МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Ярославский государственный университет им. П. Г. Демидова»

Кафедра компьютерной безопасности и

математических методов обработки информации

**Курсовая работа**

**Исследование кодирования информации с помощью алгоритма шифрование Twofish**

Научный руководитель

Старший преподаватель кафедры

компьютерной безопасности

и математических

методов обработки информации

\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Саханда

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.

Студент группы КБ-41СО

И. Д. Сахаров

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.

Оглавление

[Введение 3](#_Toc112761294)

[Глава 1. Теоретическая часть 4](#_Toc112761295)

[Историческая справка 4](#_Toc112761296)

[Общие сведения о Twofish 5](#_Toc112761297)

[Архитектура шифра 6](#_Toc112761298)

[Отбеливание 8](#_Toc112761299)

[Сеть Фейстеля 8](#_Toc112761300)

[Структура раунда 10](#_Toc112761301)

[Функция F 11](#_Toc112761302)

[Функция g 12](#_Toc112761303)

[S-boxes 13](#_Toc112761304)

[MDS-матрицы 16](#_Toc112761305)

[Псевдо-преобразование Адамара 17](#_Toc112761306)

[Ключевое расписание 18](#_Toc112761307)

[Достоинства и недостатки 20](#_Toc112761308)

[Применение шифра 21](#_Toc112761309)

[Глава 2. Практическая часть 22](#_Toc112761310)

[Заключение 22](#_Toc112761311)

[Список использованных источников 23](#_Toc112761312)

Введение

Шифрование один из самых мощных и популярных способов обеспечения безопасности данных. На сегодняшний день огромному количеству организаций приходится работают с большим количеством персональных данных своих клиентов и сотрудников. Чтобы избежать юридических санкций и поддержать деловую репутацию, защита этих данных становится одной из первостепенных обязанностей. Шифрование помогает компаниям надежно хранить и передавать данные, сохраняя целостность и конфиденциальность.

Опорой всех современных технологий, которые работают с множеством взаимосвязанных объектов, лежит организация безопасности данных. Шифрование гарантирует, что конфиденциальная информация, которой обмениваются эти объекты, не подвержена утечке или раскрытию, что делает его неотъемлемой частью любой хорошей системы работы с данными.

В данной курсовой будет проведено исследование и анализ симметричного алгоритма блочного шифрования Twofish, а также реализация данного шифра на языке программирования C#.

**Объектом** курсовой работы являются алгоритм шифрования Twofish.

**Предмет** курсовой работы — обеспечение надежной безопасности информации при помощи алгоритма шифрования Twofish.

**Целью** курсовой работы является выявление преимуществ и недостатков алгоритма шифрования Twofish и его оценка.

Для достижения цели работы были поставлены следующие **задачи**:

Исследование архитектуры и принципа работы алгоритма шифрования Twofish.

Реализация алгоритма шифрования Twofish на языке програмирования C#.

Глава 1. Теоретическая часть

Историческая справка

В 1972 и 1974 годах Национальный институт стандартов и технологий (NIST) опубликовало первый публичный запрос на алгоритм шифрования для своего нового стандарта шифрования. Таким алгоритмом стал модернизированный компанией IBM алгоритм «Люцифер», который в дальнейшем получил название DES, возможно, наиболее широко используемым и успешный алгоритм шифрования в мире на протяжении долгих лет.

Несмотря на свою популярность, DES страдает от противоречий. Некоторые криптографы возражали против закрытого процесса разработки алгоритма и задавались вопросом, добавило ли АНБ люк, позволяющий тайно взломать алгоритм. Некоторые считали 56-битный ключ слишком коротким и на то время, а к нулевым годам, с ростом производительности процессоров, методом перебора ключа алгоритм было возможно взломать на домашнем персональном компьютере.

Triple-DES появился как промежуточное решение для банковских и других консервативных систем, но для некоторых целей он слишком медленный. Что более важно, 64-битная длина блока, разделяемая DES и большинством других надежных шифров, открывает его для атак, когда большие объемы данных шифруются одним и тем же ключом.

