Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Исследование блочных шифров**

Студент: Круглик А.В.

ФИТ 3 курс 5 группа

Преподаватель:

Савельева Маргарита Геннадьевна

Минск 2023

1. **Описание приложения**

Приложение написано на языке программирования C# и позволяет:

* зашифровать текст с помощью блочного алгоритма шифрования DES;
* расшифровать текст с помощью блочного алгоритма DES.

1. **Методика выполнения поставленных задач**

Для выполнения зашифрования сообщения с помощью блочного алгоритма шифрования необходимо ввести ключ, исходный текст размешается в текстовом файле. Также в текстовых файлах размещаются зашифрованный и расшифрованный текст.

На рисунке 2.1 представлена функция для зашифрования сообщения на основании ключа и входного файла in.txt.

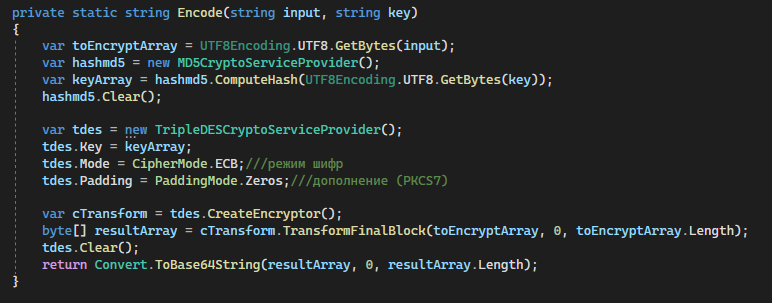


Рисунок 2.1 – Функция для зашифрования

На рисунке 2.2 представлена функция для расшифрования сообщения на основании ключа и входного файла.

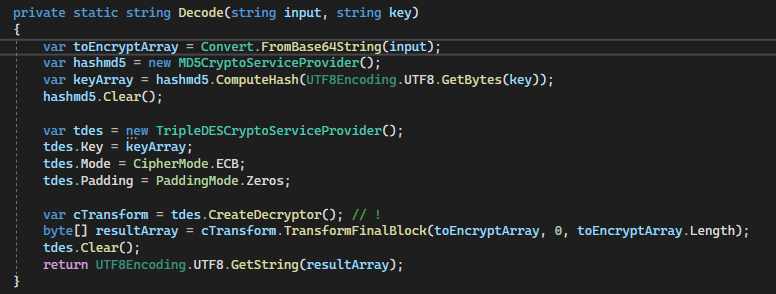


Рисунок 2.2 – Функция для расшифрования

Таким образом, были реализованы все поставленные задачи. Были исследованы блочные шифры, такие как DES. Криптостойкость такого алгоритма слабая и должна зависеть только от ключа.

Результат работы приложения представлен на рисунке 2.3.

