МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность Информационные системы и технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5 НА ТЕМУ:

Исследование блочных шифров

Выполнила:

Студентка 3 курса 1 группы ФИТ

Шимчёнок Елизавета Константиновна

**Цель:** изучение и приобретение практических навыков разработки и использования приложений для реализации блочных шифров.

**Задачи:**

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости блочных шифров.
2. Разработать приложение для реализации указанных преподавателем методов блочного зашифрования/расшифрования.
3. Выполнить анализ криптостойкости блочных шифров.
4. Оценить скорость зашифрования/расшифрования реализованных шифров.
5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**Теоретические сведения**

В 1972 г. Национальное бюро стандартов США (ныне – Национальный институт стандартов и технологии, *National Institute of* *Standarts* & *Technology* – *NIST*) инициировало программу защиты каналов связи и компьютерных данных. Одна из целей – разработка единого стандарта криптографического шифрования. Основными критериями оценки алгоритма являлись следующие:

- алгоритм должен обеспечить высокий уровень защиты;

- алгоритм должен быть понятен и детально описан;

- криптостойкость алгоритма должна зависеть только от ключа;

- алгоритм должен допускать адаптацию к различным применениям;

- алгоритм должен быть разрешен для экспорта.

В качестве начального варианта нового алгоритма рассматривался *Lucifer* – разработка компании IBM начала семидесятых годов. В основе указанного алгоритма использовались два запатентованных в 1971 г. Хорстом Фейстелем (*Horst Feistel*) устройства, реализующие различные алгоритмы шифрования, позже получившие название шифр (сеть) Фейстеля (*Feistel Cipher*, *Feistel Network*). В первой версии проекта *Lucifer* сеть Фейстеля не использовалась.

После многочисленных согласований, специальных конференций, где рассматривались в основном вопросы криптостойкости алгоритма, подлежащего утверждению в качестве федерального стандарта, в ноябре 1976 г. был утвержден стандарт *DES* (*Data Encryption Standard* – стандарт шифрования данных). Предполагалось, что стандарт будет реализовываться только аппаратно.

В 1981 г. *ANSI* одобрил *DES* в качестве стандарта для публичного использования (стандарт *ANSI Х*3.92), назвав его алгоритмом шифрования данных (*Data Encryption Algorithm* – *DEA*).

В 1987 г. были разработаны алгоритмы *FEAL* и *RC*2. Сети Фейстеля получили широкое распространение в 1990-е гг. – в годы появления таких алгоритмов, как *Blowfish* (1993), *TEA* (1994), *RC*5 (1994), *CAST*-128 (1996), *XTEA* (1997), *XXTEA* (1998), *RC*6 (1998) и др. На основе сети Фейстеля в 1990 г. в СССР был принят в качестве ГОСТ 28147–89 стандарт шифрования.

Предполагалось, что *DES* будет сертифицироваться каждые 5 лет. Срок действия последнего сертификата на территории США истек практически к концу ХХ в. К тому времени *DES* был вскрыт «лобовой атакой».

Все перечисленные стандарты и алгоритмы блочных шифров (БШ) строятся на основе подстановочных и перестановочных шифров, т. е. являются комбинационными. БШ относятся также к классу симметричных.

В симметричных системах Отправитель и Получатель используют один и тот же ключ К.

Блочное шифрование (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.

**Сеть Фейстеля**

Название конструкции Фейстеля 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).

Каждый входной блок шифруемого сообщения изначально делится на два подблока одинакового размера: левый (*L*0) и правый (*R*0).

Два блока преобразований:

- блок подстановок (*S*-блок)

Дешифратор преобразует n-разрядное двоичное число в одноразрядный сигнал по основанию 2*n*

Шифратор преобразует сигнал из одноразрядного 2*n*-ричного в *n*-разрядный двоичный

- блок перестановок (*P*-блок)

**Алгоритм DES**

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 (R0) и левую (L0) 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.

Базовые методы – подстановка и перестановка данных: 1 под. + 1 перест. – раунд (цикл).

Алгоритм состоит из 16 раундов, т.е. один блок данных длиной 64 бита обрабатывается 16 раз, в каждом из которых используется новый ключ: каждый раз биты ключа сдвигаются, затем из 56 бит выбирается 48 бит.

Блок перестановок изменяет положение цифр т.е. является линейным устройством.