В ответ на растущее желание заменить DES, NIST анонсировал программу Advanced Encryption Standard (AES) в январе 1997 года. Срок подачи заявок истекал в июне 1998 года, и 15 участников представили свои алгоритмы миру в августе на Первой конференции кандидатов AES. NIST проведет вторую конференцию кандидатов AES в Риме в марте следующего года и будет принимать общественные комментарии по алгоритмам до 15 июня 1999 года. Он выберет пять финалистов, в число которых попали: Rijndael, Serpent, Twofish, RC6, MARS, и проведет еще один раунд общественного обсуждения, проведет третью конференцию кандидатов AES примерно в январе 2000 года, а затем выберет в качестве победителя Rijndael. NIST указал несколько критериев проектирования: более длинная длина ключа, больший размер блока, более высокая скорость и большая гибкость. Хотя ни один алгоритм не может быть оптимизирован для всех нужд, NIST сделал Rijndael стандартом симметричного шифрования следующих нескольких десятилетий.

Twofish - это [блочный шифр](https://en.citizendium.org/wiki/Block_cipher) от компании [Брюса Шнайера](https://en.citizendium.org/wiki/Bruce_Schneier) [Counterpane](https://en.citizendium.org/wiki/index.php?title=Counterpane&action=edit&redlink=1). Брюс Шнайер - всемирно известный специалист в области безопасности, которого журнал The Economist называет “гуру безопасности”. В Dr. Dobb’s Journal Брюс Шнайер говорит: “Джон Келси, Крис Холл, Нильс Фергюсон, Дэвид Вагнер, Дуг Уайтинг и я разработали Twofish, чтобы он был быстрым, гибким и безопасным. Он консервативен — в нем нет радикально новых идей безопасности или элементов дизайна. И это совершенно бесплатно — нет никаких патентных отчислений за алгоритм, авторских прав на код или лицензионных платежей за что-либо. Мы не подавали заявку на патент на Twofish и не планируем этого делать.”. Да, Twofish открыт к использованию и найдется большое количество криптографов считающих, что Twofish намного безопаснее, чем алгоритм Rijndael, который сегодня является широко используемым алгоритмом и рекомендован АНБ

За исключением названия и использования S-блоков, зависящих от ключа, он имеет мало общего со своим предком [Blowfish](https://en.citizendium.org/wiki/Blowfish_(cipher)). Twofish имеет новую архитектуру. У него есть преемник под названием Threefish, используемый в алгоритме хеширования [Skein](https://en.citizendium.org/wiki/Skein_(hash_algorithm)).

Общие сведения о Twofish

Twofish разрабатывался специально с учетом требований и рекомендаций NIST для AES:

* 128-битный блочный симметричный шифр
* Длина ключей 128, 192 и 256 бит
* Отсутствие слабых ключей
* Эффективная программная (в первую очередь на 32-битных процессорах) и аппаратная реализация
* Гибкость (возможность использования дополнительных длин ключа, использование в поточном шифровании, хэш-функциях и т. д.)
* Простота алгоритма - для возможности его эффективного анализа

Twofish - 128-битный блочный шифр, который принимает ключ длины 128, 192, 256 бит. Twofish, являясь симметричным алгоритмом шифрования, то есть использует один ключ как для шифрования данных, так и для их расщифровки. Шифр имеет входное, выходное отбеливания и 16 раундов сети Фейстеля, каждый из которых обладает биективной функцией F, четырех зависящих от ключа S-блоков, фиксированной MDS матрицы, псевдопреобразования Адамара, побитовых поворотов, и тщательно разработанное ключевое расписание.

Отличительными особенностями Twofish являются использование предварительно вычисленных S-блоков, зависящих от ключа, и относительно сложное ключевое расписание.. Одна половина n-битного ключа используется в качестве фактического ключа шифрования, а другая половина n-битного ключа используется для модификации алгоритма шифрования (зависящие от ключа S-блоки). Twofish заимствует некоторые элементы из других конструкций, например, преобразование псевдо-Адамара из семейства шифров SAFER.

