МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7 НА ТЕМУ:**

**ИССЛЕДОВАНИЕ БЛОЧНЫХ ШИФРОВ**

                                                                 Выполнил студент 3 курса 5 группы

Дмитрук Илья Игоревич

Минск 2024

# Задание 1.

Алгоритм DES – это симметричный блочный шифр, предназначенный для защиты данных. Он разбивает исходные данные на блоки по 64 бита и использует ключ длиной 56 бит для шифрования. Процесс начинается с начальной перестановки битов, которая перемешивает биты блока по фиксированному порядку. Далее блок проходит через 16 раундов шифрования, в каждом из которых данные разделяются на две половины, и одна из них проходит через функцию, использующую часть ключа, после чего результат комбинируется с другой половиной с помощью операции XOR. Эти операции повторяются, чередуя половины блока. В конце блок снова подвергается перестановке, обратной начальной. В результате получается зашифрованный блок данных. Декодирование выполняется аналогично, но в обратном порядке, используя тот же ключ. Так же нам необходимо было выполнить алгоритм DES-EDE2. Это означает, что нам необходимо было зашифровать сообщение сначала одним ключом, затем расшифровать другим и результат расшифровки снова зашифровать первым ключом. Данный алгоритм был реализован в функции desEde2Encrypt. Код данной функции представлен в листинге 1.1.

function desEde2Encrypt(text, key1, key2) {

    const cipher1 = crypto.createCipheriv('des-ede', Buffer.concat([key1, key2]), null);

    const cipher2 = crypto.createCipheriv('des-ede', Buffer.concat([key2, key1]), null);

    let encrypted = cipher1.update(text, 'utf8', 'hex');

    encrypted += cipher1.final('hex');

    let doubleEncrypted = cipher2.update(Buffer.from(encrypted, 'hex'), 'hex', 'hex');

    doubleEncrypted += cipher2.final('hex');

    return doubleEncrypted;

}

Листинг 1.1 – Функции шифрования алгоритмом DES

Для расшифрования использовалась функция desEde2Decrypt. Её код представлен. В листинге 1.2.

function desEde2Decrypt(encryptedText, key1, key2) {

    const decipher1 = crypto.createDecipheriv('des-ede', Buffer.concat([key2, key1]), null);

    const decipher2 = crypto.createDecipheriv('des-ede', Buffer.concat([key1, key2]), null);

    let decrypted = decipher1.update(encryptedText, 'hex', 'hex');

    decrypted += decipher1.final('hex');

    let doubleDecrypted = decipher2.update(Buffer.from(decrypted, 'hex'), 'hex', 'utf8');

    doubleDecrypted += decipher2.final('utf8');

    return doubleDecrypted;

}

Листинг 1.2 – Функции расшифрования алгоритмом DES

Результат шифрования и расшифрования, а так же количество изменённых бит, представлены на рисунке 1.1.

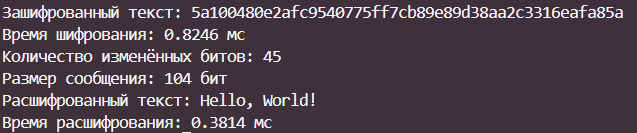


Рисунок 1.1 – Результаты зашифрования и расшифрования сообщения

Важной особенностью алгоритма шифрования DES является влияние лавинного эффекта. Его суть заключается в том, что при изменении одного бита в сообщении, приводит к массовому измению других битов при шифровании. Результат лавинного эффекта представлен на рисунке 1.2.

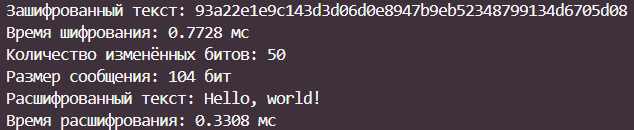


Рисунок 1.2 – Результат влияние лавинного эффекта

Так же было измерено время выполнения алгоритмов шифрования и расшифрования. Результат представлен на рисунке 1.2.

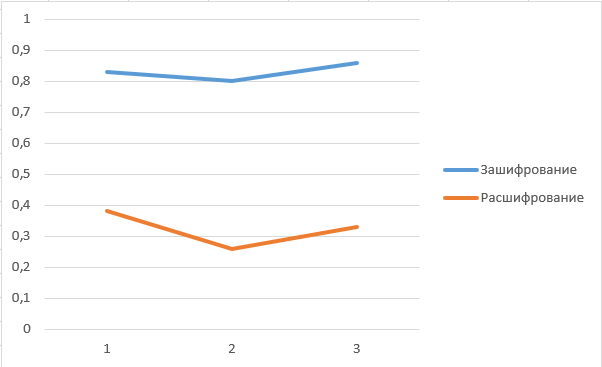


Рисунок 1.2 – Время выполнения расшифрования и зашифрования

# Задание 2.

Если использовать слабый ключ, зашифрованные данные могут быть предсказуемыми или не отличаться от исходных данных после одного или нескольких циклов шифрования и расшифрования. Лавинный эффект может быть ослаблен или даже отсутствовать, так как небольшие изменения во входных данных могут не приводить к значительным изменениям в выходных данных, что делает шифрование более предсказуемым и уязвимым к атакам. Если использовать одну из пар полуслабых ключей, зашифрованные данные, обработанные одним ключом, могут быть легко расшифрованы другим ключом из пары, что делает шифрование уязвимым. Лавинный эффект может быть неполным или искаженным, так как пара полуслабых ключей не обеспечивает сильной диффузии (распространения изменений) в данных, что ослабляет криптостойкость алгоритма. Результат шифрования полуслабым ключом представлен на рисунке 2.1, и слабым ключом на рисунке 2.2.