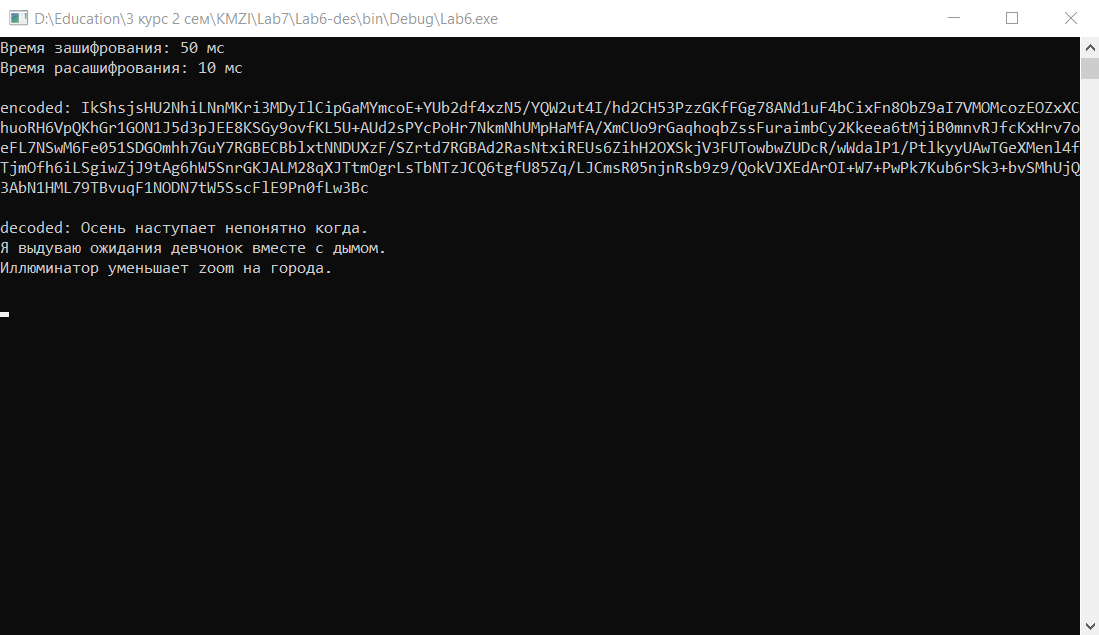


Рисунок 2.3 – Результат работы приложения

1. **Проанализировать влияние слабых ключей и полуслабых ключей на конечный результат зашифрования и на лавинный эффект**

Воспользовавшись слабым ключом для шифрования и дешифрования видно, что дешифрование выполняется намного быстрее, если бы мы использовали полуслабый или сильный ключ. Результат работы приложения c слабым ключом представлен на рисунка 2.4.

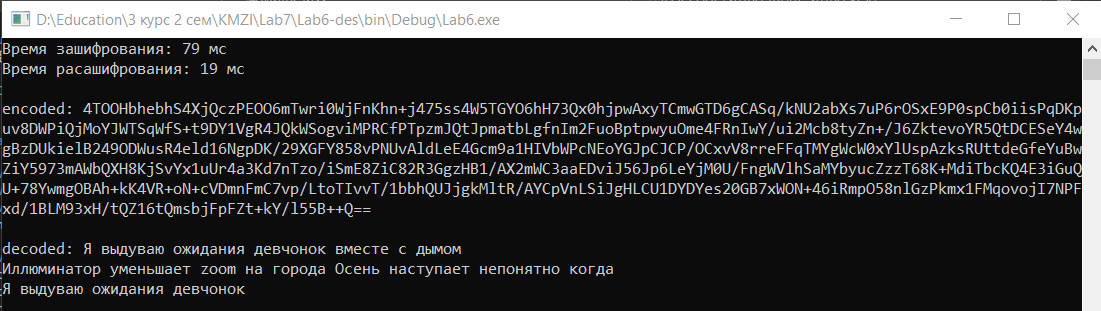


Рисунок 2.4 – Результат работы приложения со слабым ключом

Так же при работе со слабым ключом возможно расшифрование закодированного текста с помощью вообще другого слабого ключа или же вообще исходный текст и зашифрованы могут быть полностью идентичные.

Результат работы приложения c полуслабым ключом представлен на рисунка 2.5.

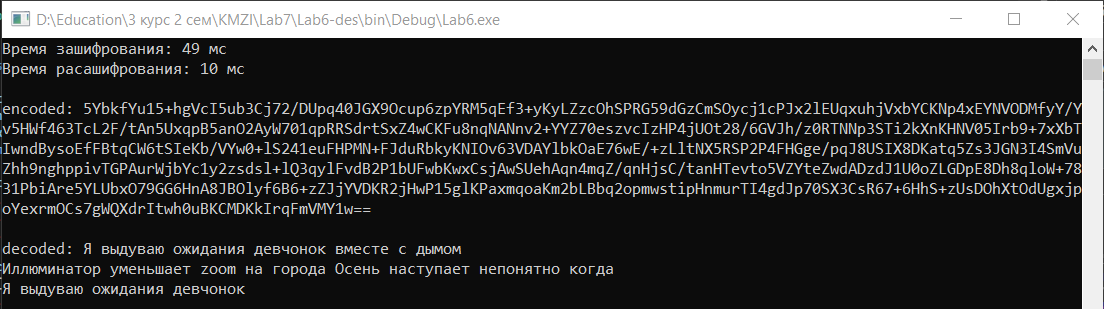


Рисунок 2.5 – Результат работы приложения со полуслабым ключом

Как видно из рисунков выше, время выполнения зашифрования, используя слабые ключи, выше, чем время выполнения зашифрования, используя полуслабые ключи. Также используя слабые ключи для зашифрования, расшифровать закодированный текст можно с помощью любого другого слабого ключа, также исходный и зашифрованный текст могут быть идентичны.

