УДК 004

Разработка программы для шифрования и дешифрования сообщения используя алгоритм сдвига символов на языке программирования C#

Звайгзне Алексей Юрьевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Научный руководитель:
Глаголев Владимир Александрович
Приамурский государственный университет имени Шолом-Алейхема
К.г.н., доцент, доцент кафедры информационных систем, математики и правовой информатики

Аннотация
В данной статье описывается процесс разработки программы для шифрования и дешифрования сообщения методом шифрования его на себя на языке программирования C#. Для создания программы используется интегрированная среда разработки Microsoft Visual 2022. В рамках статьи будет создано консольное приложение, принимающее на вход пароль пользователя для шифрования заранее подготовленного файла и повторение данного пароля для дешифрования файла.
Ключевые слова: C#, шифрование, дешифрование

Development of a program for encrypting and decrypting messages using a character shift algorithm in the C# programming language

Zvaigzne Alexey Yurievich
Sholom-Aleichem Priamursky State University
Student

Scientific supervisor:
Glagolev Vladimir Aleksandrovich
Sholom-Aleichem Priamursky State University
Ph.D, Associate Professor, Associate Professor of the Department of Information Systems, Mathematics and Legal Informatics

Abstract
This article describes the process of developing a program for encrypting and decrypting a message by encrypting it to itself in the C# programming language. The Microsoft Visual 2022 Integrated Development Environment is used to create the program. As part of the article, a console application will be created that accepts the user's password for encrypting a pre-prepared file and repeating this password to decrypt the file.

Keywords: C#, encryption, decryption

1 Введение

1.1 Актуальность

На данный момент, когда обмен информацией и данными в сети интернет стал неотъемлемой частью нашей повседневной жизни, вопросы безопасности и конфиденциальности данных стали более актуальными и насущными, чем когда-либо. Защита информации от несанкционированного доступа и её шифрование стали приоритетными задачами в области информационной безопасности.

Исследование принципов работы простых схем шифрования данных с использованием пользовательского пароля актуальна в свете необходимости защиты личных данных, финансовой информации и конфиденциальной корпоративной информации от возможных угроз.

Принципы работы алгоритмов шифрования и дешифрования данных с использованием паролей могут служить важной отправной точкой для понимания основ безопасности данных.

Понимание основ шифрования данных и безопасности важно не только для специалистов по информационной безопасности, но и для широкой аудитории, чтобы обеспечить конфиденциальность и целостность данных в цифровом мире, где угрозы для безопасности информации становятся всё более утонченными и разнообразными.

1.2 Обзор исследований

В своей работе Е.В. Ставер описал особенности разработки алгоритмов шифрования и дешифрование текстовых сообщений с использованием криптографического ключа для шифрования [1]. Р.Б. Адаев в своей статье рассмотрел вопрос шифрования, использования двух различных методов шифрования текстовых фраз: шифра Цезаря и Виженера [2]. Р.А. Лобов в своей работе провел сравнительный анализ доступных на сегодняшний день программ шифрования и дешифрования конфиденциальных данных в облачных хранилищах, выявление их недостатков [3]. В своей научной статье В.Н. Ручкин рассматривал программирование алгоритм ГОСТ 28147-89 в режиме простой замены для нейропроцессора NeuroMatrix NM6403 [4]. И.С. Аношин в своей статье на научной конференции описывает процесс реализации алгоритма RSA [5].

1.3 Цель исследования

Целью данной статьи является - создание консольного приложения, принимающего на вход пароль пользователя для шифрования заранее подготовленного файла и повторение данного пароля для дешифрования файла на языке программирования C#.

2 Материалы и методы

Для разработки консольного приложения на языке программирования C# используется интегрированная среда разработки Microsoft Visual 2022.

3 Результаты и обсуждения

Исходя из цели статьи разрабатывается следующий алгоритм взаимодействия пользователя с программой шифрования и дешифрования:

Шифрование:

1. Входные данные:
   - Определены три файла: text.dat (входной файл), coding.dat (зашифрованный файл) и decoding.dat (расшифрованный файл).
   - Пользователь вводит пароль для шифрования.
2. Шифрование файла:
   - Происходит чтение входного файла посимвольно с использованием StreamReader.
   - Введенный пользователем пароль подвергается шифрованию с использованием функции EncryptPassword.
   - Зашифрованный пароль записывается в зашифрованный файл.
   - Каждый символ из входного файла шифруется с использованием функции EncryptChar и записывается в зашифрованный файл.

