

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

Лабораторная работа №4

по дисциплине: «Методы и средства криптографической защиты информации»

На тему «Шифрование с использованием системы Вижинера и шифра «двойной квадрат» Уитстона.»

Выполнил обучающийся гр. ВКБ42

Михайлов Александр Сергеевич

          Проверила:

Сафарьян О.А

Ростов-на-Дону

2024

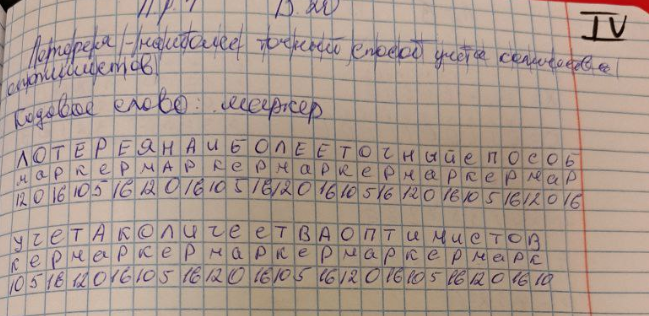
Цель работы: Формирование умений шифрования с использованием системы Вижинера и шифра «двойной квадрат» Уитстона.

Задание 1:

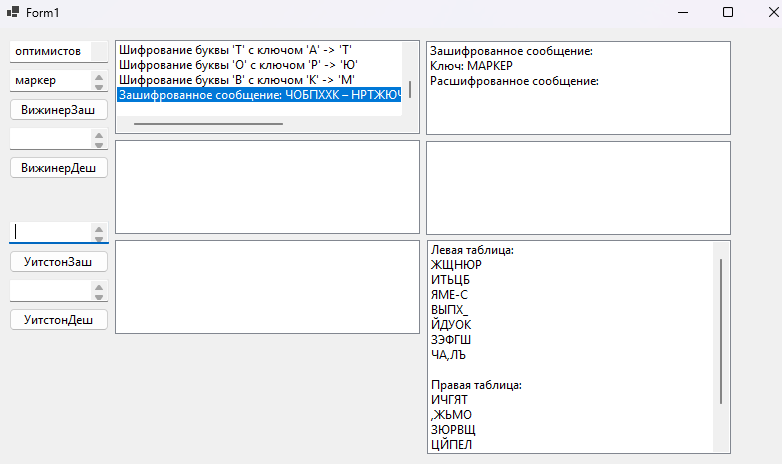
Используя систему Вижинера, зашифруйте сообщения. Текст сообщения и ключевое слово должны соответствовать варианту задания лабораторной работы 3.  
Сообщение: Лотерея – наиболее точный способ учета количества оптимистов

Ключ: Маркер

Ручная реализация:



Программная реализация:

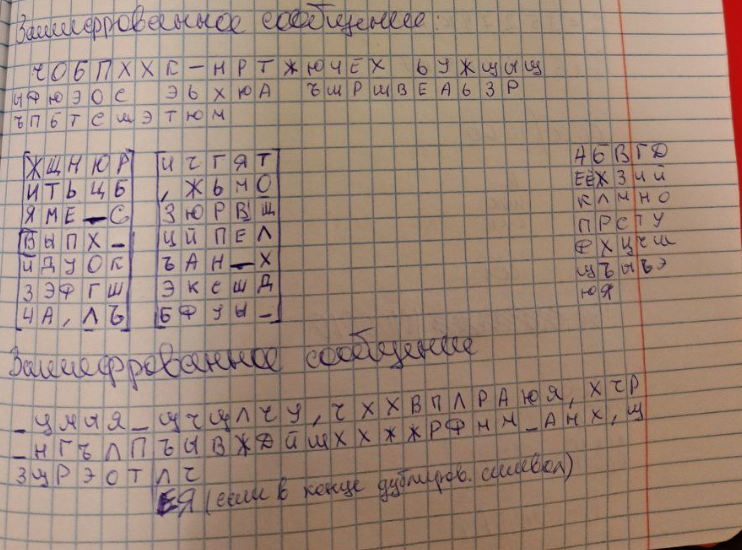


Задание 2

Используя шифр «двойной квадрат» Уитстона и шифрующие таблицы, представленные на рис. 4.1, выполните шифрование сообщения из задания лабораторной работы 3.