Лавинный эффект.

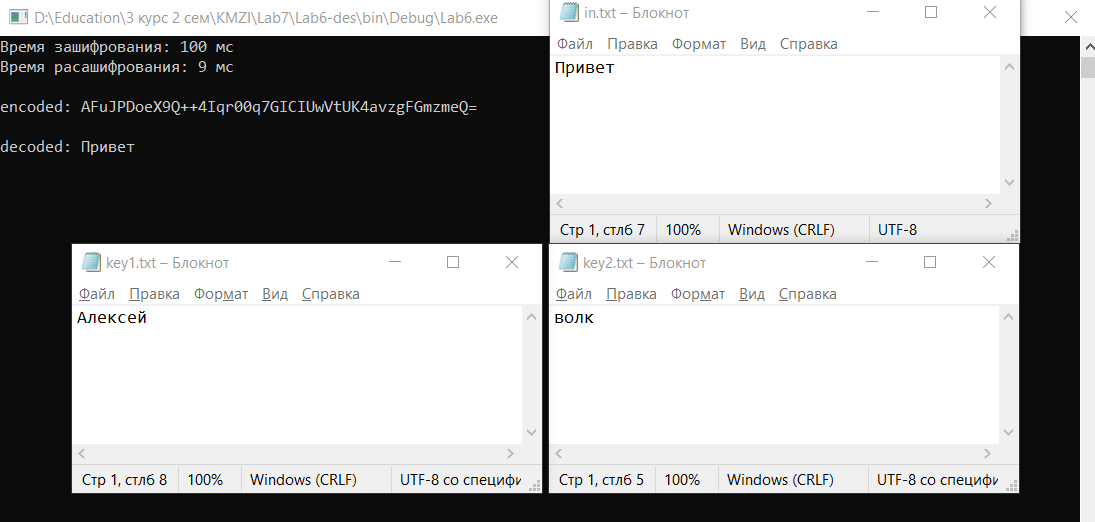


Рисунок 2.6 – Исходное сообщение

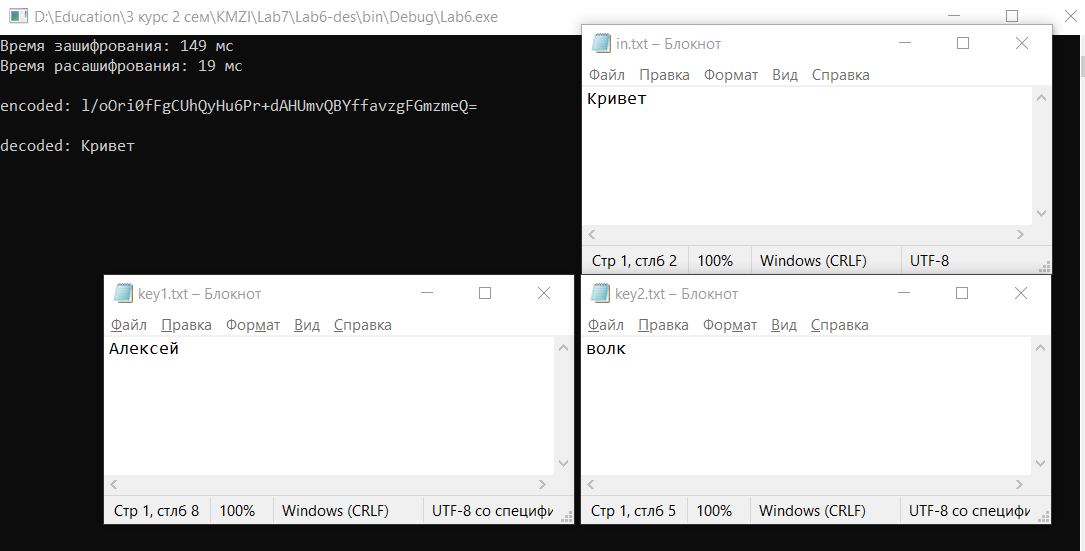


Рисунок 2.7 – Заменили один символ

После замены первой буквы в новом зашифрованном сообщении 16 символов заменились.

