МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Информационных систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Направление специальности 1-40 01 01 10 Программирование интернет-приложений

**ОТЧЁТ К ЛАБОРАТОРНОЙ РАБОТЕ №7:**

по дисциплине «Криптографические методы защиты информации»

Исполнитель

студентка 3 курса группы 5 Шкода Кристина Михайловна

(Ф.И.О.)

Руководитель работы преподаватель Савельева М. Г.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Минск 2023

**Теоретические сведения**

Само название конструкции Фейстеля (сеть) означает ее ячеистую топологию. Формально одна ячейка сети соответствует одному раунду зашифрования или расшифрования сообщения. При зашифровании сообщение разбивается на блоки одинаковой (фиксированной) длины (как правило – 64 или 128 бит). Полученные блоки называются входными. В случае, если длина входного блока меньше, чем выбранный размер, то блок удлиняется установленным способом. Каждый входной блок шифруемого сообщения изначально делится на два подблока одинакового размера: левый *(L0*) и правый (*R0*). Далее в каждом *i*-ом раунде выполняются преобразования в соответствии с формальным представлением ячейки сети Фейстеля: *Li=Ri-1, Ri=Li-1+f (Ri-1, Ki)*.

По какому-либо математическому правилу вычисляется раундовый ключ *Ki*. В приведенном выражение знак «+» соответствует поразрядному суммированию на основе «XOR». Расшифрование происходит так же, как и зашифрование, с той лишь разницей, что раундовые ключи будут использоваться в обратном порядке по отношению к зашифрованию. В своей статье [28] Х. Фейстель описывает два блока преобразований с использованием функции *f (Ri-1, Ki):*

• блок подстановок (*S*-блок, англ. *S*-box);

• блок перестановок (*P*-блок, англ. *P*-box).

Блок подстановок состоит из:

• дешифратора, преобразующего n-разрядное двоичное число в одноразрядное сигнал по основанию 2*n*;

• внутреннего коммутатора;

• шифратора, преобразующего сигнала из одноразрядного 2*n*-ричного в *n*-разрядный двоичный.

В основе криптостойкости блочных шифров лежит идея К. Шеннона в представлении составного шифра таким образом, чтобы от обладал двумя важными свойствами: рассеянием и перемешиванием. Рассеивание должно скрыть отношения между зашифрованным текстом и исходным текстом. Рассеивание подразумевает, что каждый символ (символ или бит) в зашифрованном тексте зависит от одного или всех символов в исходном тексте.

Другими словами, если единственный символ в исходном тексте изменен, несколько или все символы в зашифрованном тексте будут также изменены. Идея относительно перемешивания заключается в том, что оно должно скрыть отношения между зашифрованным текстом и ключом.

Как было указано выше, в основе сети Фейстеля лежит простейшая операция суммирования двух (А + В) *n*-разрядных чисел – XOR:А + В (mod *n*). Помимо этой операции некоторые алгоритмы (Blowfish, IDEA, ГОСТ и др.) предусматривают выполнение операций сложения чисел по модулю более высоких порядков: XOR: А + В (mod 2*n*).

В DES входной блок данных, состоящий из 64 бит, преобразуется в выходной блок идентичной длины. В алгоритме широко используются рассеивания (подстановки) и перестановки битов текста, о которых мы упоминали выше. Комбинация двух указанных методов преобразования образует фундаментальный строительный блок DES, называемый раундом или циклом. Один блок данных подвергается преобразованию (и при зашифровании, и при расшифровании) в течение 16 раундов. После первоначальной перестановки и разделения 64-битного блока данных на правую (*R0*) и левую (*L0*) половины длиной по 32 бита выполняются 16 раундов одинаковых действий.

