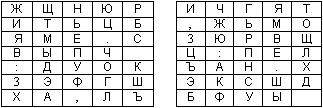
**ЗАДАНИЕ**

**Вариант №16.** Реализовать шифрование текстового сообщения, используя шифр «Уистсона».

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

В 1854 г. англичанин Чарльз Уитстон разработал новый метод шифрования биграммами, который называют «двойным квадратом». Свое название этот шифр получил по аналогии с по-либианским квадратом. Шифр Уитстона открыл новый этап в истории развития криптографии. В отличие от полибианского шифр «двойной квадрат» использует сразу две таблицы, размещенные по одной горизонтали, а шифрование идет биграммами, как в шифре Плейфейра. Эти не столь сложные модификации привели к появлению на свет качественно новой криптографической системы ручного шифрования. Шифр «двойной квадрат» оказался очень надежным и удобным и применялся Германией даже в годы второй мировой войны.

Поясним процедуру шифрования этим шифром на примере. Пусть имеются две таблицы со случайно расположенными в них русскими алфавитами. Перед шифрованием исходное сообщение разбивают на биграммы. Каждая биграмма шифруется отдельно. Первую букву биграммы находят в левой таблице, а вторую букву - в правой таблице. Затем мысленно строят прямоугольник так, чтобы буквы биграммы лежали в его противоположных вершинах. Другие две вершины этого прямоугольника дают буквы биграммы шифртекста.



Две таблицы со случайно расположенными символами русского алфавита  
для шифра «двойной квадрат»

Предположим, что шифруется биграмма исходного текста ИЛ. Буква И находится в столбце 1 и строке 2 левой таблицы. Буква Л находится в столбце 5 и строке 4 правой таблицы. Это означает, что прямоугольник образован строками 2 и 4, а также столбцами 1 левой таблицы и 5 правой таблицы. Следовательно, в биграмму шифртекста входят буква О, расположенная в столбце 5 и строке 2 правой таблицы, и буква В, расположенная в столбце 1 и строке 4 левой таблицы, т.е. получаем биграмму шифртекста ОВ.

Если обе буквы биграммы сообщения лежат в одной строке, то и буквы шифртекста берут из этой же строки. Первую букву биграммы шифртекста берут из левой таблицы в столбце, соответствующем второй букве биграммы сообщения. Вторая же буква биграммы шифртекста берется из правой таблицы в столбце, соответствующем первой букве биграммы сообщения. Поэтому биграмма сообщения ТО превращается в биграмму шифртекста ЖБ. Аналогичным образом шифруются все биграммы сообщения:

Сообщение ПР ИЛ ЕТ АЮ \_Ш ЕС ТО ГО

Шифртекст ПЕ ОВ ЩН ФМ ЕШ РФ БЖ ДЦ

Шифрование методом «двойного квадрата» дает весьма устойчивый к вскрытию и простой в применении шифр. Взламывание шифртекста «двойного квадрата» требует больших усилий, при этом длина сообщения должна быть не менее тридцати строк.

**ХОД РАБОТЫ**

На рисунке 3 представлена форма и пример работы, в первое поле которой вводится шифруемое сообщение. Нижнее поле можно заполнить для расшифровки шифрсообщения.

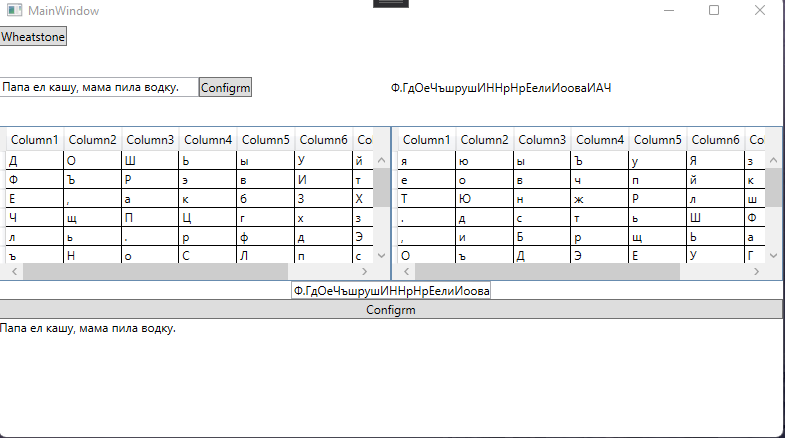
****

Рисунок 3 – Форма для шифрования c примером раблты

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

using System;

using System.Collections.Generic;

using System.Collections;

using System.Text;

using System.Windows;

namespace WpfApp1

{

public class WheatstoneEncryption

{

public char[,] LeftTable

{

get; private set;

}

public char[,] RightTable

{

get; private set;

}

private Random Random = new Random();

public WheatstoneEncryption()

{

LeftTable = new char[8,9];

RightTable = new char[8, 9];

List<char> LeftList = new List<char>();

List<char> RightList = new List<char>();

for (int i = 0; i < 64; i++)

{

LeftList.Add((char)(1040 + i));

RightList.Add((char)(1040 + i));

}

LeftList.Add('.');

RightList.Add('.');

LeftList.Add(',');

RightList.Add(',');

LeftList.Add('?');

RightList.Add('?');

LeftList.Add('!');

RightList.Add('!');

LeftList.Add(' ');

RightList.Add(' ');

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 9; j++)

{

if (LeftList.Count == 0)

continue;

var lIndex = Random.Next(LeftList.Count - 1);

LeftTable[i, j] = LeftList[lIndex];

LeftList.RemoveAt(lIndex);

var rIndex = Random.Next(RightList.Count - 1);

RightTable[i, j] = RightList[rIndex];

RightList.RemoveAt(rIndex);

}

}

}

private (Point,Point) GetCharCoordinate(char c1, char c2)

{

Point fPoint;

Point sPoint;

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 9; j++)

{

if (LeftTable[i, j] == c1)

{

fPoint = new Point(i, j);

}

}

}

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 9; j++)

{

if (RightTable[i, j] == c2)

{

sPoint = new Point(i, j);

}

}

}

return (fPoint, sPoint);

}

private (Point, Point) GetCharCoordinateForDecrypt(char c1, char c2)

{

Point fPoint;

Point sPoint;

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 9; j++)

{

if (RightTable[i, j] == c1)

{

fPoint = new Point(i, j);

}

}

}

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 9; j++)

{

if (LeftTable[i, j] == c2)

{

sPoint = new Point(i, j);

}

}

}

return (fPoint, sPoint);

}

public string Encrypt(string text)

{

try

{

var outputRes = "";

for (int i = 1; i < text.Length; i+=2)

{

char fCh = text[i - 1];

char sCh = text[i];

var coords = GetCharCoordinate(fCh, sCh);

outputRes += RightTable[(int)coords.Item2.X, (int)coords.Item1.Y];

outputRes += LeftTable[(int)coords.Item1.X, (int)coords.Item2.Y];

}

return outputRes;

}

catch (Exception)

{

return null;

}

}

public string Decrypt(string text)

{

try

{

var outputRes = "";

for (int i = 1; i < text.Length; i += 2)

{

char fCh = text[i - 1];

char sCh = text[i];

var coords = GetCharCoordinateForDecrypt(fCh, sCh);

outputRes += LeftTable[(int)coords.Item2.X, (int)coords.Item1.Y];

outputRes += RightTable[(int)coords.Item1.X, (int)coords.Item2.Y];

}

return outputRes;

}

catch (Exception)

{

return null;

}

}

}

}