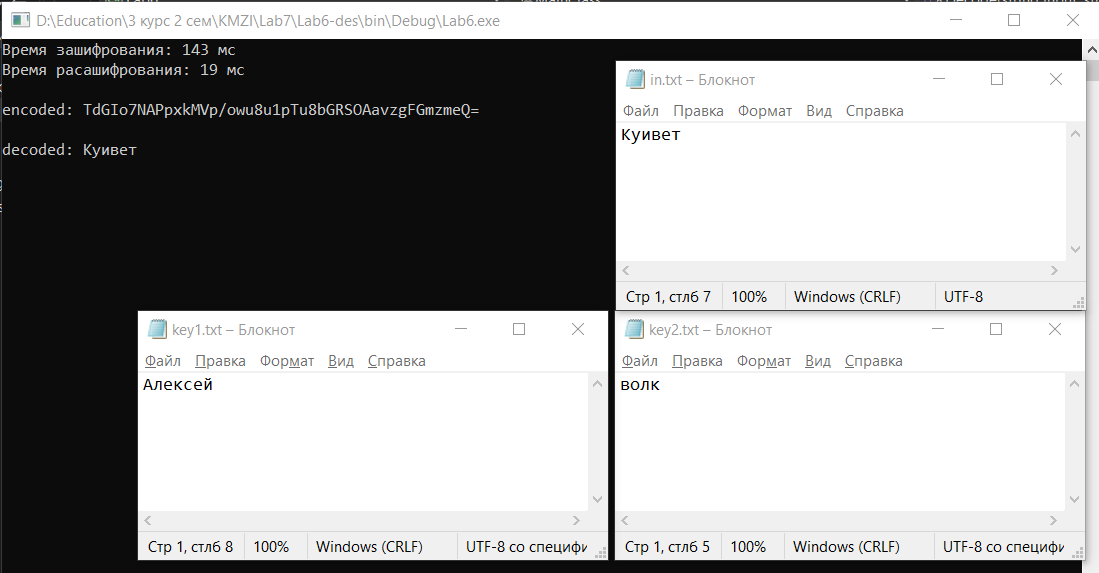


Рисунок 2.8 – Заменили 2 символа

После замены второй буквы в новом зашифрованном сообщении 25 символов заменились. А в сравнении с исходным сообщением - 24 символа.

1. **Оценить степень сжатия открытого текста и соответствующего зашифрованного текста**

Если попытаться сжать с помощью приложения WinRar два файла: с исходным текстом и с зашифрованным - то мы заметим одну закономерность: файл с зашифрованным текстом не сжимается или же практически не сжимается. Результаты сжатия представлены на рисунках 2.9, 2.10.

Если рассматривать сжатие в процентном соотношении, то in.txt сжимается на 98%, out1.txt же на 0%.

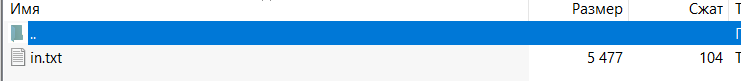
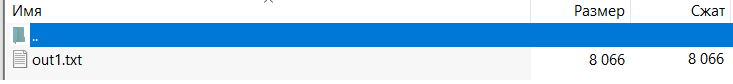
****

Рисунок 2.9 – Результат архивации исходного текста

****Рисунок 2.10 – Результат архивации зашифрованного текста

На каждом из трех этапов шифрования, при реализации алгоритма DES-EEE2, было подсчитано количество символов в шифротексте. Проанализировав его, нетрудно заметить, что на каждом шаге количество символов вырастает почти в 2 раза, что обуславливает постоянно возрастающий «лавинный эффект» и рост размера файла с шифротекстом.

**Вывод**

В ходе лабораторной работы были приобретены навыки разработки и использования блочного шифра DES. Были изучены основные принципы работы блочных шифров и оценена криптостойкость. Также было разработано приложение, на языке программирования C#, для реализации задач, связанных с шифрованием данных.