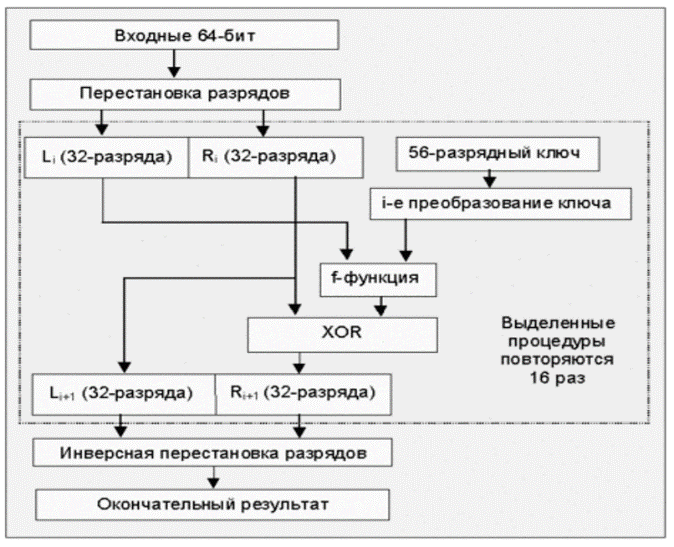


Рисунок 1 – Общая схема алгоритма DES

Первоначальные ключи являются слабыми. Вспомним, что первоначальное значение разделяется на две половины, каждая из которых сдвигается независимо. Если все биты каждой половины равны 0 или 1, то для всех раундов алгоритма используется один и тот же ключ. Это может произойти, если ключ состоит из одних 1, из одних 0, или если одна половина ключа состоит из одних 1, а другая – из одних 0.

Коммутационная панель обеспечивает самый большой набор возможных настроек. Для первого кабеля одна сторона может иметь любое из 26 положений, а другая сторона - любое из 25 оставшихся положений (одна буква коммутируются с другой). Однако, поскольку комбинация и ее обратная сторона идентичны (AB такая же, как BA), мы должны игнорировать все двойные числа во всех возможных комбинациях для одного кабеля, предоставляя (26 × 25) / (1! × 21) или 325 уникальных способов коммутаций одним кабелем. Для двух кабелей: есть (26 × 25) комбинаций – для первого кабеля и, поскольку два разъема уже используются, то получается (24 × 23) комбинаций – для второго кабеля. Следуя этой простой логике, получается  
(26 × 25 × 24 × 23) / (2! × 22) = 44 850 уникальных способов коммутаций с использованием двух кабелей. Для трех кабелей – (26 × 25 × 24 × 23 × 22 × 21)/ / (3! × 23) = 3 453 450 комбинаций и так далее. Т аким образом, с использованием 10 кабелей на коммутационной панели получаются 150 738 274 937 250 различных комбинаций. Формула, где n равно количеству кабелей, равна 26! / (26 - 2*n*)! × *n*! × 2*n*. Численно это дает: 60 × 17 576 × 676 × 150 738 274 937 250 = 107 458 687 327 250 619 360 000 или 1,07 x 1023.

Кроме того, некоторые пары ключей при зашифровании переводят открытый текст в идентичный шифртекст. Иными словами, один из ключей пары может расшифровать сообщения, зашифрованные другим ключом пары. Это происходит из-за метода, используемого DES для генерации подключей: вместо 16 различных подключей эти ключи генерируют только два различных подключа. В алгоритме каждый из этих подключей используется восемь раз. Эти ключи, называемые полуслабыми.

Существуют несколько реализаций алгоритма 3DES. Вот некоторые из них:

• DES-EEE3: шифруется 3 раза с 3 разными ключами (операции шифрование-шифрование-шифрование);

• DES-EDE3: 3DES операции шифрование-расшифрование-шифрование с разными ключами:

• DES-EEE2 и DES-EDE2: как и предыдущие, однако, на первом и третьем шаге используется одинаковый ключ.