Сообщение: Лотерея – наиболее точный способ учета количества оптимистов

Ручная реализация:



Вывод:

В ходе выполнения лабораторной работы были сформированы умения шифрования с использованием системы Вижинера и шифра «двойной квадрат» Уитстона.

Листинг кода

Задание 1:

using System.Text;

using System.Windows.Forms;

namespace ШифрВиженера

{

public partial class Form1 : Form

{

private const string LeftTable = "ЖЩНЮРИТЬЦБЯМЕ-СВЫПХ\_ЙДУОКЗЭФГШЧА,ЛЪ";

private const string RightTable = "ИЧГЯТ,ЖЬМОЗЮРВЩЦЙПЕЛЪАН-ХЭКСШДБФУЫ\_";

public Form1()

{

InitializeComponent();

DisplayTables();

}

private void buttonEncrypt\_Click(object sender, EventArgs e)

{

string message = textBoxMessage.Text.ToUpper();

string key = textBoxKey.Text.ToUpper();

listBoxSteps.Items.Clear();

listBoxSteps.Items.Add("Исходное сообщение: " + message);

listBoxSteps.Items.Add("Ключ: " + key);

string encryptedMessage = EncryptVigenere(message, key);

listBoxSteps.Items.Add("Зашифрованное сообщение: " + encryptedMessage);

}

private string EncryptVigenere(string message, string key)

{

StringBuilder encryptedMessage = new StringBuilder();

int keyIndex = 0;

foreach (char c in message)

{

if (char.IsLetter(c))

{

int messageChar = c - 'А';

int keyChar = key[keyIndex] - 'А';

int encryptedChar = (messageChar + keyChar) % 33;

char encryptedLetter = (char)(encryptedChar + 'А');

encryptedMessage.Append(encryptedLetter);

listBoxSteps.Items.Add($"Шифрование буквы '{c}' с ключом '{key[keyIndex]}' -> '{encryptedLetter}'");

keyIndex = (keyIndex + 1) % key.Length;

}

else

{

encryptedMessage.Append(c); // Сохраняем пробелы неизменными

}

}

return encryptedMessage.ToString();

}

private void DisplayTables()

{

listBox1.Items.Clear();

listBox1.Items.Add("Левая таблица:");

for (int i = 0; i < 7; i++)

{

listBox1.Items.Add(LeftTable.Substring(i \* 5, 5));

}

listBox1.Items.Add("");

listBox1.Items.Add("Правая таблица:");

for (int i = 0; i < 7; i++)

{

listBox1.Items.Add(RightTable.Substring(i \* 5, 5));

}

}

private void buttonEncrypt2\_Click(object sender, EventArgs e)

{

string message = textBoxMessage2.Text.ToUpper();

listBoxSteps2.Items.Clear();

listBoxResult2.Items.Clear();

listBoxSteps2.Items.Add("Исходное сообщение: " + message);

string encryptedMessage = EncryptDoubleSquare(message);

listBoxResult2.Items.Add("Зашифрованное сообщение: " + encryptedMessage);

}

private string EncryptDoubleSquare(string message)

{

StringBuilder encryptedMessage = new StringBuilder();

message = message.Replace(" ", "\_"); // Заменяем пробелы на \_

// Проверяем, четное ли сообщение, и если нет, добавляем "\_" в конец

if (message.Length % 2 != 0)

{

message += "\_";

}

List<string> bigrams = SplitIntoBigrams(message);

foreach (string bigram in bigrams)

{

char firstChar = bigram[0];

char secondChar = bigram[1];

int firstRow, firstCol, secondRow, secondCol;

(firstRow, firstCol) = FindPosition(firstChar, LeftTable);

(secondRow, secondCol) = FindPosition(secondChar, RightTable);

char encryptedFirstChar, encryptedSecondChar;

if (firstRow == secondRow) // Обе буквы в одной строке

{

encryptedFirstChar = RightTable[firstRow \* 5 + secondCol];

encryptedSecondChar = LeftTable[secondRow \* 5 + firstCol];

