Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Отчет к лабораторной работе**:

Изучение устройства и функциональных особенностей шифровальной машины «Исследование блочных шифров»

Выполнила:

студентка 3 курса 4 группы

специальности ПОИТ

Бородина Е. Д.

Минск 2020

1. **Теоретические сведения**

Блочный шифр — разновидность симметричного шифра, оперирующего группами бит фиксированной длины — блоками, характерный размер которых меняется в пределах 64 — 256 бит. То есть блочное зашифрование (расшифрование) предполагает разбиение исходного открытого (зашифрованного) текста на равные блоки, к которым применяется однотипная процедура зашифрования (расшифрования). Указанная однотипность характеризуется, прежде всего, тем, что процедура зашифрования (расшифрования) состоит из совокупности повторяющихся наборов преобразований, называемых раундами.

Основные требования к шифрам рассматриваемого класса можно сформулировать следующим образом:

* даже незначительное изменение исходного сообщения должно приводить к существенному изменению зашифрованного сообщения;
* устойчивость к атакам по выбранному тексту;
* алгоритмы зашифрования/расшифрования должны быть реализуемыми на различных платформах;
* алгоритмы должны базироваться на простых операциях;
* алгоритмы должны быть простыми для написания кода, вероятность появления программных ошибок должна быть низкой;
* алгоритмы должны допускать их модификацию при переходе на иные требования по уровню криптостойкости.

Современная модель блочных шифров основана на идее итеративных блочных шифров, предложенной в публикации 1949 года Клода Шеннона «Теория связи в секретных системах». Данная концепция позволяет достичь определенного уровня безопасности комбинированием простых в исполнении операций подстановки и замены.

До 70-х годов криптография была уделом военных и практически не существовало каких-либо публикаций. Первопроходцем явился шифр «Люцифер», разработанный в 1970 году компанией IBM и основанный на SP-сети. Идея шифра заключалась в использовании комбинаций простых, а следовательно, быстро вычисляемых как аппаратно, так и программно операций. Однако, схема получилась неудачной: она была слишком громоздкой, что привело к низкой скорости шифрования в программной реализации (около 8 кбайт/с) и в аппаратной (97 кбайт/с). Стали появляться опасения, связанные со стойкостью данного алгоритма. Тем не менее, принципы, выработанные при построении «Люцифера», SP-сеть и сеть Фейстеля, названная так в честь одного из разработчиков, легли в основу конструирования блочных шифров.

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:



По какому-либо математическому правилу вычисляется раундовый ключ Ki. В приведенном выражение знак «+» соответствует поразрядному суммированию на основе «XOR». На рис. 1.1 приведено графическое отображение сети Фейстеля. Расшифрование происходит так же, как и зашифрование, с той лишь разницей, что раундовые ключи будут использоваться в обратном порядке по отношению к зашифрованию.

Как было указано выше, в основе сети Фейстеля лежит простейшая операция суммирования 2-х (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 2n). Понятно, что числа А и В также являются n-разрядных.

Реализация второй из указанных операций является более сложной. Вспомним основные ее особенности.

Во-первых. Самое большое слагаемое меньше 2n. Например, при n=3 самое большое слагаемое в двоичном виде – это 111 (или 7), а 2n = 8.

Во-вторых. Результатом сложения также должно быть n-разрядное число.

В-третьих. Побитовое сложение предусматривает известную взаимосвязь между соседними символами (порядками).

В-четвёртых. В силу известных правил модулярной1 арифметики результат вычисления А + В (mod 2n) – это остаток от деления: (А + В)/ 2n.

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.

Далее рассмотрим алгоритм, основанный на сети Фейстеля, - алгоритме DES (DATA ENCRYPTION STANDARD).

Стандарт шифрования данных DES – блочный шифр с симметричными ключами, разработан Национальным Институтом Стандартов и Технологии (NIST – National Institute of Standards and Technology).

