Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Исследование блочных шифров

Студент: Валдайцев А. Д.

ФИТ 3 курс 5 группа

Преподаватель: Савельева М. Г.

Минск 2023

# Алгоритм DES

Алгоритм строится на основе сети Фейстеля. Входной блок данных, состоящий из 64 битов, 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 и левую половины длиной по 32 бита выполняются 16 раундов одинаковых действий.

Вначале правая часть блока *Ri* расширяется до 48 битов с использованием таблицы, которая определяет перестановку плюс расширение на 16 битов. Эта операция приводит размер правой половины в соответствие с размером ключа для выполнения операции XOR.

После выполнения перестановки с расширением для полученного 48-битного значения выполняется операция XOR с 48-битным подключом *Ki*. Затем полученное 48-битное значение подается на вход блока подстановки *S*, результатом которой является 32-битное значение. Подстановка выполняется в восьми блоках подстановки или восьми *S*-блоках.

При выполнении этой операции 48 битов данных делятся на восемь 6-битных подблоков, каждый из которых по соответствующей таблице замен замещается четырьмя битами. Подстановка с помощью *S*-блоков является одним из важнейших этапов DES. Таблицы замен для этой операции специально спроектированы так, чтобы обеспечивать максимальную криптостойкость. В результате выполнения этого этапа получаются восемь 4-битных блоков, которые вновь объединяются в единое 32-битное значение.

Далее полученное 32-битное значение обрабатывается с помощью перестановки *Р*, которая не зависит от используемого ключа. Целью перестановки является такое максимальное переупорядочивание битов, чтобы в следующем раунде шифрования каждый бит с большой вероятностью обрабатывался другим *S*-блоком.

И наконец, результат перестановки объединяется с помощью операции XOR с левой половиной первоначального 64-битного блока данных. Затем левая и правая половины меняются местами, и начинается следующий раунд.

# Зашифрование

Для начала, нам необходимо убедиться, что ключ, подаваемый на вход функции зашифрования, будет иметь размерность 64 бита. Для этого реализована функция, представленная на рисунке 1.1.

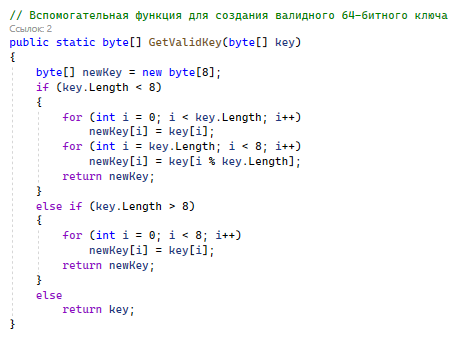


Рисунок 1.1 – Функция для создания 64-битного ключа

Для зашифрования необходимо убедиться, что блоки исходного текста будут иметь размерность 64 бита и при необходимости дополняться до данного размера. Для реализации этого используется значение PKCS7 в перечислении PaddingMode. Блоки будут дополняться байтами, равными длине дополнения. Например, если размер блока 8 байт, то для дополнения блока длиной 6 байт будет добавлено 2 байта со значением 0x02. Функция зашифрования представлена на рисунке 1.2.

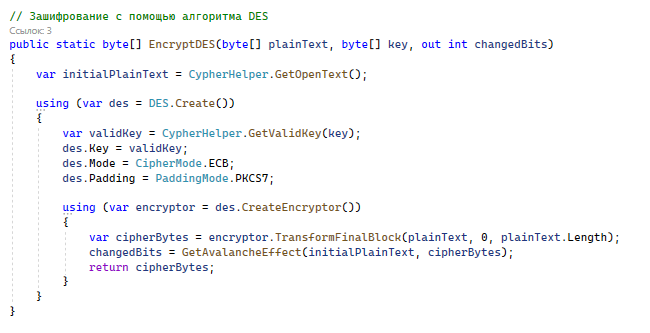


Рисунок 1.2 – Функция зашифрования алгоритмом DES

# Расшифрование

Для расшифрования необходимо повторить тот же самый алгоритм с обратным порядком ключей. Если использовались раундовые ключи с 1-го по 16-ый, то при расшифровании они будут применять в обратном порядке – с 16-го по 1-ый. Функция расшифрования представлена на рисунке 1.3.