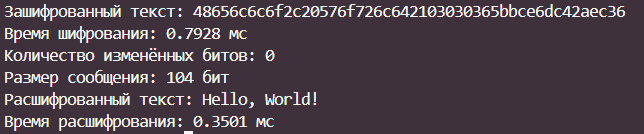


Рисунок 2.1 – Результат шифрования сообщения полуслабым ключом

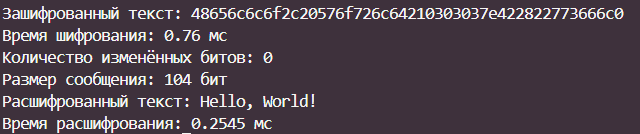


Рисунок 2.2 – Результат шифрования сообщения слабым ключом

Как мы видим, при использовании слабых и полуслабых ключей, количество изменённых битов равно нулю, что означает что часть сообщения будет составлять исходный текст.

# Задание 3.

Для выполнения данного задания использовались два txt файла, в одном был открытый текст, во втором зашифрованный. Оба файла были сжаты. Результат сжатия открытого текста представлен на рисунке 3.1 и зашифрованного текста на рисунке 3.2.

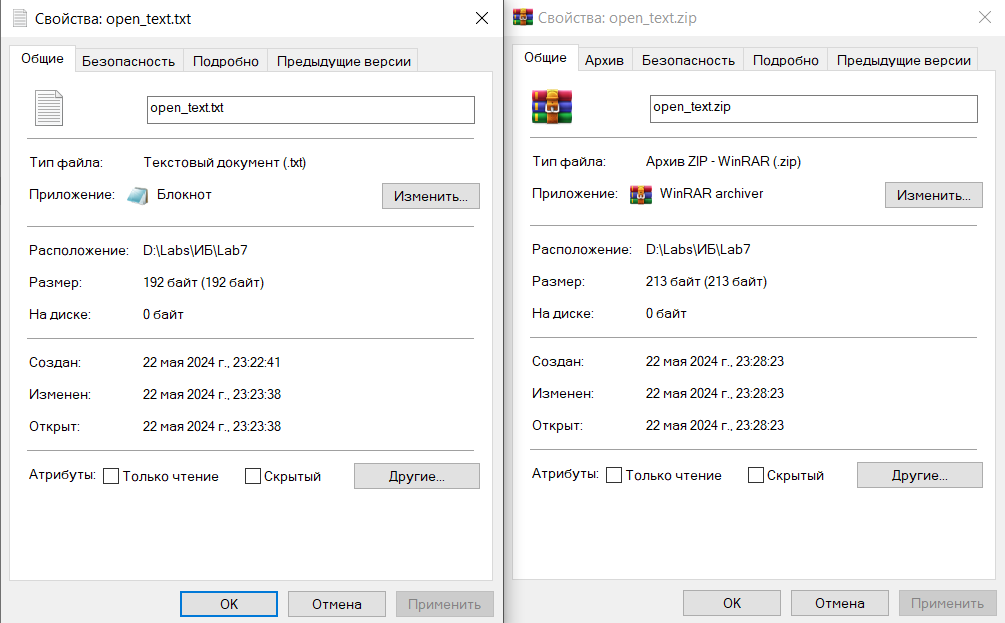


Рисунок 3.1 – Результат сжатия открытого текста

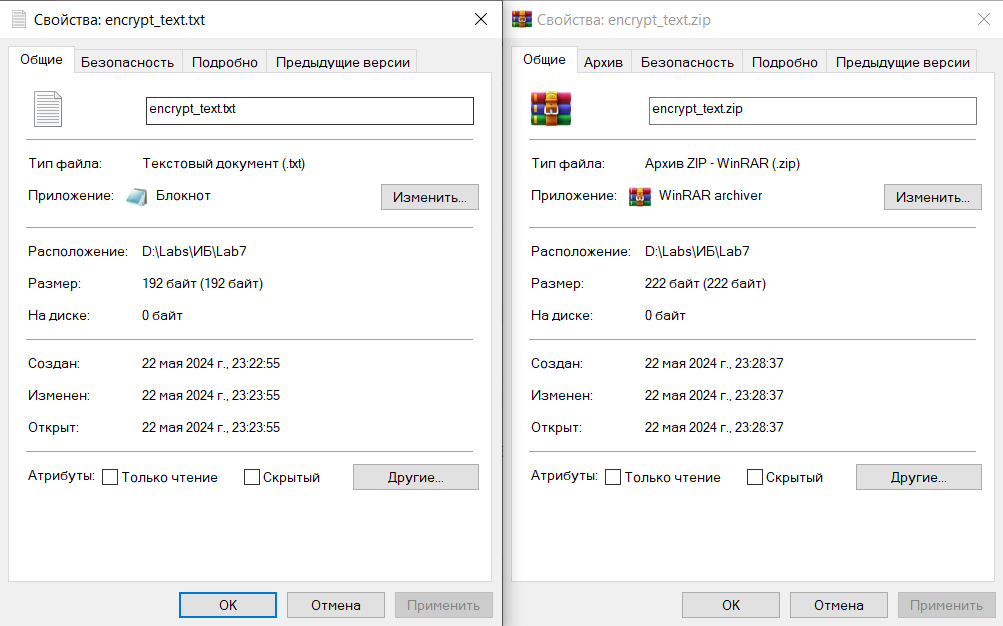


Рисунок 3.2 – Результат сжатия зашифрованного текста

Как мы видим, при сжатии зашифрованного текста, размер оказался больше. Это связано с тем, что одним из свойств алгоритма DES является то, что шифротекст практически нельзя сжать.

# Вывод

В ходе лабораторной работы были приобретены навыки шифрования алгоритмом DES-EDE2. Было разработано приложение, выполняющее данный алгоритм.