МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Лабораторная работа №1

По предмету «Защита информации»

«Методы перестановки»

Выполнил:

студент группы ПО(б) –81

Пшеничный Д. О.

Проверил:

Резак Е. В.

Хабаровск – 2021г.

**Цель:**

Реализовать шифрование своих фамилии, имени и отчества с помощью одного, двух ключей и по алгоритму Цезаря (по примеру 1, 2 и 3). Реализацию осуществить вручную, с помощью любого приложения, программно на любом языке программирования.

Составить отчет с пошаговыми описанием ручного расчета, листингом программы с результатами работы программы.

К отчету приложить рабочий программный продукт.

**Теория:**

При шифровании перестановкой с одним ключом K длиной N, исходный текст разбивается на группы по N символов, и в пределах этих групп на K[i] место встаёт i-ый символ.

При использовании двух ключей (K1 и K2, длиной соответственно N1 и N2), исходный текст разбивается на группы по N2 символов. Из этих групп составляется таблица, в которой i-ая строка встаёт на позицию K1[i]. Далее посредством чтения столбцов в порядке, установленном ключом K2, составляется зашифрованный текст.

При использовании шифра Цезаря с ключом K, буквы исходного текста заменяются на буквы, находящиеся на позиции +K в алфавите.

**Ручной расчет:**

**Перестановка с одним ключом**

Исходный текст: “пшеничный данил олегович”

Ключ: 8-1-7-5-6-4-3-2

Группы символов: “пшеничны”, “й данил ”, “олегович ”

Перестановка: “ыпничнеш”, “да ынй”, “шепинл”

Изображение выглядит как текст

Автоматически созданное описание

**Перестановка с двумя ключами**

Исходный текст: “пшеничный данил олегович”

Ключ 1: 6-5-3-1-2-4

Ключ 2: 5-2-1-3-4

Группы символов: “пшени”, “чный ”, “данил”, “ олег”, “ович\_”, “\_\_\_\_\_”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| K1\K2 | 1 | 2 | 3 | 4 | 5 |
| 1 |  | О | Л | Е | Г |
| 2 | О | В | И | Ч | \_ |
| 3 | Д | А | Н | И | Л |
| 4 | \_ | \_ | \_ | \_ | \_ |
| 5 | Ч | Н | Ы | Й |  |
| 6 | П | Ш | Е | Н | И |

Изображение выглядит как текст

Автоматически созданное описание

**Скриншот работы программы:**

**Изображение выглядит как текст

Автоматически созданное описание**

**Код программы:**

internal class Program  
{  
 static string alphabet = "абвгдежзийклмнопрстуфхцчшщъыьёюя\_";  
  
 static string value = "Данил";  
   
 public static void Main(string[] args)  
 {  
 Console.WriteLine("Матричный метод:");  
 MatrixMethod(value);  
 Console.WriteLine("\nМетод укладки ранца:");  
 BackpackMethod(value);  
 Console.WriteLine("\nМетод полиномов:");  
 PolinomMethod(value);  
 Console.WriteLine("\nЭкспоненциальный метод:");  
 ExpoMethod(value);  
 }  
  
 private static void MatrixMethod(string inputValue)  
 {  
 int[,] cMatr = new int[,]  
 {  
 {2, 2, 3},  
 {1, 2, 3},  
 {1, 2, 3}  
 };  
  
 inputValue = inputValue.ToLower();  
 inputValue = ExtendToTargetLength(inputValue, 3);  
  
 int[] result = new int[inputValue.Length];   
   
 for (int i = 0; i < inputValue.Length; i += 3)  
 {  
 int[] X = new[] {0, 0, 0};  
 for (int j = 0; j < 3; j++)  
 {  
 X[j] = alphabet.IndexOf(inputValue[i + j]) + 1;  
 }  
  
 for (int j = 0; j < 3; j++)  
 {  
 result[i + j] = cMatr[j, 0] \* X[0] + cMatr[j, 1] \* X[1] + cMatr[j, 2] \* X[2];  
 }  
 }  
  
 foreach (var i in result)  
 {  
 Console.Write(i + " ");  
 }  
 }  
  
 private static void BackpackMethod(string inputValue)  
 {  
 int[] cVec = new[] {1, 3, 5, 7, 11, 15};  
 int[] result = new int[inputValue.Length];  
   
 inputValue = inputValue.ToLower();  
  
 for (int i = 0; i < inputValue.Length; i++)  
 {  
 int chValue = 0;  
 var chBin = ConvertToBinary(alphabet.IndexOf(inputValue[i]) + 1, cVec.Length);  
   
 for (int j = 0; j < cVec.Length; j++)  
 {  
 chValue += cVec[j] \* chBin[j];  
 }  
  
 result[i] = chValue;  
 }  
  
 foreach (var i in result)  
 {  
 Console.Write(i + " ");  
 }  
 *// Console.WriteLine();* }  
 private static void PolinomMethod(string inputValue)  
 {  
 inputValue = inputValue.ToLower();  
 int p = 991;  
  
 List<int> resultValue = new List<int>();  
  
 foreach (var i in inputValue)  
 {  
 int iSum = 0;  
  
 int maxVal = 4;  
   
 for (int j = 1; j <= maxVal; j++)  
 {  
 iSum += j \* (int)Math.Pow((alphabet.IndexOf(i) + 1), maxVal - j);  
 }  
   
 resultValue.Add(iSum % p);  
 }  
  
 foreach (var i in resultValue)  
 {  
 Console.Write(i + " ");  
 }  
 }  
  
 private static void ExpoMethod(string inputValue)  
 {  
 inputValue = inputValue.ToLower();  
   
 int a = 5;  
 int p = 991;  
  
 List<long> resultValue = new List<long>();  
  
 foreach (var i in inputValue)  
 {  
 resultValue.Add((long)Math.Pow(a, alphabet.IndexOf(i) + 1) % p);  
 }  
  
 foreach (var i in resultValue)  
 {  
 Console.Write(i + " ");  
 }  
 }  
 private static string ExtendToTargetLength(string input, int partLength)  
 {  
 while (input.Length % partLength != 0)  
 {  
 input += "\_";  
 }  
 return input;  
 }  
  
 private static int[] ConvertToBinary(int value, int vecLength)  
 {  
 int[] result = new int[vecLength];  
  
 string stringRepr = "";  
  
 while (value > 0)  
 {  
 stringRepr += value % 2;  
 value /= 2;  
 }  
  
 string tempString = stringRepr;  
 stringRepr = "" + tempString[0];  
 for (int i = 1; i < tempString.Length; i++)  
 {  
 stringRepr = stringRepr.Insert(0, tempString[i] + "");  
 }  
  
 while (stringRepr.Length < vecLength)  
 {  
 stringRepr = stringRepr.Insert(0, "0");  
 }  
  
 for (int i = 0; i < stringRepr.Length; i++)  
 {  
 string t = "" + stringRepr[i];  
 result[i] = int.Parse(t);  
 }  
   
 *// Console.WriteLine(stringRepr);* return result;  
 }  
}

**Вывод:** в ходе выполнения лабораторной работы были изучены алгоритмы шифрования на основе односторонней функции.