Рисунок 1.3 – Функция расшифрования алгоритмом DES

# Лавинный эффект

Алгоритму DES присуще такое свойство, как «лавинный» эффект. Оно заключается в том, что изменение значения малого количества битов во входном тексте или в ключе ведет к «лавинному» изменению значений выходных битов шифротекста.

Для оценки «лавинного» эффекта можно побитово сравнить исходный текст и шифротекст и посчитать количество изменённых битов. Функция, высчитывающая «лавинный» эффект, представлена на рисунке 1.4.

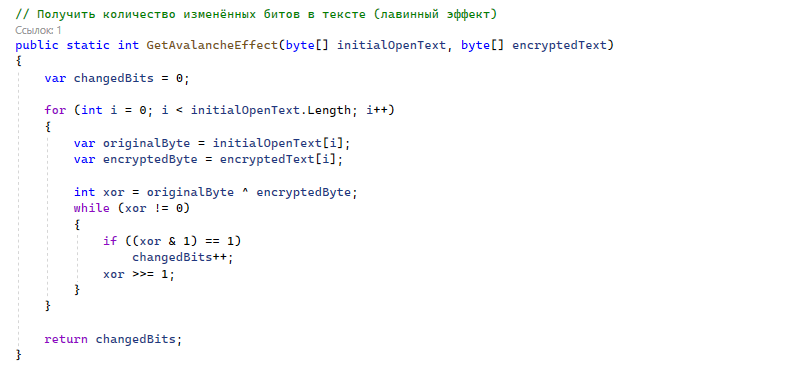


Рисунок 1.4 – Функция оценки «лавинного» эффекта

# Алгоритм DES-EEE2

Существует модификация DES, называемая 3DES или TripleDES. Смысл модификации в том, что алгоритм DES применяется три раза. Есть несколько разновидностей TripleDES:

* DES-EEE3: шифруется три раза с тремя разными ключами (операции шифрование-шифрование-шифрование);
* DES-EDE3: 3DES операции шифрование-расшифрование шифрование с тремя разными ключами;
* DES-EEE2 и DES-EDE2: как и предыдущие, однако на первом

и третьем шаге используется одинаковый ключ.

Рассмотрим подробнее алгоритм DES-EEE2.

Для реализации данного алгоритма используется функция, представленная на рисунке 1.5.

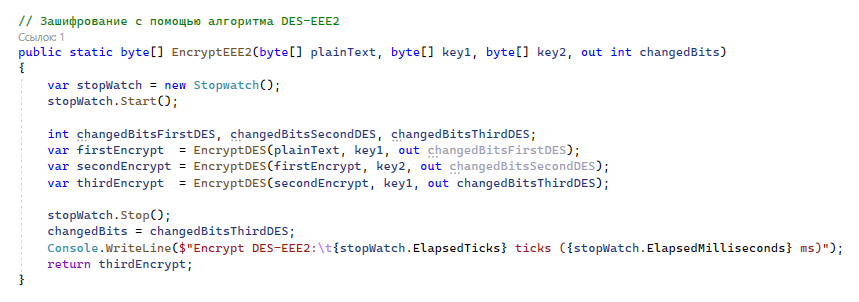


Рисунок 1.5 – Функция зашифрования алгоритмом DES-EEE2

Для расшифрования необходимо выполнить три операции расшифрования с одинаковыми ключами при первом и третьем расшифровании. Функция представлена на рисунке 1.6.



Рисунок 1.6 – Функция расшифрования алгоритмом DES-EEE2

# Оценка лавинного эффекта

Для оценки лавинного эффекта зашифруем алгоритмом DES с использованием ключа №1, изменим один символ в исходном тексте, и сравним количество изменённых символов.

Зашифрование изначального текста с изначальным ключом представлено на рисунке 3.1.