В 1973 году NIST издал запрос для разработки предложения национальной криптографической системы с симметричными ключами Предложенная IBM модификация проекта, названная LUCIFER, была принята как DES. В марте 1975 криптоалгоритм DES был издан в эскизном виде в Федеральном Регистре года как Федеральный Стандарт Обработки Информации (FIPS – Federal Information Processing Standard).

После публикации алгоритм жестко критиковался по двум причинам. Первая: критиковалась сомнительно маленькая длина ключа 56 битов, что могло сделать шифр уязвимым к атаке "грубой силой". Вторая причина: критики были обеспокоены некоторым скрытым построением внутренней структуры DES. Они подозревали, что S-боксы имеет скрытую лазейку, которая позволит Национальному агенству по безопасности США расшифровывать сообщения без ключа. Впоследствии проектировщики IBM сообщили, что внутренняя структура была доработана, чтобы предотвратить криптоанализ. Федеральный Регистр объявил DES стандартом шифрования и он быстро стал наиболее широко используемым блочным шифром. Позже NIST предложил новый стандарт, рекомендующий использовать трехкратно повторенный шифр DES. В 2000 г. новый стандарт AES заменил DES.

Для шифрования DES принимает 64-битовый открытый текст и порождает 64-битовый зашифрованный текст и наоборот, получив 64 бита зашифрованного текста, он выдает 64 бита расшифрованного. В обоих случаях для шифрования и дешифрования применяется один и тот же 56-битовый ключ.

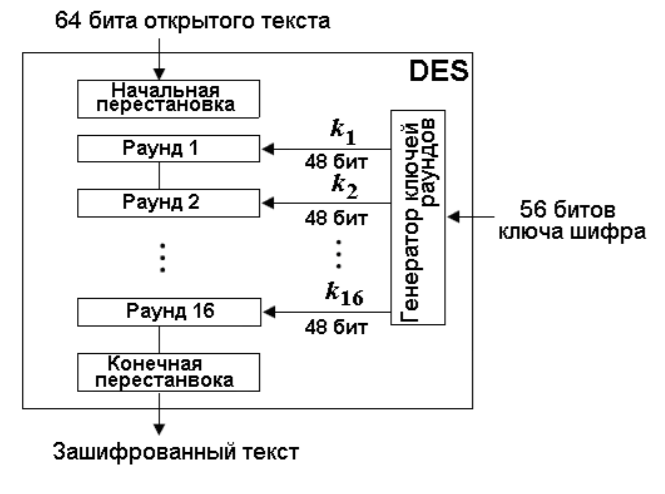


Рисунок 1.1 – Структура DES

Процесс шифрования состоит из двух перестановок, которые называют начальной и финальной (конечной) перестановками, и 16 раундов Фейстеля. Каждый раунд использует различные сгенерированные 48-битовые ключи.

На вход каждой из них поступает 64 бита, которые затем переставляются в соответствии с заданными таблицами. Эти перестановки взаимно обратны. Другими словами, 58-й бит на входе начальной перестановке переходит в 1-ую позицию на выходе из нее. А финальная перестановка 1-ый входной бит переведет в 58-ую позицию на выходе.



Рисунок 1.2 – Начальная и конечная перестановки

DES использует 16 раундов. Каждый раунд DES применяет шифр Фейстеля, как это показано на рисунке 1.3.

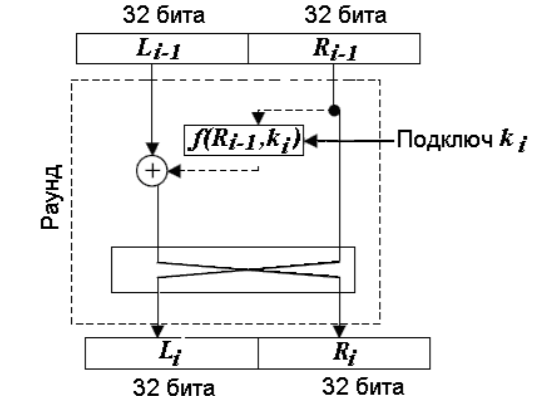


Рисунок 1.3 – Начальная и конечная перестановки

Раунд принимает полублоки Li−1 и Ri−1 от предыдущего раунда (или начального блока перестановки) и создает полублоки Li и Ri для входа в следующий раунд (или конечный блок перестановки). Все необратимые элементы сосредоточены в функции ʄ=(Ri-1,ki).