Дешифрование:

3. Ввод пароля для дешифрования:
   - Пользователю предлагается ввести пароль для дешифрования.
4. Дешифрование файла:
   - Считывается зашифрованный пароль из файла.
   - Введенный пользователем пароль дешифруется с использованием функции DecryptPassword.
   - Если дешифрованный пароль не совпадает с введенным, процесс дешифрования прерывается с сообщением об ошибке.
   - Если пароли совпадают, каждый символ из зашифрованного файла дешифруется с использованием функции DecryptChar и записывается в расшифрованный файл.

Алгоритмы шифрования и дешифрования символов:

5. EncryptChar и DecryptChar:
   - Для каждого символа определен алгоритм шифрования и дешифрования, основанный на простом алгоритме сдвига в алфавите.
   - Сдвиг определяется с использованием функции GetShift, которая вычисляет сумму ASCII-кодов символов в ключе.

Листинг код программы:

```
using System;
using System.IO;
using System.Text;

class Program
{
   static void Main()
   {
      string inputFile = "text.dat";    // Входной файл
      string encryptedFile = "coding.dat"; // Зашифрованный файл
      string decryptedFile = "decoding.dat"; // Расшифрованный файл
      Console.Write("Введите пароль для шифрования: ");
      string password = Console.ReadLine();

      // Шифрование
      EncryptFile(inputFile, encryptedFile, password);
      Console.Write("Введите пароль для дешифрования: ");
      string decryptionPassword = Console.ReadLine();
      // Дешифрование
      bool decryptionSuccessful = DecryptFile(encryptedFile, decryptedFile,
decryptionPassword);
      if (decryptionSuccessful)
      {
         Console.WriteLine("OK");
         string decryptedText = File.ReadAllText(decryptedFile);
         Console.WriteLine("Расшифрованный текст: " + decryptedText);
      }
      else
      {
         Console.WriteLine("Пароли не совпадают. Файл не дешифрован");
      }
   }
   static void EncryptFile(string inputFile, string outputFile, string password)
   {
      using (StreamReader reader = new StreamReader(inputFile))
      using (StreamWriter writer = new StreamWriter(outputFile))
      {
         string encryptedPassword = EncryptPassword(password, password);
         writer.WriteLine(encryptedPassword);
         while (!reader.EndOfStream)
         {
            char c = (char)reader.Read();
            writer.Write(EncryptChar(c, password));
         }
      }
   }
   static bool DecryptFile(string inputFile, string outputFile, string password)
   {
      using (StreamReader reader = new StreamReader(inputFile))
      using (StreamWriter writer = new StreamWriter(outputFile))
      {
```

```
            string encryptedPassword = reader.ReadLine();
            string decryptedPassword = DecryptPassword(encryptedPassword, password);
            if (decryptedPassword != password)
            {
                return false; // Неверный пароль дешифрования
            }
            while (!reader.EndOfStream)
            {
                char c = (char)reader.Read();
                writer.Write(DecryptChar(c, password));
            }
        }
        return true; // Дешифрование успешно выполнено
    }
    static string EncryptPassword(string password, string key)
    {
        StringBuilder encryptedPassword = new StringBuilder();
        foreach (char c in password)
        {
            encryptedPassword.Append(EncryptChar(c, key));
        }
        return encryptedPassword.ToString();
    }
    static string DecryptPassword(string encryptedPassword, string key)
    {
        StringBuilder decryptedPassword = new StringBuilder();
        foreach (char c in encryptedPassword)
        {
            decryptedPassword.Append(DecryptChar(c, key));
        }
        return decryptedPassword.ToString();
    }
    static char EncryptChar(char c, string key)
    {
        int shift = GetShift(key);
        if (char.IsLetter(c))
        {
            char baseChar = char.IsUpper(c) ? 'A' : 'a';
            return (char)(((c - baseChar + shift) % 26) + baseChar);
        }
        else
        {
            return c;
        }
    }
    static char DecryptChar(char c, string key)
    {
        int shift = GetShift(key);
        if (char.IsLetter(c))
        {
            char baseChar = char.IsUpper(c) ? 'A' : 'a';
            return (char)(((c - baseChar - shift + 26) % 26) + baseChar);
```

```
            }
            else
            {
                return c;
            }
        }
        static int GetShift(string key)
        {
            int shift = 0;
            foreach (char c in key)
            {
                shift += (int)c;
            }
            return shift % 26;
        }
    }
```

Как и описано выше для работы программы заготавливаются файлы text.dat (рис. 1), содержащий оригинальный текст, coding.dat содержащий зашифрованное сообщение и decoding.dat для хранения дешифрованного сообщения для последующей работы программы (рис. 2).