Архитектура шифра

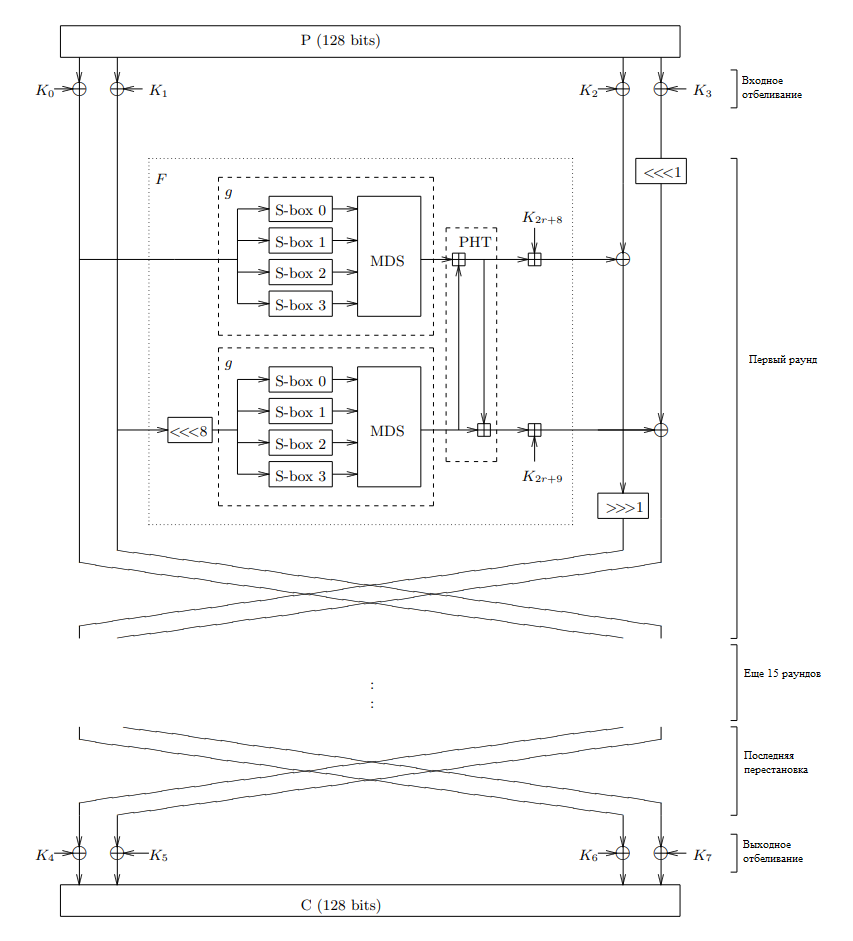
В Twofish используется достаточно сложную структуру, которая сильно усложняет его анализ, что является как плюсом так и минусом алгоритма. В алгоритме можно выделить несколько основных элементов участвующих в процессе шифрования:

* **Отбеливание.** Метод до первого и после последнего раундов, при котором над частью ключевого материала и входными блоками открытого текста выполняется операция исключающего «или» (XOR).
* **Сеть Фейстеля. М**етодов построения [блочных шифров](https://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80), состоящий из многократно повторяющихся раундов, на каждом из которых вычисляется функция F от части шифруемого блока, результат которой складывается с другой частью этого же блока**.**
* **S-боксы.**Широко распространённый в блочных шифрах метод нелинейной замены бит. В Twofish S-блоки зависит от ключевого материала, принимают 8 бит на входе и возвращают 8 бит на выходе.
* **MDS(**maximum distance separable) **- матрицы разделения на максимальное расстояние.** В Twofish используется заранее известная MDS матрицу размером 4 на 4 над полем Галуа. Она используется для перемешивания ключа, чтобы гарантировать, что небольшое изменение ключа создаст совершенно новый зашифрованный текст на выходе шифра.
* **Псевдо-преобразование Адамара (PHT).**Обратимое преобразование битовых строк, заключающиеся во взаимном сложении по модулю двух выходов g-функций.
* **Ключевое расписание.** Метод создания множества различных раундовых ключей из одного ключевого материала, которые будут использованы на каждом раунде, процессе **отбеливания и определении S-боксов**

