Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №7

по дисциплине "Администрирование и безопасность информационных систем"

ТЕМА РАБОТЫ:

Шифрование с помощью аналитических преобразований

Выполнил:

студент гр. ПРИ-120

Парахин К.В.

Принял:

Доцент кафедры ИСПИ

Курочкин С.В.

Владимир 2023 г.

Цель работы:

Освоить на практике применение алгоритма шифрования с помощью аналитических преобразований.

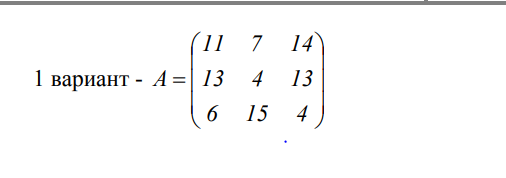
Выполнение работы:

Задание.

Зашифровать описанным методом следующее выражение: «Помехоустойчивое кодирование – это кодирование с возможностью восстановления потерянных или ошибочно принятых данных». В качестве ключа использовать следующую матрицу.

Индивидуальный вариант №9

9 % 4 = 1



Листинг программы:

Класс Program:

namespace AnalyticCrypter;

public class Program

{

public static void Main()

{

Console.WriteLine("Введите сообщение для аналитического шифрования");

var text = Console.ReadLine();

if (string.IsNullOrWhiteSpace(text))

{

throw new ArgumentException(nameof(text));

}

var encryptedMessage = AnalyticCrypter.Encrypt(text);

Console.WriteLine($"Сообщение после шифрования = {encryptedMessage}");

}

}

Класс AnalyticCrypter:

using System.Text;

namespace AnalyticCrypter;

public static class AnalyticCrypter

{

public static int[] Multiply(int[][] matrixA, int[] matrixB)

{

var colsCountA = matrixA[0].Length;

var colsCountB = 1; // кол-во столбцов прямоугольной матрицы у вектора

var rowsCountA = matrixA.Length;

var rowsCountB = matrixB.Length;

if (colsCountA != rowsCountB)

{

throw new InvalidOperationException("Матрицы нельзя перемножить из-за неправильной размерности");

}

\_resultMatrix = new int[rowsCountA];

for (var i = 0; i < colsCountB; i++)

{

for (var j = 0; j < rowsCountA; j++)

{

var temp = 0;

for (var k = 0; k < rowsCountB; k++)

{

temp += matrixA[j][k] \* matrixB[k];

}

\_resultMatrix[j] = temp;

}

}

return \_resultMatrix;

}

public static List<int> CreateVector(string word)

{

var length = \_defaultMatrix.Length;

if (word.Length != length)

{

throw new InvalidOperationException("Длина слова не равна заданной длине вектора");

}

var vector = new List<int>();

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

{

vector.Add(\_defaultAlphabet.IndexOf(word[i]));

}

return vector;

}

public static string Encrypt(string message)

{

var newMessage = new StringBuilder("");

message = message.ToUpper();

foreach (var symbol in message)

{

if (\_defaultAlphabet.Contains(symbol))

{

newMessage.Append(symbol);

}

}

var length = \_defaultMatrix.Length;

var iterations = Math.DivRem(newMessage.Length, length, out var remainder);

if (remainder > 0)

{

iterations++;

for (int i = 0; i < length - remainder; i++)

{

newMessage.Append("А"); // добавляю в сообщение символы, чтобы получить кол-во символов кратное длине вектора

}

}

var result = new List<int>();

var mes = newMessage.ToString();

for (var i = 0; i < iterations; i++)

{

var subString = mes.Substring(i \* length, length);

var vector = CreateVector(subString);

var slice = Multiply(\_defaultMatrix, vector.ToArray());

foreach (var number in slice)

{

result.Add(number);

}

}

return string.Join(", ", result);

}

private const string \_defaultAlphabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

private static int[] \_resultMatrix;

public static readonly int[][] \_defaultMatrix = new int[][]

{

new int[] { 11, 7, 14 },

new int[] { 13, 4, 13 },

new int[] { 6, 15, 4 }

};

}

Вывод

В результате выполнения работы я научился выполнять шифрование сообщения путем аналитических преобразований.