Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Исследование блочных шифров**

Студент: Буданова К. А.

ФИТ 3 курс 5 группа

Преподаватель:

Савельева Маргарита Геннадьевна

1. **Цель работы**

Изучение и приобретение практических навыков разработки и использования приложений для реализации блочных шифров.

1. **Задание**

Разработать авторское приложение в соответствии с целью лабораторной работы. При этом можно воспользоваться готовыми библиотеками либо программными кодами, реализующими некоторые блочные алгоритмы.

Приложение должно реализовывать следующие операции:

1. разделение входного потока данных на блоки требуемой длины с необходимым дополнением последнего блока;
2. выполнение требуемых преобразований ключевой информации;
3. выполнение операций зашифрования/расшифрования;
4. оценка скорости выполнения операций зашифрования/расшифрования;
5. пошаговый анализ лавинного эффекта с подсчетом количества изменяющихся символов по отношению к исходному слову.

Также, нужно проанализировать влияние слабых ключей и полуслабых ключей на конечный результат зашифрования и на лавинный эффект и оценить степень сжатия (используя любой доступный архиватор) открытого текста и соответствующего зашифрованного текста. Дать пояснения к полученному результату.

1. **Ход работы**

В соответствии с вариантом был выбран алгоритм DES-EDE3.

Для зашифрования сообщений была разработана функция Encrypt(). Она принимает в качестве параметра строку и ключ, с помощью которого будет осуществляться шифрование. Поскольку алгоритм DES-EDE3, то сначала мы будем шифровать сообщение, потом расшифровывать, потом обратно зашифровывать. И все эти три операции проводятся с разными ключами. Код функции для шифрования представлен на рисунке 3.1.

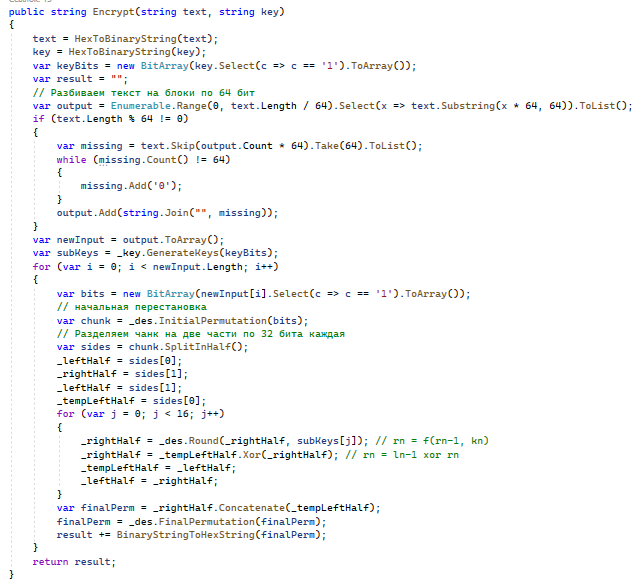


Рисунок 3.1 – Функция Encrypt(), реализующая алгоритм 3DES

Для расшифрования сообщения была разработана функция Decrypt(). Стоить заметить, что для расшифрования сообщения нужно использовать функции в обратном порядке: расшифрование – зашифрование – расшифрование. Код функции для расшифрования представлен на рисунке 3.2.

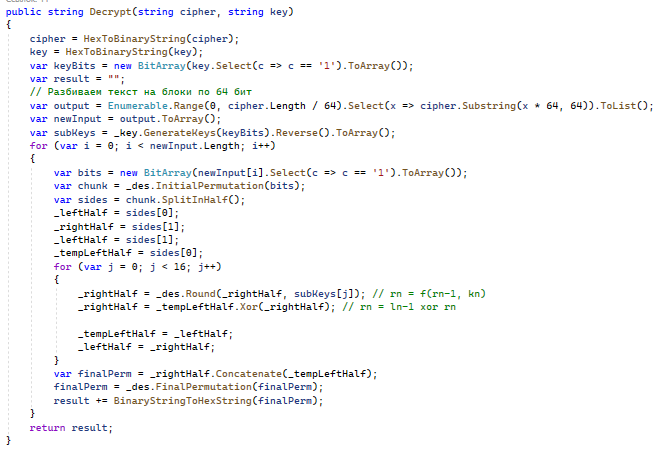


Рисунок 3.2 – Функция Decrypt(), реализующая алгоритм 3DES

Далее рассмотрим пример использования этих функций. Он представлен на рисунке 3.3.



Рисунок 3.3 – Использование функций Encrypt() и Decrypt()

Результаты выполнения функций Encrypt() и Decrypt() представлены на рисунках 3.4 и 3.5.

