МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Костромской государственный университет»

(КГУ)

Институт физико-математических и естественных наук

Кафедра защиты информации

Направление подготовки: 10.03.01 Информационная безопасность

Дисциплина: Языки и методы программирования

КУРСОВАЯ РАБОТА

«Шифр Виженера»

Выполнил студент: Сотов Сергей Сергеевич
Группа 20-ИББо-6

Проверил: доцент кафедры "Защиты информации", кандидат технических наук Мозохин Александр Евгеньевич

Оценка _____

Подпись преподавателя _____

Кострома

2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Цель работы: Используя знания, полученные за курс изучения дисциплины «Языки и методы программирования», создать программу, реализующую шифрование алгоритмом Вижерена.

Задача: Изучить теоретический материал по теме и разработать приложение Windows Forms на языке C#, которое:

1. Шифрует введенный пользователем текст при помощи введенного им же ключа алгоритмом Виженера, зашифровывая информацию содержащуюся в тексте;
2. Дешифрует текст введенный пользователем при помощи введенного им же ключа;
3. Открывает и сохраняет ключ, зашифрованное сообщение и расшифрованное сообщение.

1. ТЕОРИЯ

1.1. Описание

Шифр Виженера (фр. Chiffre de Vigenère) — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова.

Этот метод является простой формой многоалфавитной замены. Шифр Виженера изобретался многократно. Впервые этот метод описал Джовани Баттиста Белласо (итал. Giovan Battista Bellaso) в книге La cifra del. Sig. Giovan Battista Bellaso в 1553 году, однако в XIX веке получил имя Блеза Виженера, французского дипломата. Метод прост для понимания и реализации, он является недоступным для простых методов криптоанализа.

Хотя шифр легко понять и реализовать, на протяжении трех столетий он противостоял всем попыткам его сломать; чем и заработал название le chiffre indéchiffrable (с французского 'неразгаданный шифр'). Многие люди пытались реализовать схемы шифрования, которые по сути являлись шифрами Виженера.

В шифре Цезаря каждая буква алфавита сдвигается на несколько позиций; например в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так далее. 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 (рис. 1.1). 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. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

В XIX в. большое распространение получил так называемый метод блокнотного шифрования. Им пользовались революционеры-народники, шпионы и т.п. Шифр использует фразы, взятые из языка, как ключ шифрования.