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

- *DES*-*EEE*3: шифруется 3 раза с 3 разными ключами (операции шифрование-шифрование-шифрование);

- *DES*-*EDE*3: 3*DES* операции шифрование-расшифрование-шифрование с разными ключами;

- *DES*-*EEE*2 и *DES*-*EDE*2: как и предыдущие, однако на первом

и третьем шаге используется одинаковый ключ.

**Ход работы**

Был взят 1 вариант с алгоритмом *DES*.

Разработанное приложение реализовывает разделение входного потока данных на блоки требуемой длины с необходимым дополнением последнего блока, выполняет преобразование ключевой информации, шифрует и дешифрует сообщение.

На рисунке 1 изображен результат выполнения программы:

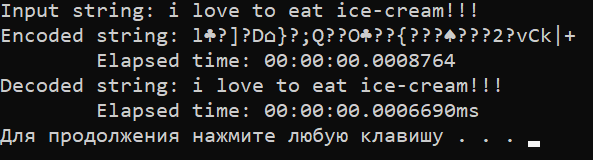
****

Рис. 1 – Консоль приложения после выполнения

В моем приложении ключ задается внутри кода. Так как *DES* принимает 64-битовые ключи, то в приложении нужно задать 8 символов. Я использовала: *shimliza*.

Как видно на рисунке 1, при помощи объекта для диагностики работы программы *Stopwatch* я смогла засечь скорость выполнения операций шифрования/дешифрования. Исходя из данных цифр, можно сказать о том, что шифрование требует больше времени, нежели дешифрование. Однако это не говорит ни о чем особенном.

Использование слабых и полуслабых ключей в алгоритмах блочного шифрования может представлять серьезную угрозу безопасности, поскольку они могут сделать шифрование уязвимым для атак. Слабый ключ – это ключ, который при использовании с определенным алгоритмом шифрования создает зашифрованный текст, который легко расшифровывается без знания ключа. Полуслабый ключ – это ключ, который создает несколько слабых ключей при использовании с алгоритмом шифрования.

Использование слабых и полуслабых ключей может позволить злоумышленникам использовать слабые места в алгоритме шифрования и потенциально восстановить открытый текст, не зная ключа.

Известно, что алгоритм стандарта шифрования данных (*DES*) имеет слабые и полуслабые ключи. Использование этих ключей может облегчить злоумышленникам проведение атак грубой силы и восстановление открытого текста. Чтобы уменьшить эту угрозу, важно использовать сильные ключи и избегать слабых и полуслабых ключей в алгоритмах блочного шифрования.

Лавинный эффект в криптографии, которым орудует *DES* и другие блочные шифры, снижает зависимость шифртекста от исходного сообщения, так как при изменении хотя бы одного бита в нем, выходная последовательность сильно изменяется.

Далее нужно было оценить степень сжатия исходного и зашифрованного текстов. Число бит в сжатом виде этих сообщений большее, нежели в исходных вариантах. Однако количество бит зашифрованного сообщения больше, чем в исходном сообщении. В сжатых сообщениях также.

**Ответы на вопросы**

1. Какие простейшие операции применяются в блочных алгоритмах шифрования?

Перестановка, подстановка, сложение по модулю, циклические сдвиги.

1. В чем отличие блочных алгоритмов шифрования от потоковых?

Основное различие между блочным шифром и потоковым шифром заключается в том, что блочный шифр зашифровывает и дешифрует блок текста одновременно. А потоковый шифр шифрует и дешифрует текст, беря один байт текста за раз.

1. Что понимается под «раундом» алгоритма шифрования?

Один из последовательных шагов обработки данных в алгоритме блочного шифрования.

1. Охарактеризовать и привести формальное описание сети Фейстеля.

Сеть Фейстеля – это тип блочных алгоритмов шифрования, который был предложен Хорстом Фейстелем в 1973 году. Она используется в таких известных алгоритмах, как *DES*, *Blowfish*, и *Twofish*.

Основная идея сети Фейстеля заключается в том, что исходный блок данных разбивается на две половины, и затем происходит ряд раундов, каждый из которых преобразует правую половину блока на основе левой половины и некоторого ключа.

После всех раундов происходит обмен местами левой и правой половин блока и повторение процедуры до тех пор, пока не будет достигнуто нужное количество раундов или пока не будет достигнут конечный результат.

1. Какие стандартные операции используются в блочных алгоритмах шифрования?

Сложение по модулю, циклические сдвиги, перестановки, подстановки. Эти операции выполняются несколько раз, образуя раунды.

