МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Чувашский государственный университет имени И.Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

Лабораторная работа №1

По дисциплине “Защита информация”

на тему:

«Шифрование данных методом перестановки»

Вариант 6

Выполнил: ст. гр. ИВТ-11-21

Еремеев С. А.

Проверил: канд. техн. наук. Ковалев С.В.

Чебоксары 2024

## Теоретическая часть

## Цель работы

Целью работы является знакомство с классическим криптографическим алгоритмом - алгоритмом шифрования данных при помощи перестановки.

## Основные сведения

Перестановочное шифрование, также как и подстановочное, относится к классической криптографии (см. лабораторную работу №1): здесь тоже используется только одна единица секретной информации - ключ, знание которого позволяет отправителю зашифровать информацию в шифртекст, а получателю - расшифровать его.

Перестановочное шифрование основывается на использовании некоторой взаимно однозначной функции , где *M*={1,...,*m*} - множество номеров позиций символов в *m*-символьном блоке открытого текста. В процессе шифрования блок открытого текста, имеющий вид *x*=*v1v2..vm*, где *vk*, *k*=1..*m*, означает символ алфавита *V*, стоящий на *k*-ой позиции, заменяется блоком *y*=*w1w2 ..wm*, где *wj*=*vq*, . Дешифрование состоит в обратной замене блока *y*=*w1w2 ..wm* на *v1v2 ..vr ..vm*, где *vr=wj*, .

Например, пусть алфавит *V* = {0,1, .., 9, <пробел>, A, B, .., Z, a, b, .., z, А, Б, .., Я, а, б, .., я}, длина блока шифрограммы и блока открытого текста  
*m* = 15. Допустим, что необходимо зашифровать открытый текст X = “Произвольный блок открытого текста”. Разобьем открытый текст X на *m*-символьные (пятнадцатисимвольные в нашем примере) блоки: “Произвольный\_бл”, ”ок\_открытого\_те”, “кста\_\_\_\_\_\_\_\_\_\_\_”. Пробелы обозначены символом \_, при необходимости последний блок может быть дополнен с правой стороны необходимым количеством пробелов. Если пробел не входит в алфавит языка, то его функцию (функцию разделительного элемента) может выполнять любой другой символ алфавита, если стороны, обменивающиеся сообщениями достигли соответствующей договоренности.

Для шифрования необходимо иметь функцию *CN*, ставящую *i*-й позиции символа в блоке открытого текста *CN*(*i*)-ю позицию в блоке шифртекста. Такая функция может быть задана, например, при помощи таблицы:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CN(i)* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| *i* | 13 | 5 | 1 | 8 | 10 | 12 | 2 | 14 | 4 | 6 | 3 | 15 | 7 | 11 | 9 |

В каждом пятнадцатисимвольном блоке шифртекста, в нашем примере, на первой позиции будет стоять символ, занимающий 13-ю позицию в блоке открытого текста, на второй позиции - символ, занимающий 5-ю позицию в блоке открытого текста, и т.д. Таким образом, наш открытый текст даст следующие блоки шифртекста: “\_зПлнйрбиволоыь”, “\_тоыоокток\_ергт”, “\_\_к\_\_\_с\_а\_т\_\_\_\_”, или целиком: “\_зПлнйрбиволоыь\_тоыоокток\_ергт\_\_к\_\_\_с\_а\_т\_\_\_\_”

Поскольку функция *CN* является взаимно однозначной, эта же таблица используется и для дешифрации шифртекста.

Очевидно, что размер таблицы, задающей функцию шифрования, определяется принятым размером блока открытого текста: всего необходимо определить *m* значений функции.

**Практическая часть**

## Индивидуальное задание

| Задание | | |
| --- | --- | --- |
| № п.п. | *V* | *m* |
| 6 | {А,Б,..,Я,<пробел>} | 17 |

**Функция шифрования**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CN(i)* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| *i* | 2 | 1 | 4 | 3 | 6 | 5 | 8 | 7 | 10 | 9 | 12 | 11 | 14 | 13 | 16 | 15 | 17 |

Функция шифрования меняет соседние символы местами.

**Текст программы, реализующей индивидуальное задание**

Сначала мы создаем пустую коллекцию и добавляем в нее исходный текст. Далее с помощью функции происходит шифрование текста.

Console.Write("Введите текст: ");

string inputText = Console.ReadLine().ToUpper();

string text = inputText.Replace(' ', '\_');

List<char> list = new List<char>(text.ToCharArray());

while (list.Count < 17)

{

list.Add('\_');

}

static string Encrypt(string text)

{

char[] chars = text.ToCharArray();

for (int i = 0; i < chars.Length - 1; i += 2)

{

char temp = chars[i];

chars[i] = chars[i + 1];

chars[i + 1] = temp;

}

return new string(chars);

}

static string Decrypt(string text)

{

char[] chars = text.ToCharArray();

for (int i = 0; i < chars.Length - 1; i += 2)

{

char temp = chars[i];

chars[i] = chars[i + 1];

chars[i + 1] = temp;

}

return new string(chars);

}

text = new string(list.ToArray());

Console.WriteLine($"Исходный текст: {text}");

string encryptedText = Encrypt(text);

Console.WriteLine($"Зашифрованный текст: {encryptedText}");

string decryptedText = Decrypt(encryptedText);

decryptedText = decryptedText.Replace('\_', ' ');

Console.WriteLine($"Расшифрованный текст: {decryptedText}");

**Пример открытого текста и соответствующей ему шифрограммы**

Открытый текст: ЕРЕМЕЕВ СЕРГЕЙ

Шифртекст: РЕМЕЕЕ\_ВЕСГРЙЕ\_\_\_

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *xi* | Е | Р | Е | М | Е | Е | В |  | С | Е | Р | Г | Е | Й |  |  |  |
| *CV*  *(xi)* | Р | Е | М | Е | Е | Е | \_ | В | Е | С | Г | Р | Й | Е | \_ | \_ | \_ |

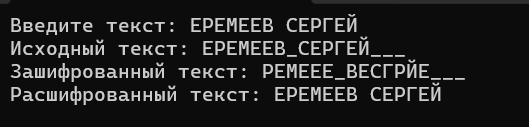


Рисунок –Пример работы программы

Открытый текст: ПРИВЕТ ЧУВАШИЯ

Шифртекст: РПВИТЕ\_\_Ч\_ВУШАЯИ\_

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *xi* | П | Р | И | В | Е | Т |  |  |  | Ч | У | В | А | Ш | И | Я |  |
| *CV*  *(xi)* | Р | П | В | И | Е | Е | \_ | \_ | Ч | \_ | В | У | Ш | А | Я | И | \_ |

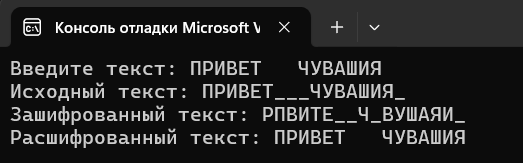


Рисунок 2 –Пример работы программы

**Вывод**

Ознакомился с классическим криптографическим алгоритмом – алгоритмом шифрования данных при помощи подстановки