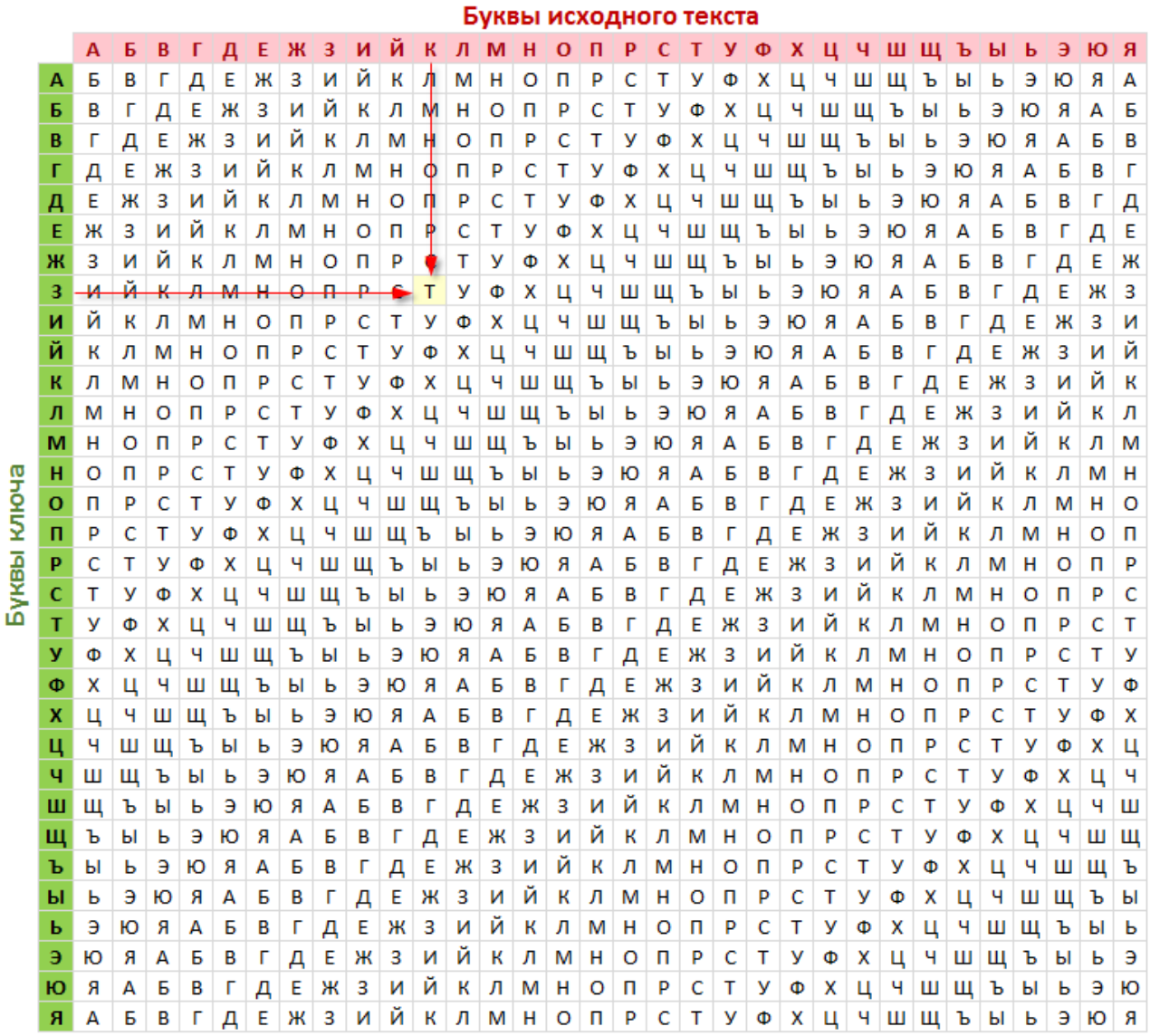

Рис. 1.1. «Квадрат» Виженера

1.2. Криптоанализ

Шифр Виженера «размывает» характеристики частот появления символов в тексте, но некоторые особенности появления символов в тексте остаются. Главный недостаток шифра Виженера состоит в том, что его ключ повторяется. Поэтому простой криптоанализ шифра может быть построен в два этапа:

1. Поиск длины ключа. Можно анализировать распределение частот в зашифрованном тексте с различным прореживанием. То есть брать текст, включающий каждую 2-ю букву зашифрованного текста, потом каждую 3-ю и т. д. Как только распределение частот букв будет сильно отличаться от равномерного (например, по энтропии), то можно говорить о найденной длине ключа.

2. Криптоанализ. Совокупность l шифров Цезаря (где l — найденная длина ключа), которые по отдельности легко взламываются.

Тесты Фридмана и Касиски могут помочь определить длину ключа.

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

Я разрабатывал свою программу в интегрированной среде разработки Microsoft Visual Studio 2019 Community. Целевая рабочая среда .NET Framework 4.6.1.

Свою работу с кодом я начал с создания и заполнения таблицы Виженера.

2.1. Таблица Виженера

Моя задача состояла в том, чтобы создать нечто похожее на то, что изображена на рисунке 1.1. Реализовал я это через создание двумерного массива. Мой алфавит состоит из строчных и заглавных букв кириллицы и латиницы, а также из цифр, таким образом наш алфавит состоит из 128 символов.

Массив заполняется следующим образом, описанным дальше. Есть два цикла, один проходил по строкам, другой по столбцам; сначала цикл проходил по нулевой строке и заполняет [0, j], где j = 0, 1, 2, 3, 4, ..., 127. Далее заполняется следующая строка: [1, j], где j = 0, 1, 2, 3, 4, ..., 127. И так далее, пока массив полностью не заполнится.

