Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчёт по лабораторной работе №1

“Шифр Цезаря.”

Выполнил:

студент гр. 953505

Яцына А. М.

Проверил:

Олисейчик В.В.

Минск 2022

# **1.Введение**

Необходимо реализовать программные средства шифрования и дешифрования текстовых файлов при помощи **шифра Цезаря**, (шифра сдвига, кода Цезаря) и **шифра Виженера**.

# **2.Теоретическая часть**

**Шифр Цезаря** – это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом 4 А была бы заменена на Г, Б станет Д, и так далее.

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами модульной арифметики:

**Y = (X + K + N) mod n;**

**X = (Y – K + N) mod n;**

**X** – Символ открытого текста;

**Y** – Символ шифрованного текста;

**N** – Мощность алфавита;

**K** – Ключ;

С точки зрения математики шифр Цезаря является частным случаем аффинного шифра.

Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря.

Поиск индекса символа можно выразить так:

(Мощность алфавита + индекс символа ОТ + индекс символа пароля) % мощность алфавита.

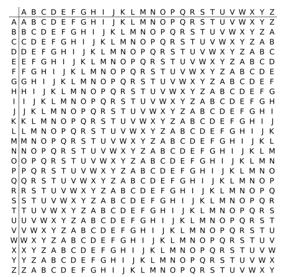


Рис 1. Квадрат Виженера.

# **3.Блок-схема алгоритма**

## **3.1. Шифр Цезаря**

## 

Рис 2. Схема шифра Цезаря.

## **3.2. Шифр Виженера**

## Рис 4. Схема функции GetAllPass

Рис 3. Схема шифра Виженера.

# **4.Результаты работы программы**

*Key for Caesar(digit):*

*3*

*Key word for Vigenere:*

*телефон*

*Text for encrypt:*

*В понедельник вечером.*

*Encrypt with key 3 : Е тсрзжзоярлн езъзусп.*

*Decrypted : В понедельник вечером.*

*Encrypted with key word телефон: ф ЫУвУСЧРзТЭЩ ФЙгЙеЭЪ.*

*Decrypted : В понедельник вечером.*

# **5.Исходный код программы**

*Шифр Цезаря:*

public class Caesar

{

readonly string alf;

public Caesar(string alf = null)

{

var defaultAlf = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

defaultAlf += defaultAlf.ToLower();

this.alf = string.IsNullOrEmpty(alf) ? defaultAlf : alf;

}

private string Encode(string text, int offset)

{

var fullAlf = alf + alf.ToLower();

var alfSize = fullAlf.Length;

var result = "";

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

{

var c = text[i];

var index = fullAlf.IndexOf(c);

if (index == -1)

{

result += c.ToString();

}

else

{

result += fullAlf[(alfSize + index + offset) % alfSize].ToString();

}

}

return result;

}

public string Encrypt(string text, int offset) => Encode(text, offset);

public string Decrypt(string text, int offset) => Encode(text, -offset);

}

*Шифр Виженера:*

public class Vigenere

{

readonly string alf;

public Vigenere(string alf = null)

{

var defaultAlf = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

defaultAlf += defaultAlf.ToLower();

this.alf = string.IsNullOrEmpty(alf) ? defaultAlf : alf;

}

private string GetAllPass(string password, int textLength)

{

var temp = password;

while (temp.Length < textLength)

{

temp += temp;

}

var result = temp.Substring(0, textLength);

return result;

}

private string Encode(string text, string password, bool isEncrypt = true)

{

var allPass = GetAllPass(password, text.Length);

var alfSize = alf.Length;

var result = "";

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

{

var c = text[i];

var index = alf.IndexOf(c);

var cp = allPass[i];

var passIndex = alf.IndexOf(cp);

passIndex \*= isEncrypt ? 1 : -1;

if (index == -1)

{

result += c.ToString();

}

else

{

result += alf[(alfSize + index + passIndex) % alfSize].ToString();

}

}

return result;

}

public string Encrypt(string text, string password) => Encode(text, password);

public string Decrypt(string text, string password) => Encode(text, password, false);

}

Программа:

static void Main(string[] args)

{

var caesar = new Caesar();

var vigenere = new Vigenere();

Console.WriteLine("Key for Caesar(digit):");

var key = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Key word for Vigenere:");

var keyWord = Console.ReadLine();

string text = "";

using (FileStream fstream = File.OpenRead(Directory.GetParent(Environment.CurrentDirectory).Parent.Parent.FullName + "/ms.txt"))

{

byte[] buffer = new byte[fstream.Length];

fstream.Read(buffer, 0, buffer.Length);

text = Encoding.UTF8.GetString(buffer);

Console.WriteLine($"Текст из файла: {text}");

}

var encryptedByCaesar= caesar.Encrypt(text, key);

Console.WriteLine($"Encrypt with {key} : {encryptedByCaesar}");

using (FileStream fstream = new FileStream(Directory.GetParent(Environment.CurrentDirectory).Parent.Parent.FullName + "/res.txt", FileMode.OpenOrCreate))

{

var result = encryptedByCaesar ;

byte[] buffer = Encoding.UTF8.GetBytes(result);

fstream.Write(buffer, 0, buffer.Length);

}

using (FileStream fstream = File.OpenRead(Directory.GetParent(Environment.CurrentDirectory).Parent.Parent.FullName + "/res.txt"))

{

byte[] buffer = new byte[fstream.Length];

fstream.Read(buffer, 0, buffer.Length);

encryptedByCaesar = Encoding.UTF8.GetString(buffer);

}

var decryptedByCaesar = caesar.Decrypt(encryptedByCaesar, key);

Console.WriteLine($"Decrypted : {decryptedByCaesar}");

var encryptedByVigenere = vigenere.Encrypt(text, keyWord);

Console.WriteLine($"Encrypted with key word {keyWord}: {encryptedByVigenere}");

using (FileStream fstream = new FileStream(Directory.GetParent(Environment.CurrentDirectory).Parent.Parent.FullName + "/res.txt", FileMode.OpenOrCreate))

{

var result = encryptedByVigenere;

byte[] buffer = Encoding.UTF8.GetBytes(result);

fstream.Write(buffer, 0, buffer.Length);

}

using (FileStream fstream = File.OpenRead(Directory.GetParent(Environment.CurrentDirectory).Parent.Parent.FullName + "/res.txt"))

{

byte[] buffer = new byte[fstream.Length];

fstream.Read(buffer, 0, buffer.Length);

encryptedByVigenere = Encoding.UTF8.GetString(buffer);

}

var decryptedByVigenere = vigenere.Decrypt(encryptedByVigenere, keyWord);

Console.WriteLine($"Decrypted : {decryptedByVigenere}");

Console.ReadKey();

}

# **6.Вывод**

В данной лабораторной работе были рассмотрены шифры Цезаря и Виженера. Были построены алгоритмы, выполняющие шифрование и дешифрование текстового файла. Шифр Виженера несколько безопаснее шифра Цезаря: в качестве ключа в нем используется слово и его сложно взломать вручную с помощью одного только частотного анализа или перебора.