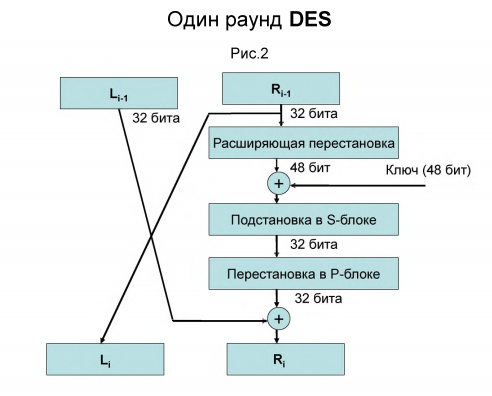


Рисунок 2 – Диаграмма, описывающая один раунд DES

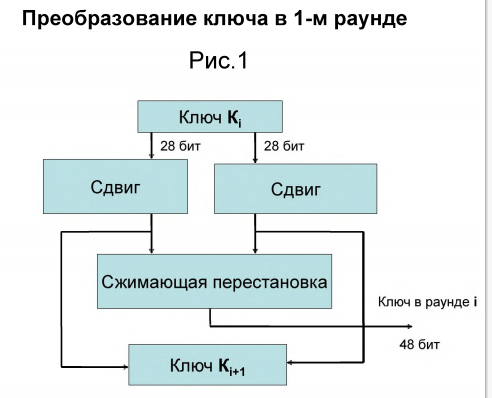


Рисунок 3 – Диаграмма, описывающая шифрование ключа

**Практическая часть**

Приложение было разработано на языке Node.JS.

На рисунке 4 представлена функция шифрования. В данной функции с помощью метода *time* и *timeEnd* засекается время выполнения шифрования. Функция encryptDES\_EDE3 принимает сообщение открытого текста и три 64-битных ключа, и возвращает шифрованный текст.

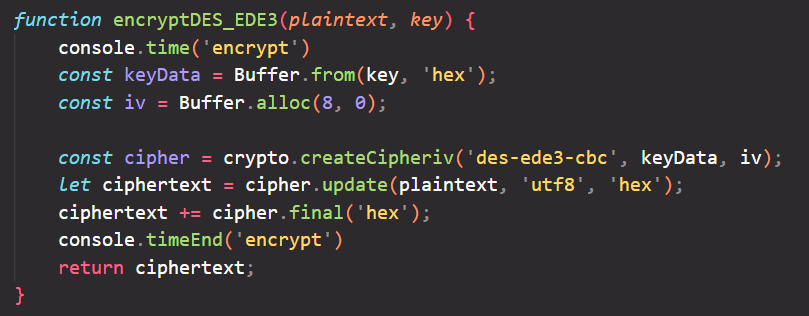


Рисунок 4 – Функция шифрования

На рисунке 5 представлена функция расшифрования. Функция decryptDES\_EDE3 принимает сообщение шифрованного текста и те же три ключа, и возвращает расшифрованный открытый текст.

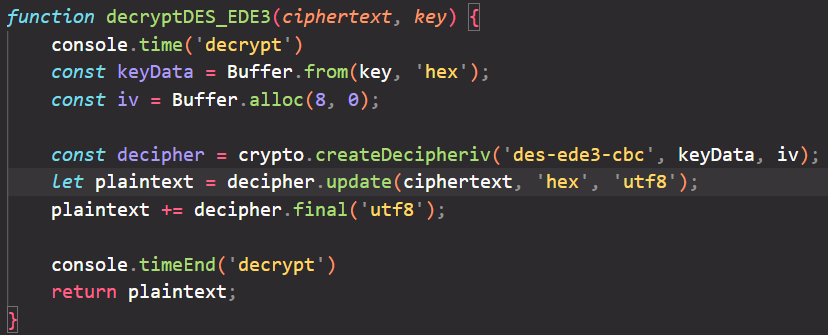


Рисунок 5 – Функция расшифрования

На рисунке 6 представлен пример вызова данных функций. Для вызова сначала надо определить открытый текст, после чего ключ, с помощью которого будет происходить шифрование.

