Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Исследование блочных шифров**

Студент: Чистякова Ю.А.

ФИТ 3 курс 5 группа

Преподаватель: Берников В.О.

Минск 2020

1. **Описание приложения**

Приложение написано на языке программирования C# и позволяет зашифровать и расшифровать текстовый документ с помощью алгоритма DES-EEE2. Приложение реализует следующие операции:

* разделение входного потока данных на блоки требуемой длины с дополнением последнего блока нулями;
* преобразование ключевой информации по алгоритму MD5;
* пошаговый подсчет количества символов по отношению к исходному слову.

Также приложение позволяет оценивать время выполнения операций за(рас)шифрования и записать исходной текст и соответствующий ему зашифрованный текст в выходной файл, чтобы в последующем оценить степень их сжатия.

1. **Методика выполнения поставленных задач**

Для зашифрования исходного текста были использованы встроенные экземпляры классов из пространства имен System.Security.Cryptography. Реализация функции зашифрования представлена на рисунке 2.1.

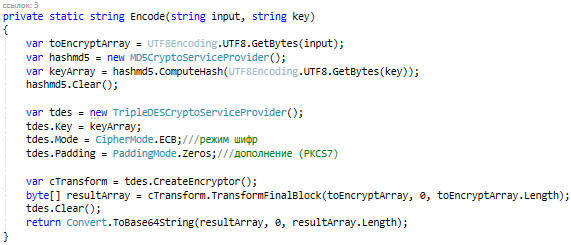


Рис. 2.1 – Реализация функции Encode

Сначала исходная строка разбивается на блоки фиксированной длины с дополнением последнего блока. Поскольку мы указали режим шифрования ECB, входной поток будет разбит на блкои длиной 8 байт (64 бит) по умолчанию. Режим дополнения Zeros указывает, что последний блок будет дополняться нулями. Также выполняется преобразование ключевой информации – переданный в параметрах ключ хешируется с помощью алгоритма MD5.

Данное программное средство должно реализовать алгоритм 3DES-EEE2: операции шифрование-расшифрование-шифрование, на первом и третьем шаге используется одинаковый ключ. Реализация данного алгоритма представлена на рисунке 2.2.

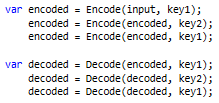


Рис. 2.2 – Реализация алгоритма DES3-EEE2

Для зашифрования и расшифования используются ключи key1, key2, считанные из файловой системы. Расшифрование происходит в обратном порядке к процедуре зашифрования, ключи также используются в обратном порядке.

Для зашифрования исходного текста использована простая маршрутная перестановка. Сначала исходная строка разбивается на подстроки длиной k (ключ), после чего последовательно считываются первый символ каждой подстроки, затем второй и т.д. по возрастанию. Реализация данного алгоритма представлена на рисунке 2.1.

Результат выполнения данного консольного приложения представлен на рисунке 2.3.

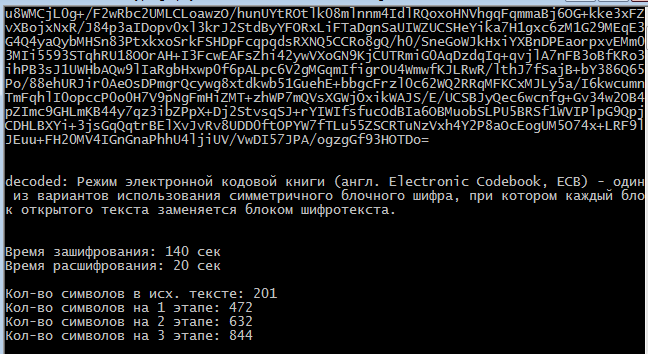


Рис. 2.3 - Результат работы приложения

Была проведена оценка скорости выполнения операций зашифрования – 140 сек, и расшифрования – 20 сек. Из этого можно сделать вывод, расшифрование происходит в разы быстрее, так как использует уже кэшированные при зашифровании данные. В целом, у алгоритма достаточно высокая скорость выполнения, которая обусловлена малой длиной ключей key1, key2.

На каждом из трех этапов шифрования, при реализации алгоритма DES-EEE2, было подсчитано количество символов в шифротексте. Результат вычислений показан на рисунке 2.3. 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.

Также, была оценена степень сжатия открытого текста (351 байт) и соответствующего зашифрованного текста (1,65 Кбайт). Такую ощутимую разницу можно объяснить тем, что с каждым вызовом зашифрования возрастает «лавинный эффект» - растет зависимость всех битов результата от битов исходных данных и ключа, а также растет количество символов в зашифрованном тексте по отношению к количеству символов исходного текста.

**Вывод**

В ходе лабораторной работы были приобретены практические навыки разработки и использования приложений для реализации блочных шифров. Также, был выполнен анализ криптостойкости блочных шифров, оценена скорость зашифрования/расшифрования и сделаны соответствующие выводы.