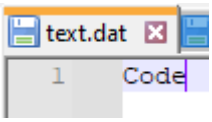

Рисунок 1. Вид заполненого фалйа text.dat

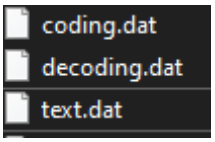

Рисунок 2. Вид созданных файлов в системе

В результате компиляции программы, выводится консольное окно, в котором пользователю предлагается ввести пароль для шифрования (рис. 3).

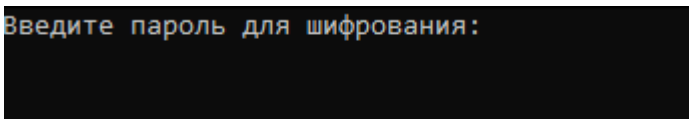

Рисунок 3. Вывод в консоль запроса на ввод пароля

После ввода пользователем пароля значения в файле шифруется и записывается в файл coding.dat (рис. 4).

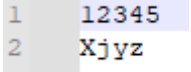

Рисунок 4. Вид зашифрованного файла coding.dat

В первой строчке указывается пароль для шифрования, а во второй зашифрованное слово.

Далее пользователю предлагается ввести пароль для дешифрования (рис. 5), если пользователь введет некорректное значение, а в консоль будет выведено следующее сообщение (рис. 6) «Пароли не совпадают. Файл не дешифрован».

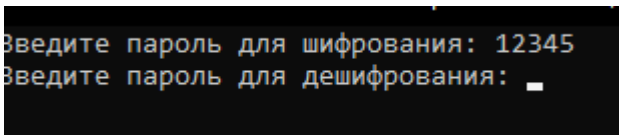

Рисунок 5. Ввод пароля для дешифрования

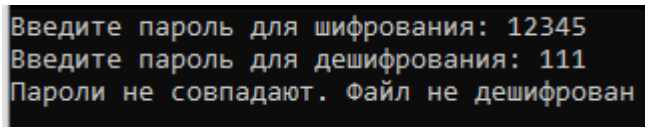

Рисунок 6. Вывод в консоль сообщения об ошибке ввода пароля

В случае корректного ввода значения зашифрованного файла будут дешифрованы и записаны в файл decoding.dat (рис. 7), а в консоль будет выведено сообщение «OK» и дешифрированное содержимое зашифрованного файла coding.dat (рис. 8).

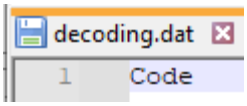

Рисунок 7. Содержимое файла decoding.dat после дешифрования

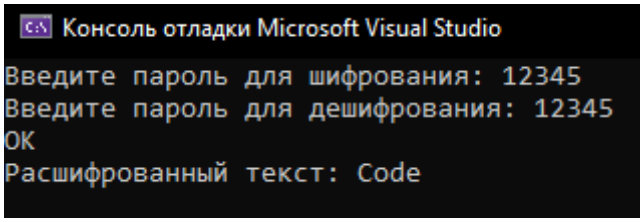

Рисунок 8. Вид консоли после успешного повтора ввода пароля для дешифрования

Выводы

Данный код демонстрирует простой метод шифрования текстовых данных с использованием пароля и алгоритма сдвига символов. Важно отметить, что такие методы не являются безопасными для применения в реальных системах безопасности данных и служат исключительно для образовательных целей. Дополнительные меры безопасности, такие как использование сильных алгоритмов шифрования, ключей и векторов инициализации, следует рассматривать при разработке систем шифрования для реальных приложений.

Библиографический список

1. Ставер Е. В. Алгоритм RSA. Шифрование и дешифрование текстовых сообщений //Научный аспект. 2012. №. 3. С. 88-89.
2. Адаев Р. Б. Программная реализация шифрования текстовых фраз //Инженерный вестник Дона. 2021. №. 11 (83). С. 172-180.
3. Лобов Р. А., Мартышкин А. И. Обзор систем шифрования и дешифрования конфиденциальных данных в облачных хранилищах //Современные информационные технологии. 2021. №. 33. С. 39-44.
4. Ручкин В. Н. и др. Шифрование и дешифрование информации на нейропроцессоре NM6403 //Информационные технологии, межвузовский сборник научных трудов. 2017. С. 123-128.
5. Аношин И. С. Шифрование и дешифрование текста на Python //Прогрессивные технологии и экономика в машиностроении: сборник трудов XIV Всероссийской научно-практической конференции для студентов и учащейся молодежи, 6-8 апреля 2023 г., Юрга. Томский политехнический университет, 2023. С. 109-111.