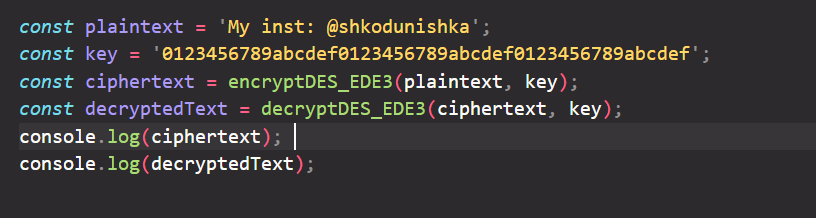


Рисунок 6 – Вызов функций

То, как описанные выше функции сработали представлено на рисунке 7.

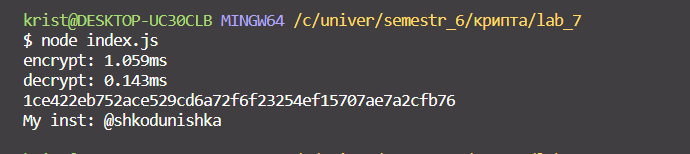


Рисунок 7 – Вызов функций

Так же на рисунке 8 представлен график зависимости времени выполнения шифрования и расшифрования от количества символов.

Рисунок 8 – График зависимости времени от количества символов

(**–** зашифрование, – расшифрование)

Лавинный эффект:

В таблице 1 показана зависимость зашифрованного текста от минимальных изменений в исходном тексте.

Таблица 1

**Лавинный эффект**

|  |  |
| --- | --- |
| Исходный текст | Зашифрованный текст |
| s | q½¿}ó3t¸ |
| sh | 1Z, |
| shk | ☺ÜêI`»Rõ |
| shko | P▬ÊnôñLÑ |
| shkod | ~«QCy▲³ |
| shkoda | s:♀´6Mä |

Влияние слабых ключей и полуслабых ключей на конечный результат зашифрования и на лавинный эффект

В данном эксперементе будем использовать в качестве слабого ключа ключ: 0000000000000000, полуслабого: E0E0E0E0F5F5F5F5

Таблица 2

**Шифрование с слабым ключом**

|  |  |
| --- | --- |
| Исходный текст | Зашифрованный текст |
| s | ³Äû§f"/ |
| sh | %bT¾aL |
| shk | á☺´2¦Ù` |
| shko | à«Ý#☻ß |
| shkod | Æ½´ÿ¢%X |
| shkoda | ¾AV$\=Ë |

Таблица 3

**Шифрование с полуслабым ключом**

|  |  |
| --- | --- |
| Исходный текст | Зашифрованный текст |
| s | ÄWÙÞè↓ |
| sh | ÜH♥Y[♠÷ç |
| shk | §²â¶¬v |
| shko | èvV^äÒT |
| shkod | R.▲7‼☼ |
| shkoda | ut♥°ìú |

В обоих случаях использование слабых и полуслабых ключей может привести к низкому уровню безопасности и возможностям атаки на систему. При использовании таких ключей изменения входных данных и ключей могут привести к малым изменениям в выходных данных или не привести к ним вовсе, что снижает уровень безопасности алгоритма и может облегчить атаку на систему.

Лавинный эффект в Triple DES означает, что небольшие изменения входных данных или ключа приводят к значительным изменениям в выходных данных. Использование слабых и полуслабых ключей может усиливать лавинный эффект и повышать вероятность возникновения случайных и непредсказуемых выходных данных при изменении входных данных или ключей. Это может привести к трудностям при обработке и анализе данных и ухудшить качество шифрования.

