|  |  |
| --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования «Полоцкий государственный университет» | |
|  | Факультет информационных технологий  Кафедра технологий программирования |
| Лабораторная работа №4 по курсу «Теория информации»  «Криптоанализ методов простой подстановки» | |
| Выполнил | Студент гр. 21-ИТ-1  Макеёнок Д.И. |
| Проверила | Васильева Д.М. |
| Полоцк, 2023г. | |

**Ход работы**

Простейшие шифры подстановки (substitution) реализуют замену каждого символа исходного текста на один из символов алфавита шифротекста. В общем случае подстановочный шифр описывается таблицей подстановки, состоящей из двух строк и n столбцов. Количество столбцов таблицы подстановки соответствует количеству различных символов в алфавите исходного текста. Верхняя строка таблицы подстановки содержит все возможные символы исходного текста, а нижняя – соответствующие им символы шифротекста.

**Задание:**

1. Реализовать программное средство, осуществляющее шифрование и дешифрование текстового файла, содержащего текст на заданном языке (нечетный номер в списке – английский язык, четный номер – русский).

2. Реализовать программное средство, осуществляющее криптоанализ зашифрованного по методу Виженера текста. Для криптоанализа использовать тест Касиски.

3. Провести экспериментальное исследование зависимости вероятности успешного проведения атаки по методу Касиски от длины шифротекста.

4. Провести экспериментальное исследование зависимости вероятности успешного проведения атаки по методу Касиски от длины использованного при шифровании ключевого слова.

Листинг 1 - реализация программы:

using Microsoft.VisualBasic;

using System.Text;

string alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

string Crypt(string input, string key, Dictionary<char, string> shift)

{

var result = string.Empty;

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

{

if (input[i] == ' ')

{

result += ' ';

continue;

}

result += shift[key[i % key.Length]][alphabet.IndexOf(input[i])].ToString();

}

return result;

}

Dictionary<char, string> BuildShift(string input)

{

var shift = new Dictionary<char, string>();

for (var i = 0; i < input.Length; i++)

{

shift.Add(input[i], string.Empty);

var shiftIndex = alphabet.IndexOf(input[i]);

for (var j = 0; j < alphabet.Length; j++)

{

if (shiftIndex == alphabet.Length)

shiftIndex = 0;

shift[input[i]] = shift[input[i]] + alphabet[shiftIndex];

shiftIndex++;

}

}

return shift;

}

Dictionary<string, List<int>> RepeatedBlocks(string input)

{

var repeatedMap = new Dictionary<string, List<int>>();

for (var i = 0; i < input.Length - 3; i++)

{

string block = input.Substring(i, 3);

if (!repeatedMap.ContainsKey(block))

repeatedMap[block] = new List<int>();

repeatedMap[block].Add(i);

}

return repeatedMap;

}

List<int> CalcKeyLengths(Dictionary<string, List<int>> repeatedMap)

{

var keyLengths = new List<int>();

foreach (var map in repeatedMap)

{

var positions = map.Value;

for (var i = 0; i < positions.Count - 1; i++)

{

for (var j = i + 1; j < positions.Count; j++)

{

var distance = positions[j] - positions[i];

if (distance > 0)

keyLengths.Add(distance);

}

}

}

return keyLengths;

}

var cryptedText =

Crypt(

"DJEOIJOI JEOIDJEOIHFOJED EHFIOEJOIEHFOEJDOI JEDHOEIJDOEJDEHFEJDJNAONOEJDOAD", //A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

"MOUSE", //M N O P Q R S T U V W X Y Z A B C D E F G H I J K L

BuildShift("MOUSE"));

var lengths = CalcKeyLengths(RepeatedBlocks(cryptedText));

Console.WriteLine($"Зашифрованый текст: {cryptedText}");

foreach (var length in lengths)

Console.WriteLine($"Возможная длинна ключа: {length}");

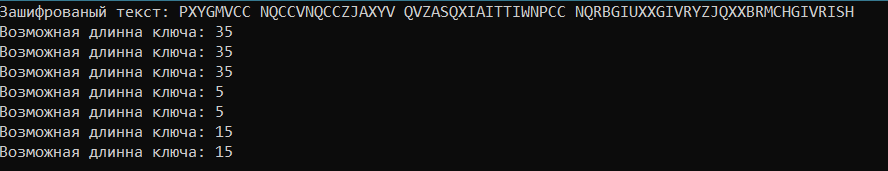


Рисунок 1. Результат работы программы