Каждая последующая строка заполняется со сдвигом на один элемент влево, а в конец дописываются символы из начала алфавита. Код представлен на рисунке 2.1, блок-схема на рисунке 2.2.

```
char[,] TableVigen = new char[ALF.Length, ALF.Length];
int s = 0;
for (int i = 0; i < ALF.Length; i++)
{
    for (int j = 0; j < ALF.Length; j++)
    {
        TableVigen[i, j] = ALF[(j + s) % ALF.Length];
    }
    s++;
}
```

Рис. 2.1 Код создания и заполнения «таблицы» Виженера

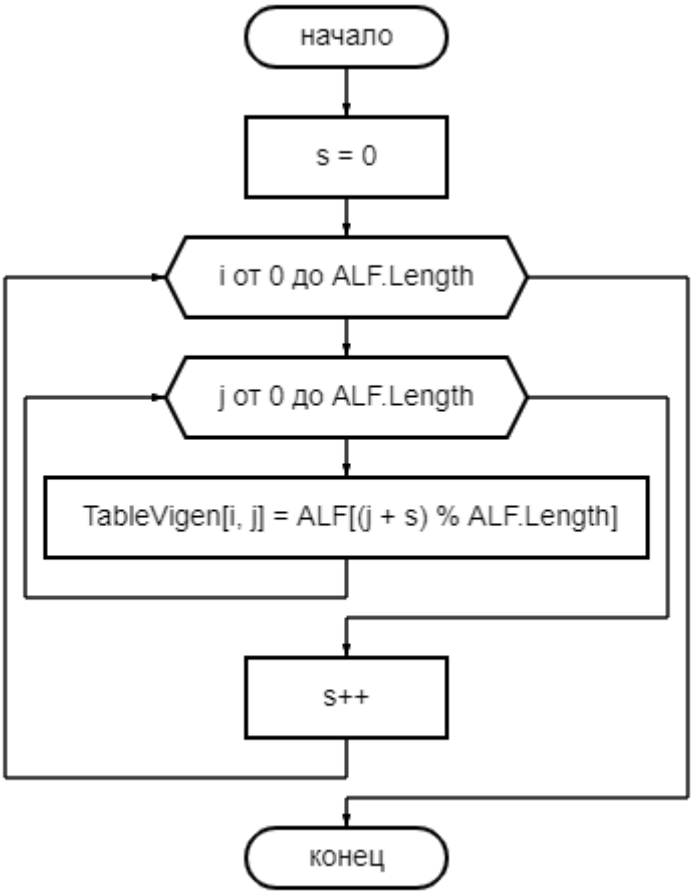

Рис. 2.2 Блок-схема заполнения «таблицы» Виженера

2.2. Шифрование

Работа в программе начинается с того, что пользователь вводит текст, который нужно зашифровать, и ключ, с помощью которого будет происходить шифрование. После того как всё введено корректно, пользователь нажимает на кнопку «Зашифровать» и начинается работа кода.

Сначала берется ключ и методом IndexOf() в алфавите находится положение каждого символа ключа. Создается массив в который будут вложены положения ключа. Если сообщение длиннее ключа, то ключ повторяется. Далее берется текст, который нужно зашифровать, и поочередно находится положение в алфавите каждого символа, существующего в алфавите, потом положение первого символа ключа и первого символа сообщения отправляются в метод с двумерным массивом, «таблицей» Виженера. В методе в этой таблице находится элемент с номером [положение символа сообщения, положение символа ключа]

(поиск по строке и столбцу). Этот символ возвращается в основной метод и записывается в выходную строку. И так поочередно каждый символ сообщения проходит этот путь. Код представлен на рисунках 2.3 и 2.4.

```
for (int i = 0; i < VH.Length + 1; i++)
{
    if (i < KL.Length)
    {
        if (ALF.IndexOf(KL[i]) >= 0)
        {
            PKL[i] = ALF.IndexOf(KL[i]);
        }
    }
    else PKL[i] = PKL[i - KL.Length];

}
int s = 0;
for (int i = 0; i < VH.Length; i++)
{

    if (ALF.IndexOf(VH[i]) >= 0)
    {
        VIH += Table(a: ALF.IndexOf(VH[i]), b: PKL[s]);
        s++;
    }
    else
    {
        VIH += VH[i];
    }
}
```

Рис. 2.3 Код поиска положения символов ключа и сообщения

```
return TableVigen[a, b];
```

Рис. 2.4 Поиск зашифрованного символа, где a – положение символа сообщения, b – положение символа ключа