**Помимо всего вышеперечисленного в алгоритме используются следующие операции работы c информацией, представленной в двоичном коде:**

|  |  |  |
| --- | --- | --- |
| Схема | **Операция** | **Уравнение** |
|  | **Сложение** | **Z = X + Y** |
|  | **Исключающее ИЛИ** | **Z = X XOR Y** |
|  | **Циклический сдвиг влево** | **Z = X ROL Y** |
|  | **Циклический сдвиг вправо** | **Z = X ROR Y** |

Ниже представлена общая схема алгоритма Twofish:

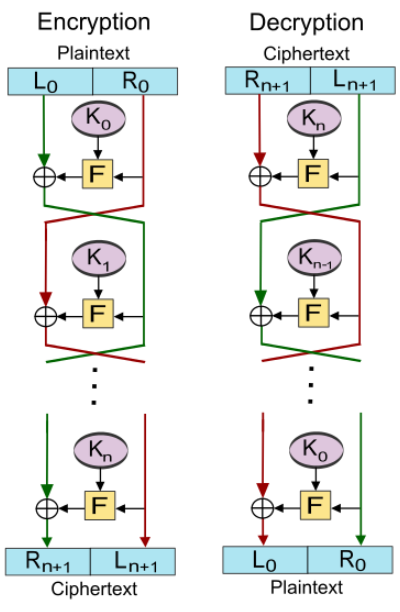


Отбеливание

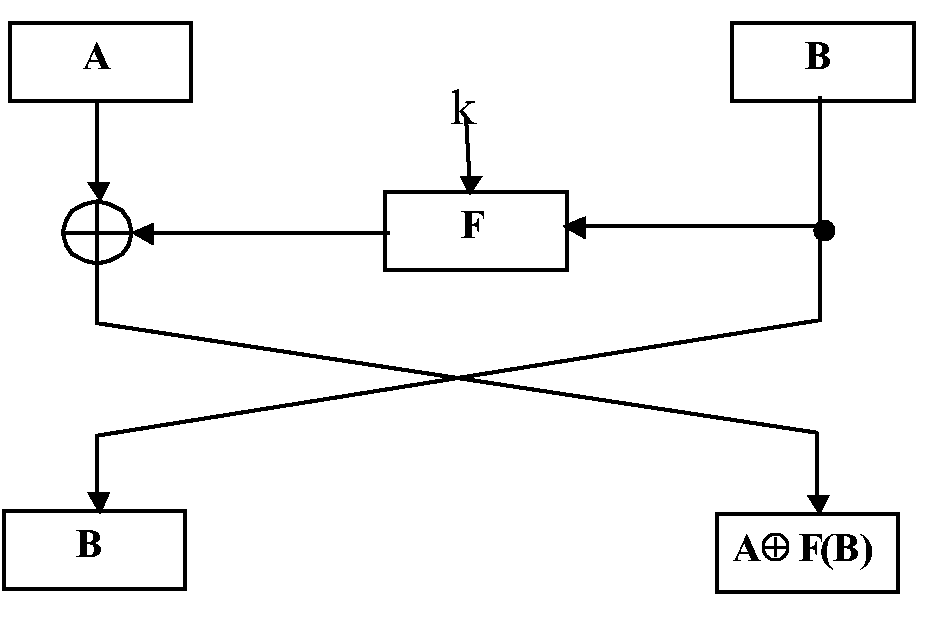
Отбеливание – метод повышения надежности алгоритма шифрования, путем объединения данных с частями ключа. В Twofish оно происходит сразу после получения нового блока открытого текста, то есть перед первым раундом и после последнего раунда сети Фейстеля. На процесс отбеливания поступает 128 бит текста, который заранее поделен на 4 блока по 32 бита. С каждым из этих блоков складываются по модулю 2 ключи K0, K1, K2, K3 на входном отбеливании и ключи K4, K5, K6, K7 на выходном отбеливании, которые были заранее посчитаны по тому же алгоритмы что и раундовые ключи. Способ их вычисления будет описан в главе Key schedule. Отбеливание существенно увеличивает сложность атаки на шифр, скрывая от атакующего конкретные входные данные для функций F первого и последнего раундов.