1. В чем состоит особенность сложения чисел по модулю 2*n*?

Особенность сложения чисел по модулю 2*n* заключается в том, что результатом операции сложения является остаток от деления суммы чисел на 2*n*. То есть если мы складываем два числа a и b по модулю 2*n*, то результат будет равен (*a* + *b*) *mod* 2*n*.

1. Сложить по модулю 102 пары чисел: 55 и 14; 76 и 24; 99 и 99.

69 *mod* 102 = 69;

100 *mod* 102 = 100;

198 *mod* 102 = 96.

1. Сложить по модулю 28: двоичные числа 10101100 и 11001010; 01111111 и 01101101.

Для сложения двоичных чисел по модулю 28 мы должны использовать только последние 8 битов каждого числа и затем сложить эти биты по модулю 28.

В данном случае, последние 8 битов каждого числа равны:

10101100 -> 0101100

11001010 -> 1001010

0101100 + 1001010 = 1110110

Результат сложения по модулю 28 двоичных чисел 10101100 и 11001010 равен 1110110 = 118.

В следующем задании последние 8 битов каждого числа равны:

01111111 --> 1111111

01101101 --> 1101101

1111111 + 1101101 = 11001000

Результат сложения по модулю 28 двоичных чисел 01111111 и 01101101 равен 11001000 = 200.

1. Дать пояснение принципам реализации «лавинного» эффекта.

Лавинный эффект в криптографии – это свойство криптографической функции, при котором любое небольшое изменение входных данных приводит к значительному и непредсказуемому изменению выходных данных. Это означает, что даже небольшое изменение на входе, например изменение одного бита, вызовет резкое изменение на выходе.

Цель лавинного эффекта – повысить безопасность криптографических алгоритмов, затруднив получение злоумышленником какой-либо полезной информации о входных данных или ключе, используемом для шифрования. Это свойство важно во многих криптографических приложениях, включая шифрование, хэш-функции и схемы цифровой подписи.

1. Выбрать два произвольных блочных алгоритма. В чем состоят отличия между ними?

*AES* является одним из самых популярных блочных алгоритмов, используемых в настоящее время. Он использует блок размером 128 бит и ключи длиной 128, 192 или 256 бит. Одной из главных особенностей *AES* является его высокая степень безопасности, обусловленная сильной нелинейностью и необратимостью функций замены и перестановки. Это делает атаки на алгоритм *AES*, основанные на анализе статистики блоков, неэффективными.

*DES* является одним из старейших блочных шифров, используемых до сих пор несмотря на то, что его ключ длиной 56 бит уже давно считается небезопасным. *DES* использует блок размером 64 бит и ключ. Одной из особенностей *DES* является использование множества преобразований блока, включая перестановки, замены и циклические сдвиги. Это делает *DES* относительно медленным алгоритмом, но при этом он остается достаточно безопасным при использовании достаточно длинных ключей.

Основное отличие между *AES* и *DES* заключается в их безопасности и скорости работы. *AES* является более безопасным и быстрым алгоритмом, по сравнению с *DES*. Он также использует ключи более длинной длины и имеет более эффективную структуру. *DES* является устаревшим алгоритмом, который стал подвержен атакам со стороны хакеров, и его использование уже не рекомендуется для защиты конфиденциальных данных.

1. Представить графически и пояснить функционал одного раунда блочного алгоритма DES (АES, ГОСТ 28147–89, *Blowfish*).

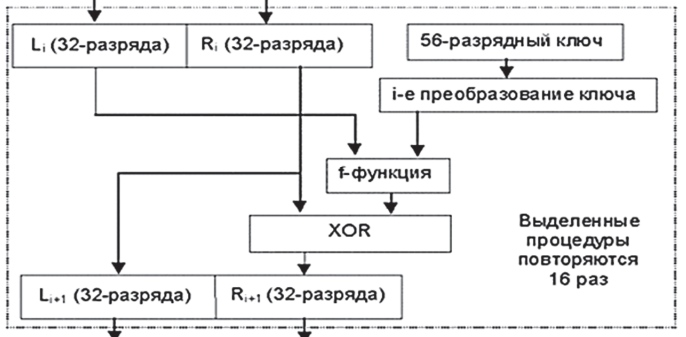


Рис. 2 – Схема одного раунда в БШ DES

Один раунд *DES* включает в себя следующие шаги:

- Начальная перестановка (*IP*): биты блока входных данных переставляются в соответствии с определенной таблицей перестановки.