}

else // Прямоугольник

{

encryptedFirstChar = RightTable[firstRow \* 5 + secondCol];

encryptedSecondChar = LeftTable[secondRow \* 5 + firstCol];

}

listBoxSteps2.Items.Add($"Биграмма '{firstChar}{secondChar}' -> '{encryptedFirstChar}{encryptedSecondChar}'");

encryptedMessage.Append(encryptedFirstChar);

encryptedMessage.Append(encryptedSecondChar);

}

return encryptedMessage.ToString();

}

private (int, int) FindPosition(char c, string table)

{

int index = table.IndexOf(c);

if (index == -1)

{

throw new ArgumentException($"Символ '{c}' не найден в таблице.");

}

int row = index / 5;

int col = index % 5;

return (row, col);

}

private List<string> SplitIntoBigrams(string message)

{

List<string> bigrams = new List<string>();

for (int i = 0; i < message.Length; i += 2)

{

if (i + 1 < message.Length)

{

bigrams.Add(message.Substring(i, 2));

}

else

{

bigrams.Add(message[i] + "Х"); // Добавляем фиктивную букву, если длина нечетная

}

}

return bigrams;

}

private void button1\_Click(object sender, EventArgs e)

{

string encryptedMessage = textBoxEncryptedMessage.Text.ToUpper();

string message = textBoxMessage2.Text.ToUpper();

listBoxDecryptSteps.Items.Clear();

listBoxResult2.Items.Clear();

listBoxDecryptSteps.Items.Add("Зашифрованное сообщение: " + encryptedMessage);

string decryptedMessage = DecryptDoubleSquare(encryptedMessage);

listBoxResult2.Items.Add("Расшифрованное сообщение: " + message);

}

private string DecryptDoubleSquare(string encryptedMessage)

{

StringBuilder decryptedMessage = new StringBuilder();

List<string> bigrams = SplitIntoBigrams(encryptedMessage);

foreach (string bigram in bigrams)

{

char firstChar = bigram[0];

char secondChar = bigram[1];

int firstRow, firstCol, secondRow, secondCol;

(firstRow, firstCol) = FindPosition(firstChar, RightTable);

(secondRow, secondCol) = FindPosition(secondChar, LeftTable);

char decryptedFirstChar = LeftTable[firstRow \* 5 + secondCol];

char decryptedSecondChar = RightTable[secondRow \* 5 + firstCol];

listBoxDecryptSteps.Items.Add($"Биграмма '{firstChar}{secondChar}' -> '{decryptedFirstChar}{decryptedSecondChar}'");

decryptedMessage.Append(decryptedFirstChar);

decryptedMessage.Append(decryptedSecondChar);

}

return decryptedMessage.ToString();

}

private void buttonDecrypt2\_Click(object sender, EventArgs e)

{

string encryptedMessage = textBoxEncryptedMessage2.Text.ToUpper();

string key = textBoxKey.Text.ToUpper();

listBoxDecryptSteps2.Items.Clear();

listBoxDecryptSteps2.Items.Add("Зашифрованное сообщение: " + encryptedMessage);

listBoxDecryptSteps2.Items.Add("Ключ: " + key);

string decryptedMessage = DecryptVigenere(encryptedMessage, key);

listBoxDecryptSteps2.Items.Add("Расшифрованное сообщение: " + decryptedMessage);

}

private string DecryptVigenere(string encryptedMessage, string key)

{

StringBuilder decryptedMessage = new StringBuilder();

int keyIndex = 0;

foreach (char c in encryptedMessage)

{

if (char.IsLetter(c))

{

int encryptedChar = c - 'А';

int keyChar = key[keyIndex] - 'А';

int decryptedChar = (encryptedChar - keyChar + 33) % 33;

char decryptedLetter = (char)(decryptedChar + 'А');

decryptedMessage.Append(decryptedLetter);

listBoxDecryptSteps.Items.Add($"Расшифрование буквы '{c}' с ключом '{key[keyIndex]}' -> '{decryptedLetter}'");

keyIndex = (keyIndex + 1) % key.Length;

}

else

{

decryptedMessage.Append(c); // Сохраняем пробелы неизменными

}

}

return decryptedMessage.ToString();

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

}