Сеть Фейстеля

Сеть Фейстеля это способ организовать блочный алгоритм шифрования таким образом, что криптограф не должен заботиться о том чтобы шифр был восстановим. Так же эта методика обеспечивает выполнение требования о многократном использовании ключевого материала и блока открытого текста. В общем варианте сеть Фейстеля имеет следующий вид:



При шифровании в сети Фейстеля блок открытого текста делится на части или ветви. К правой ветви и к раундовому ключу применяется некоторая образующая функция Фейстеля, результат которой накладывается на другую ветвь, а дальше происходит обмен ветвей местами, что означает конец раунда. Структура раунда или как принято называть ячейки Фейстеля схематично в общем виде выглядит следующим образом:



В результате одного раунда сети Фейстеля меняется половина текста. Эти раунды повторяются множество раз, благодаря чему обеспечивается эффект надежного шифра. Особенностью такого способа организации алгоритма является то, что для расшифрования не требуется проходить через функцию Фейстеля в обратную сторону, то есть функция Фейстеля однонаправленно и криптографу не нужно заботиться об обратимости функции F. Обратимость функции шифрования гарантируется конструкцией ячейки Фейстеля. Для расшифрования шифртекста используется та же самая функция F. Покажем это.

В i-ом раунде на вход попадают два блока данных – XR1 и XL1. В результате работы ячейки Фейстеля на выходе мы получаем

XL2 = XR1

XR2 = XL1 XOR Fk(XR1)

Для восстановления XR1 и XL1 из XL2 и XR2 нужно применить ту же функцию F с тем же ключом k

XR1 =XL2

XL1 = XR2 XOR Fk(XL2)

В Twofish используется 16 раундовая сеть Фейстеля.

Структура раунда

На вход раунда поступает блок открытого текста длинной 128 бит, который в последствии делится на 4 ветви Rr,0, Rr,1, Rr,2, Rr,3 по 32 бита каждая. Ветви Rr,0 и Rr,1 идут на вход функции Фейстеля. Результатом является 2 блока по 32 бита, один из которых складывается по модулю 2 с Rr,2 и выполняется циклический сдвиг вправо, а второй блок складывается по модулю 2 с Rr,3 к которому заранее было применен циклический сдвиг влево.

Алгоритм раунда описывается следующим образом

(Fr,0, Fr,1) = F(Rr,0, Rr,1, r)

Rr+1,0 = ROR(Rr,2 XOR Fr,0, 1)

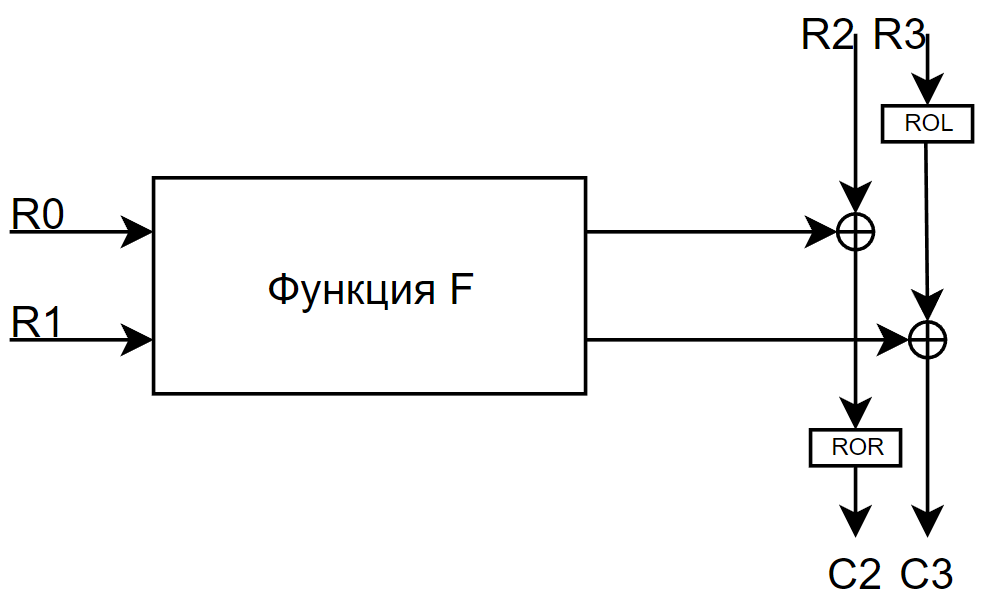
Rr+1,1 = ROL(Rr,3, 1) XOR Fr,1

Rr+1,2 = Rr,0

Rr+1,3 = Rr,1

Где r = 0,…,15 – номер раунда.