Функция DES с помощью 48-битового ключа зашифровывает 32 самых правых бит Ri−1 , чтобы получить на выходе 32-битовый результат. Эта функция содержит, как это показано на 4 составляющие: операция XOR, P-бокс расширения, группу S -боксов и прямой бокс. Структура функции представлена на рисунке 1.4.

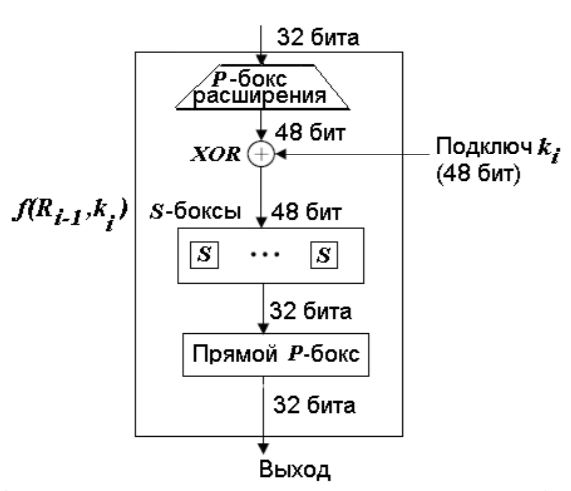


Рисунок 1.4 – Функция DES

Лавинный эффект – проявление зависимости всех выходных битов шифротекста от каждого входного бита открытого текста (в криптографии такой анализ проводят для блочных шифров и хэшфункций). Лавинный эффект проявляется в зависимости всех выходных битов от каждого входного бита. Термин введен Фейстелем, хотя концептуальное понятие использовалось еще Шенноном. Если криптографический алгоритм не обладает лавинным эффектом в достаточной степени, противник может сделать предположение о входной информации, основываясь на выходной информации. Таким образом, достижение лавинного эффекта является важной целью при разработке криптографического алгоритма.

В DES лавинный эффект проявляется уже на 4-5 раунде. Так, если зашифровать на одном ключе с помощью DES, 2 блока открытого текста, отличающиеся одним битом, то блоки шифротекстов будут отличаться на 29 бит, т.е. изменение открытого текста на 1,5% вызывает 45% изменений шифротекста.

Доказано, что после того как текст зашифрован в 8 раундах, каждый бит шифрованного текста — функция каждого бита открытого текста и ключа. Но эксперименты показывают, что DES с менее чем 16 раундами более уязвимы к атакам на основе открытого текста, чем к атаке грубой силы, требующей использования 16 раундов.

DES давно характеризуется низкой криптостойкостью: в январе 1999 г. закодированное посредством DES сообщение было взломано с помощью связанных через Internet в единую сеть 100 тыс. персональных компьютеров за 24 часа. Данному алгоритму присуща проблема так называемых «слабых» и «частично слабых» ключей.

Основное достоинство DES – относительно высокая скорость (из-за малой длины ключа); бесплатное распространение по всему миру; общедоступность и отсутствие необходимости лицензионных отчислений.

Модификацией DES является 3DES. Создан У. Диффи, М. Хеллманом, У. Тачманном в 1978 г. Формальная запись:



Cуществуют 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.

Расшифрование происходит, как и в простом DES, в обратном порядке по отношению к процедуре зашифрования.

3DES с тремя ключами реализован во многих Интернет-приложениях. Например, в PGP (Pretty Good Privacy) – позволяет выполнять операции шифрования и цифровой подписи сообщений, файлов и другой информации, представленной в электронном виде, например, на жёстком диске ); в S/mime для обеспечения криптографической безопасности электронной почты. 3DES используется при управлении ключами в стандартах ANSI X9.17 (метод генерации 64-битных ключей) и ISO 8732 (управление ключами в банковском деле), а также в PEM (Privacy Enhanced Mail).