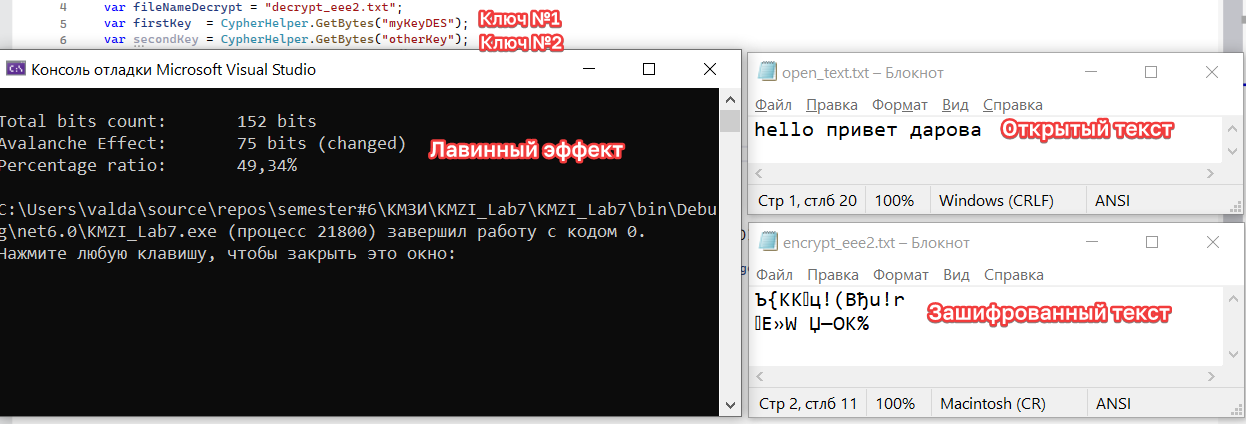


Рисунок 3.1 – Лавинный эффект, неизменённый открытый текст

Как видно из рисунка, при одном зашифровании алгоритмом DES лавинный эффект составляет 49% – то есть в шифротексте, в сравнении с открытым текстом, изменена примерно половина бит.

Далее изменим один символ открытого текста и оценим, насколько изменился шифротекст. Данные действия представлены на рисунке 3.2.

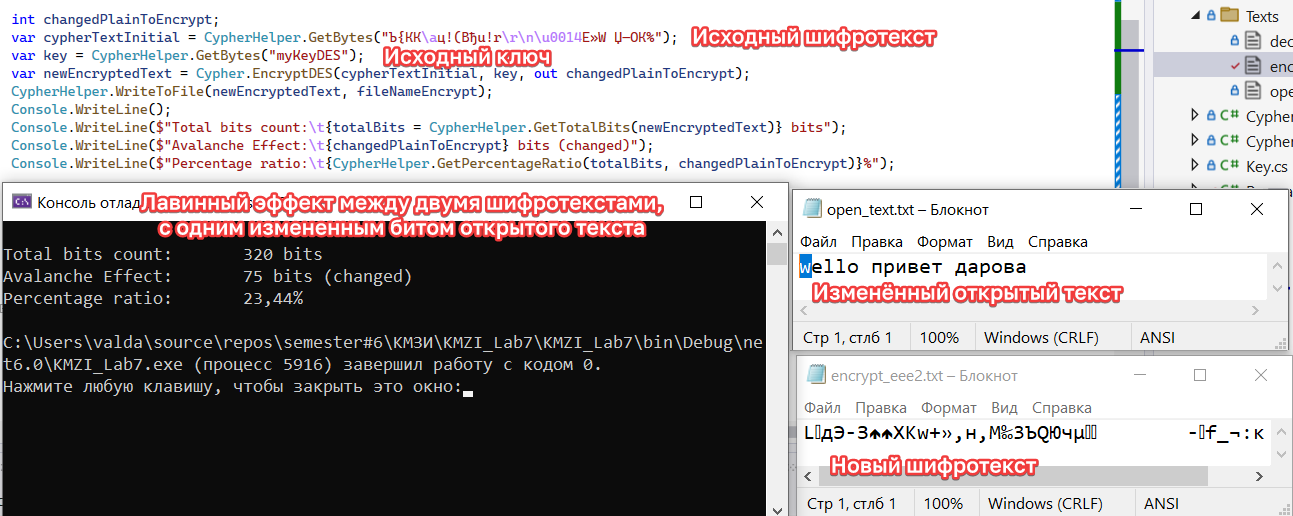


Рисунок 3.2 – Лавинный эффект, текст изменён на один символ

То есть, при изменении всего одного символа открытого текста, в шифротексте изменяется 75 битов, то есть примерно 23% всех битов текста.

# Слабые и полуслабые ключи

# Время выполнения зашифрования и расшифрования

Просто графики

# Оценка степени сжатия шифротекста

+ пояснения почему так нахуй

# Вывод