Сжатие файлов

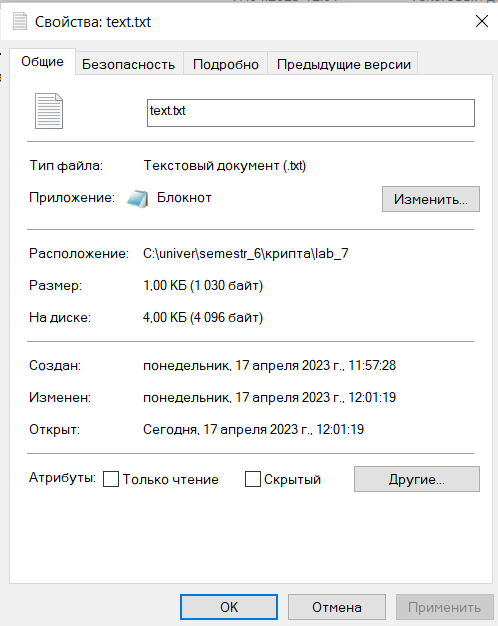
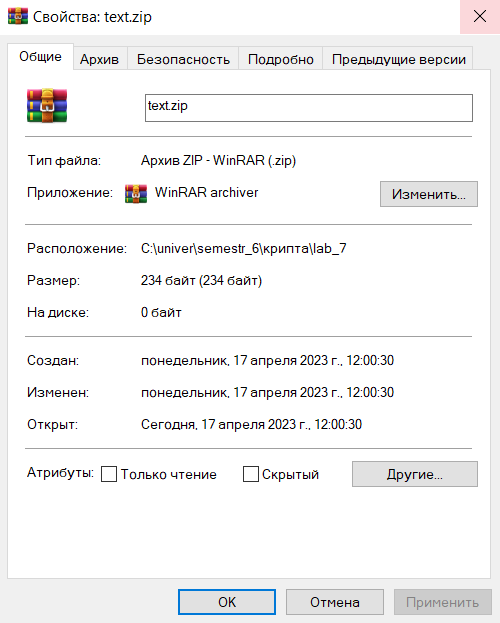
 

Рисунок 9 – Сжатие исходного файла

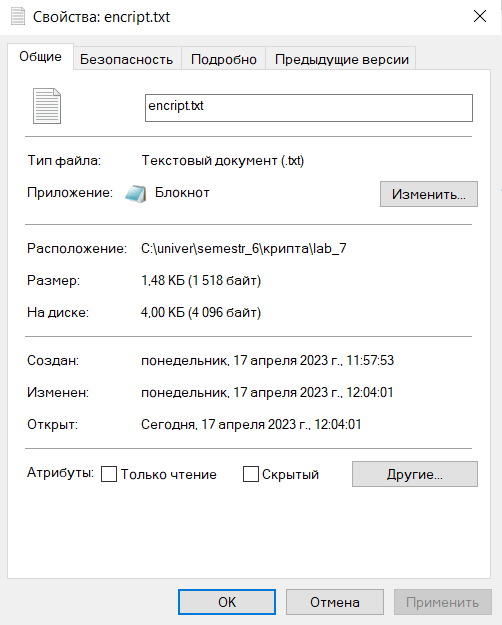
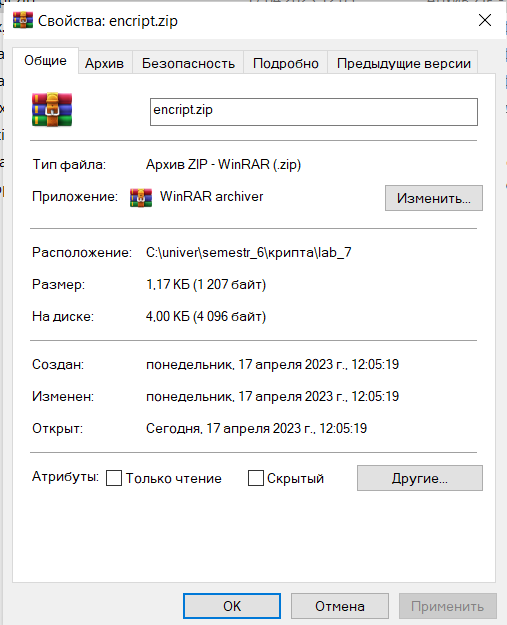
 

Рисунок 10 – Сжатие зашифрованного файла

Файл с исходным текстом получилось сжать на 78%.

Файл с зашифрованным текстом расширился на 21%.

Вывод: проделав данную лабораторную работу мы убедились в хорошей криптостойкости данного алгоритма.