1. **Практическая часть**

В данной лабораторной работе необходимо разработать пользовательское приложение, которое должно реализовывать алгоритм DES. При этом можно было воспользоваться готовыми библиотеками либо программными кодами, реализующими некоторые блочные алгоритмы.

Приложение должно реализовывать следующие операции:

* разделение входного потока данных на блоки требуемой длины с необходимым дополнением последнего блока;
* выполнение требуемых преобразований ключевой информации;
* выполнение операций зашифрования/расшифрования;
* оценка скорости выполнения операций зашифрования/расшифрования;

В связи с поставленными требованиями было разработано приложение, представленное на рисунке 2.1.



Рисунок 2.1 – Алгоритм DES (пользовательское приложение)

Шифрование осуществляется за счёт подключаемой библиотеки Cryptography, которая предоставляет некоторые блочные шифры.

Чтобы зашифровать некоторое приложение необходимо просто поместить его в соответствующее поле и нажать кнопку «Начать».

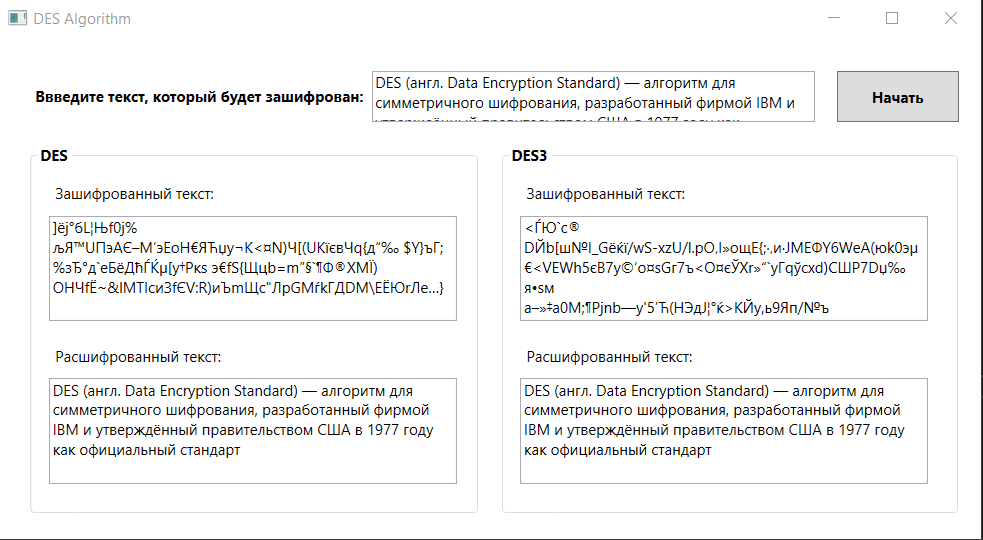


Рисунок 2.2 – Шифрование и расшифрование сообщения

Также необходимо было подсчитать время выполнения каждого метода, что также реализовано в пользовательском приложении. Время, потраченное на выполнение методов шифрования и расшифрования (в соответствии с вариантом с помощью DES и DES3) с входными данными из предыдущего примера (в качестве входной строки определение DES), представлено на рисунке ниже:

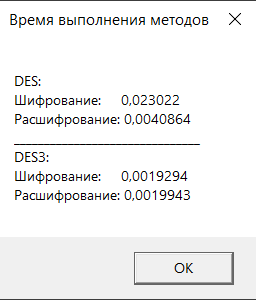


Рисунок 2.3 – Время, затраченное на выполнение методов

Более подробно пример шифрования алгоритмом DES будет рассматриваться в приложенном к данному отчёту файле DES.xlsx.

**Вывод**

В данной лабораторной работе я закрепила теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости блочных шифров. Разработала приложение для реализации указанных преподавателем методов блочного зашифрования/расшифрования, а именно алгоритмы DES и DES3. Выполнила анализ криптостойкости блочных шифров. Оценила скорость зашифрования/расшифрования реализованных шифров.