Схематично алгоритм можно изобразить как показано на схеме ниже



Блоки C2, C3, R0, R1 являются результатом раунда.

Функция F

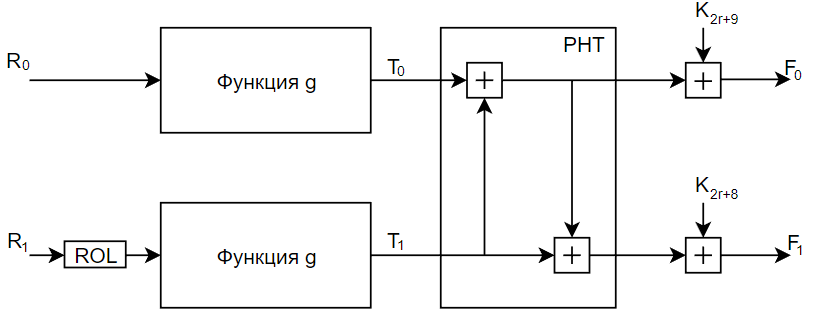
Функция F есть функция Фейстеля. Она получает на вход две ветви по 32 бита и номер раунда r, а её результатом являются две последовательности по 32 бита, которые складываются с двумя другими ветвями. R0 передается в функцию g и на выходе получается T0, а к R1 сначала применяется операция циклического сдвига влево и затем он уже поступает на вход функции g для получения T1. Затем T0 и T1 попадают на вход PHT(Псевдо-преобразование Адамара) и полученные две последовательности складываются с раундовыми ключами K2r+8 и K2r+9.

T0 = g(R0)

T1 = g(ROL(R1,8))

F0 = (T0 + T1 + K2r+8) mod 232

F1 = (T0 + 2T1 + K2r+9) mod 232

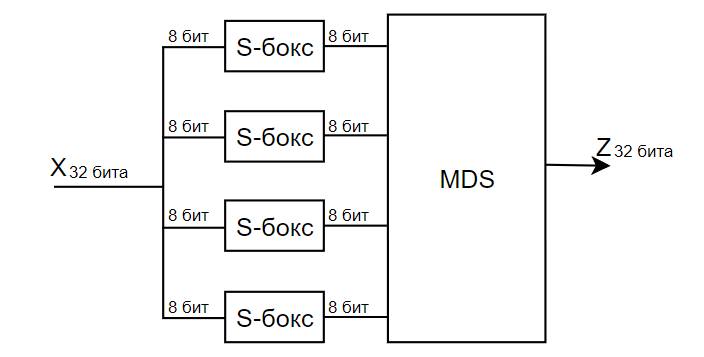


Функция g

Функция g это сердце алгоритма Twofish. Данная функция применяется дважды в функции F, а так же участвует в создании раундовых ключей. На вход функция принимает 32 бита и возвращает 32 бита. Входное значение разбивается на 4 части по 8 бит каждая. Каждая из этих 8-ми битных частей идет в S-бокс. S-бокс принимает и выдает на выходе 8 бит. 4 выхода S-боксов интерпретируются как вектор длины 4 над GF(28) и умножается на матрицу MDS. Результат интерпретируется как 32-ухбитная последовательность и именно она является результатом функции g.