- Разделение блока на левую и правую части: входной блок размером 64 бита разбивается на две части по 32 бита каждая.

- Функция расширения (*E*): правая часть блока размером 32 бита расширяется до 48 бит с помощью определенной таблицы расширения.

- Ключевая функция (*K*): 48-битный ключ, полученный из основного ключа *DES* в соответствии с определенным алгоритмом, складывается по модулю 2 с расширенной правой частью блока.

- Функция подстановки (*S*): результат функции K разбивается на 8 блоков по 6 бит, каждый из которых заменяется на 4-битное значение, выбираемое из соответствующей таблицы замены.

- Функция перестановки (*P*): результат функции *S* подвергается перестановке с помощью определенной таблицы перестановки.

- XOR с левой частью: результат функции P складывается по модулю 2 с левой частью блока.

- Обмен правой и левой частями блока: левая часть блока становится новой правой частью, а правая - новой левой.

1. Сколько можно реализовать (теоретически) разновидностей алгоритма 3*DES*?

3*DES* (*Triple DES*) – это блочный шифр, который использует блоки размером 64 бита и работает с тремя ключами, каждый из которых имеет длину 56 бит. Существуют два режима работы 3*DES*: *EDE* (*Encrypt*-*Decrypt*-*Encrypt*) и *EEE* (*Encrypt*-*Encrypt*-*Encrypt*).

Количество возможных вариантов 3*DES* можно оценить, учитывая следующее: cуществует три ключа, каждый из которых имеет длину 56 бит. Это означает, что можно создать 2 в 56 степени различных ключей для каждой из трех позиций. Существует два режима работы: *EDE* и *EEE*. Кроме того, можно изменить порядок ключей в режиме *EDE*, что дает еще один вариант. Таким образом, общее число возможных вариантов 3*DES* можно оценить как: 256 \* 256 \* 256 \* 2 \* 2 = 2168 \* 4.

1. Какие факторы влияют на стойкость блочного алгоритма шифрования?

* Размер блока;
* Длина ключа;
* Количество раундов.

1. В чем состоит сущность дифференциального криптоанализа?

Сущность дифференциального криптоанализа заключается в анализе различий (дифференциалов) между парами открытого текста и соответствующих им шифртекстов, полученных с использованием одного и того же ключа.

1. В чем состоит сущность линейного криптоанализа?

Сущность линейного криптоанализа заключается в анализе линейных зависимостей между битами ключа, открытым текстом и соответствующим шифротекстом. Линейный криптоанализ ищет линейную функцию (или комбинацию функций), которая наилучшим образом аппроксимирует зависимость между входными и выходными данными алгоритма шифрования.

1. Какие ключевые комбинации относятся к слабым (к полуслабым) и почему?

К слабым ключевым комбинациям в БШ относятся такие, что они позволяют провести атаки на шифр с помощью известных методов криптоанализа с меньшей вычислительной сложностью, чем при использовании случайно сгенерированных ключей.

Полуслабые комбинации относятся к ключам, которые можно использовать для взлома шифра в некоторых конкретных случаях, но общих методов их взлома не существует. Блочные шифры, такие как DES или 3DES, могут служить примерами таких шифров, если они используются с определенными значениями ключа.

1. Где применяются блочные криптоалгоритмы?

Блочные криптоалгоритмы используются для защиты электронной почты, обмена сообщениями и других форм цифровых данных.

Интернет-магазины и банковские операции полагаются на блочные криптоалгоритмы для защиты конфиденциальной информации, такой как номера кредитных карт и реквизиты банковского счета, во время транзакций.

Медицинские записи и данные пациентов часто шифруются с использованием блочных криптоалгоритмов для защиты конфиденциальности пациентов и предотвращения несанкционированного доступа.

Пправительство и военные используют блочные криптоалгоритмы для защиты секретной информации и защиты национальной безопасности.

Блочные криптоалгоритмы используются при торговле акциями, финансовых транзакциях и других областях финансов для предотвращения мошенничества и защиты конфиденциальной финансовой информации.

Поставщики облачных услуг также используют блочные алгоритмы шифрования для шифрования данных клиентов, гарантируя, что к ним не смогут получить доступ неавторизованные пользователи.

**Вывод:** в ходе лабораторной работы были приобретены практические навыки разработки и использования приложений для реализации блочных шифров. Также, был выполнен анализ криптостойкости блочных шифров, оценена скорость зашифрования/расшифрования и сделаны соответствующие выводы.