2.3. Дешифрование

Процесс дешифрования очень схож с шифрованием. Отличается лишь то, что в метод с двумерным массивом отправляется символ из зашифрованной строки и положение символа ключа. В самом методе запускается цикл проверки, он проверяет на соответствие символ зашифрованного сообщения и элемент из таблицы [i, положение символа ключа], где i = 0, 1, 2, 3, …, 127. Когда эти символы совпадут, тогда метод вернет элемент i – тый символ алфавита. Код представлен на рисунке 2.5, блок-схема на рисунке 2.6.

```
for (int i = 0; i < ALF.Length; i++)
{
    if (TableVigen[i, b] == c)
    {
        return ALF[i];
    }
}
```

Рис. 2.5 Код поиска символа расшифрованного сообщения

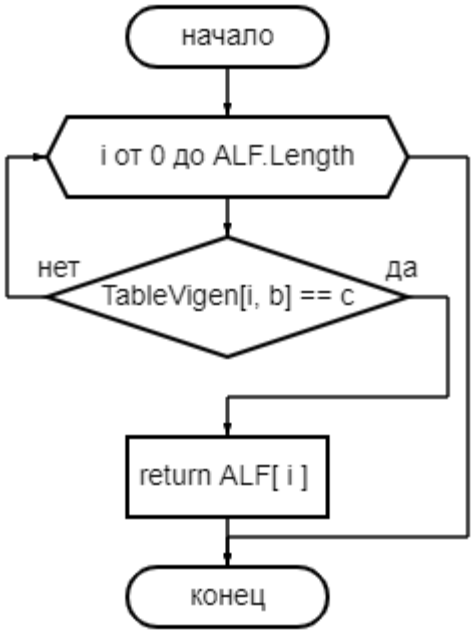

Рис. 2.6 Блок-схема поиска символа расшифрованного сообщения

3. ОПИСАТЕЛЬНАЯ ЧАСТЬ

3.1. Внешний вид

Программа выглядит как обычное приложение Windows (рис. 3.1).

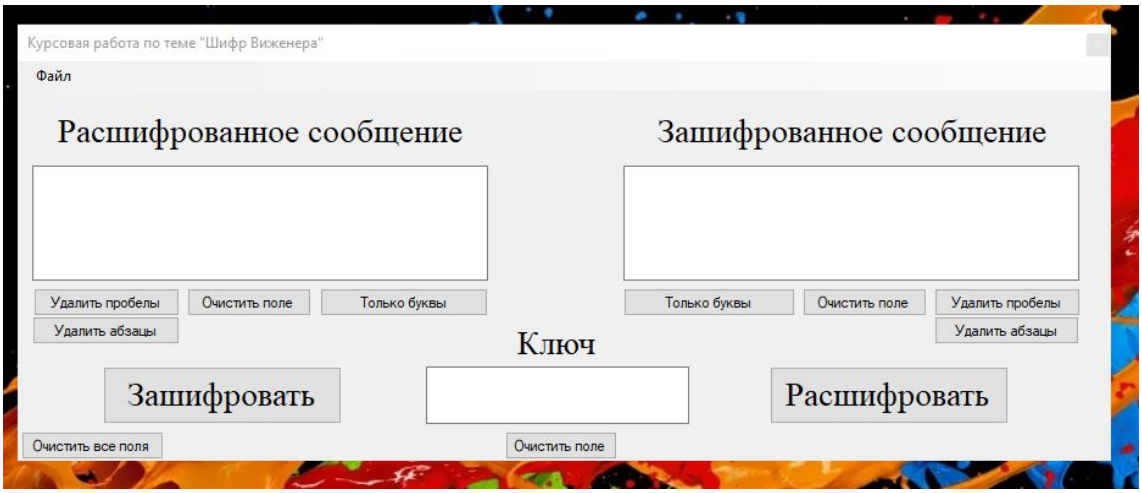

Рис. 3.1 Окно программы

3.2. Функционал

Как видно на рисунке 4.1 в окне программы есть три текстовых поля, две основные кнопки и мелкие кнопки для редактирования содержимого текстовых полей.

Две основные кнопки нужны для шифрования и дешифрования. С помощью мелких кнопок можно очистить поля, удалить пробелы или абзацы, оставить в текстовом поле только буквы и цифры, т.е. символы входящие в алфавит.

При отсутствии текста или неправильности написания появится окно с соответствующим предупреждением (рис. 3.2).