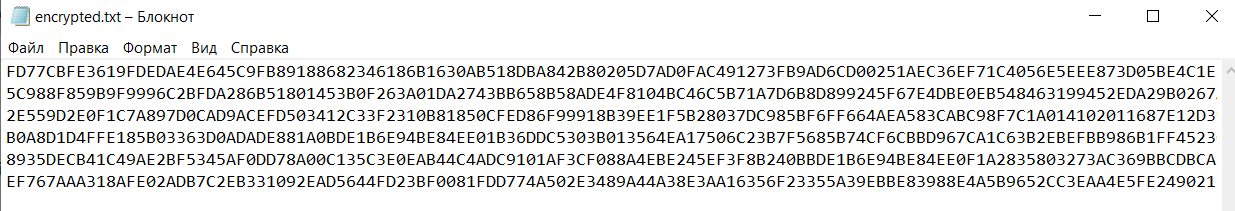


Рисунок 3.4 – Результат зашифрования сообщения

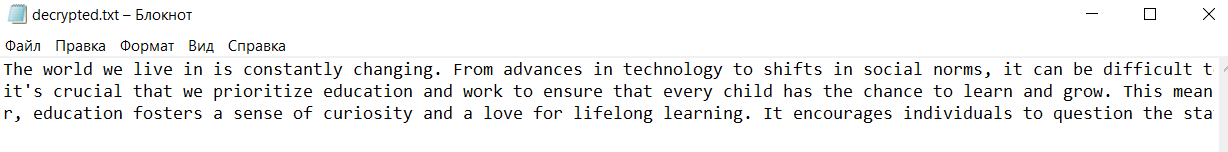


Рисунок 3.5 – Результат расшифрования сообщения

Также оценим скорость выполнения данных операций. Результаты представлены на рисунках 3.6 – 3.7.

Рисунок 3.6 – Скорость выполнения операции зашифрования

Рисунок 3.7 – Скорость выполнения операции расшифрования

Следующим шагом нужно было пошагово проанализировать лавинный эффект с подсчетом количества изменяющихся символов по отношению к исходному слову.

Лавинный эффект в криптографии означает, что небольшое изменение входного сообщения должно приводить к значительному изменению зашифрованного сообщения. Для проверки лавинного эффекта можно изменить один символ в исходном сообщении, затем зашифровать исходное сообщение и измененное сообщение и сравнить количество измененных символов в зашифрованных сообщениях.

Следующий код, представленный на рисунке 3.8 зашифрует исходное сообщение, изменит один символ в исходном сообщении и зашифрует его, затем посчитает количество измененных символов в зашифрованных сообщениях. Кроме того, он расшифрует исходное зашифрованное сообщение и выведет его на экран для проверки.



Рисунок 3.8 – Код для анализа лавинного эффекта

Далее, нужно было проанализировать влияние слабых ключей и полуслабых ключей на конечный результат зашифрования и на лавинный эффект. Ниже, на рисунке 3.9, приведён результат использования слабых и полуслабых ключей и то, как они влияют на количество изменённых символов в сравнении с исходным сообщением, а также на лавинный эффект.

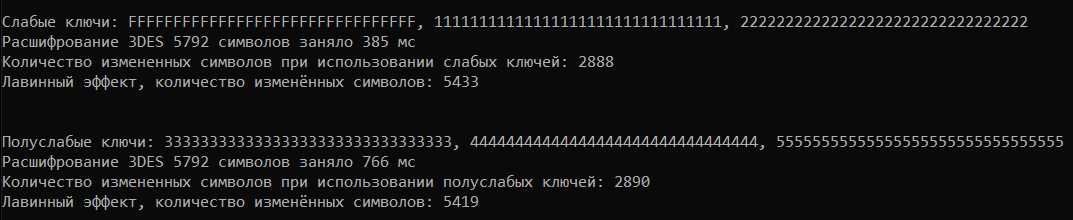


Рисунок 3.9 – Влияние слабых и полуслабых ключей на результат шифрования и на лавинный эффект

Количество измененных символов в зашифрованном сообщении, особенно в таком большом объеме, является плохим с точки зрения безопасности и целостности данных. Когда большое количество символов изменяется в процессе шифрования, это может привести к таким проблемам, как отслеживаемость, нарушение целостности данных, уязвимость к атакам. В целом, количество измененных символов в зашифрованном сообщении должно быть минимальным, чтобы обеспечить надежность и безопасность шифрования.

Кроме того, нужно было оценить степень сжатия открытого и зашифрованного текста. Для этого посмотрим на рисунки 3.10 – 3.11.

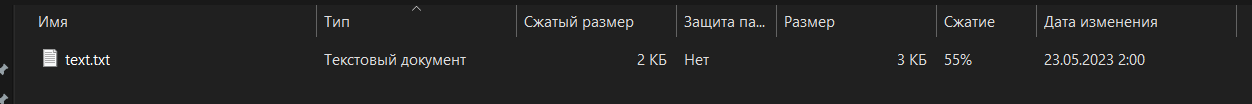


Рисунок 3.10 – Сжатие исходного текстового файла

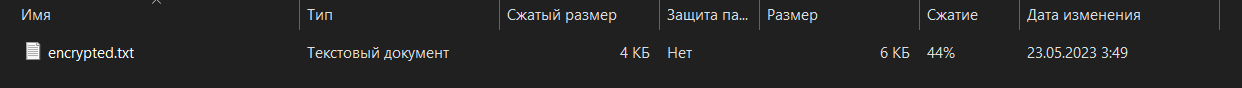


Рисунок 3.11 – Сжатие зашифрованного текстового файла

Таким образом, можно увидеть закономерность, что зашифрованные файлы меньше или же вовсе не сжимаются.

**Вывод**

В ходе лабораторной работы были изучены и приобретены практические навыки разработки и использования приложений для реализации блочных шифров.

Также было разработано авторское приложение в соответствии с целью лабораторной работы.