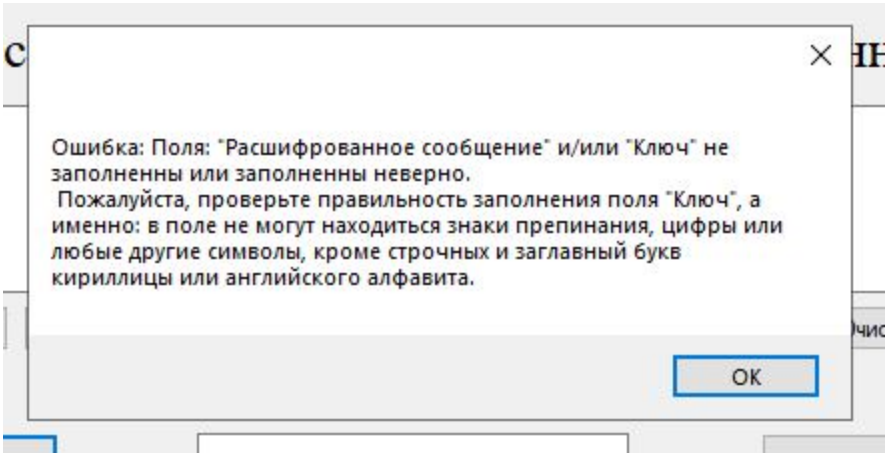

Рис. 3.2 Окно ошибки

Также есть меню сохранения и открывания текстовых файлов (рис. 3.3). Можно открыть ил сохранить содержимое каждого поля отдельно или же вместе. Сохраняет файлы она в своем собственном расширении: .vig. Но возможность открытия обычным текстовым редактором присутствует.

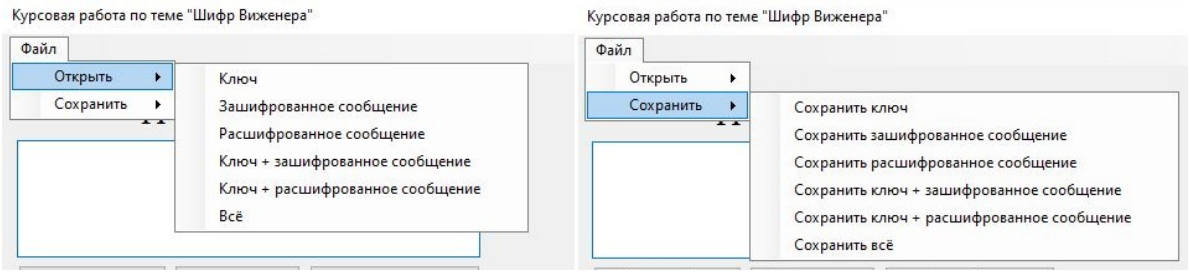

Рис. 3.3 Меню сохранения и открытия текстовых файлов

При успешном сохранении выскочит сообщение (рис. 3.4).

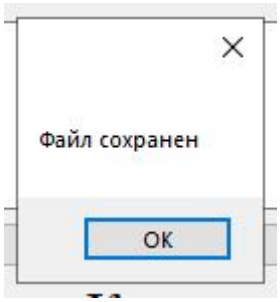

Рис. 3.4 Окно успешного сохранения файла

ВЫВОД

При выполнении этой работы я использовал знания, полученные за курс. Получилась программа, которая выполняет всё, что было в задаче, а значит курсовая работа выполнена полностью.

Я считаю, что шифр Виженера достаточно эффективный для использования его для шифрования не особо секретных сообщений. Так как в современности этот шифр могут расшифровать различными методами, например, частотным анализом. По моему мнению лучше будет комбинировать его с другими шифрами, тогда его эффективность значительно повысится.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Сетевые ресурсы:

1. Шифр Виженера — Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Шифр_Виженера/ (дата обращения: 05.04.2021).

2. Шифр Виженера - Криптография - Google Sites [Электронный ресурс] – Режим доступа: https://www.sites.google.com/site/kriptografics/sifr-vizenera/ (дата обращения: 05.04.2021).

3. Курс: Языки и методы программирования [Электронный ресурс] / А. Е. Мозохин. – Режим доступа: https://sdo.ksu.edu.ru/course/view.php?id=3013 (дата обращения: 12.04.2021).