xi = [X/28i] mod 28 i = 0,1,2,3

yi = si[xi] i = 0,1,2,3

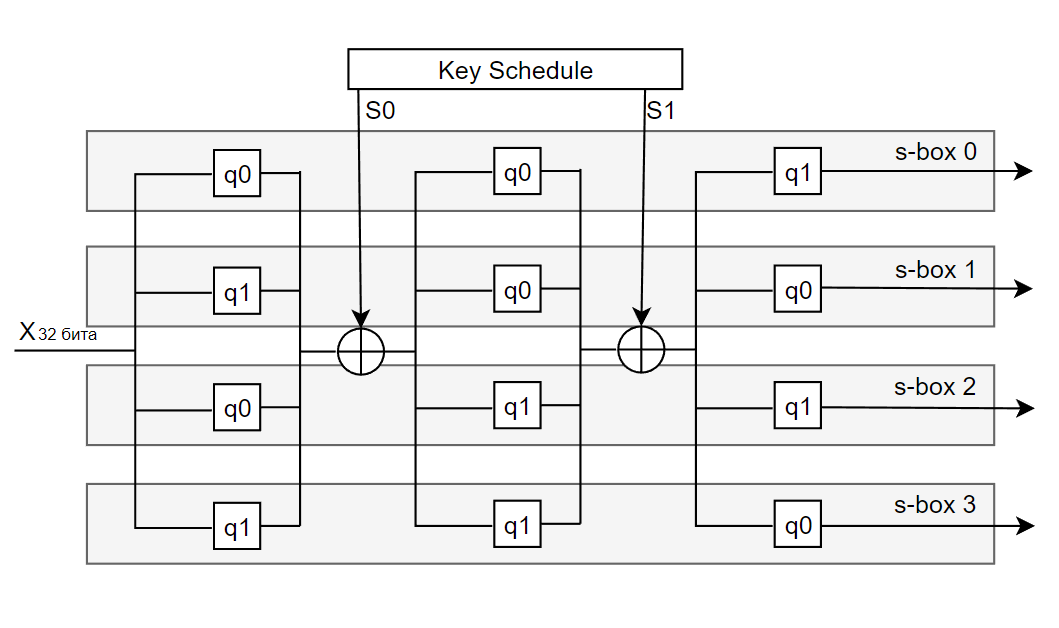


S-boxes

Блок подстановки или S-box является одним из основных компонентов большинства блочных шифров. S-блок принимает на вход n бит, преобразует их по определённому алгоритму и возвращает на выходе m бит. Функция заложенная в S-блок из себя представляет операцию нелинейной замены, управляемую таблицей поиска. Данные таблицы могут генерироваться динамически на основе ключа или заранее быть тщательно подобранными так, чтобы противостоять линейному и дифференциальному криптоанализу.

Twofish использует четыре разных, биективных, зависящих от ключа S-бокса размером 8 на 8 бит. Эти S-боксы построены с использованием двух фиксированных 8-на-8-битных перестановок q0 и q1 и ключевого материала.

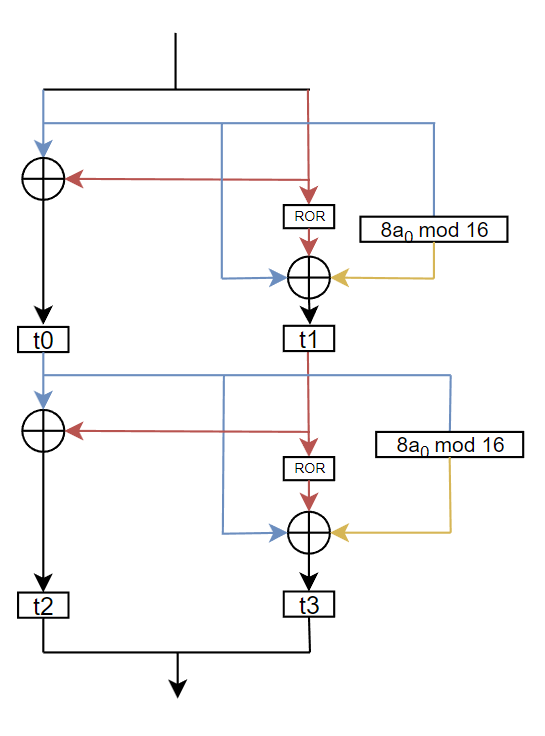
Каждый S-бокс имеет набор из трех перестановок составленного из двух функций q0 и q1. Между этими тремя перестановками есть операции XOR, выполняющие сложение по модулю два с ключевым материалом.



Каждая из перестановок q0 и q1 имеет одинаковую структуру и отличаются только используемыми таблицами поиска t0…t3.

Перестановки q0 и q1:

Схематично их можно изобразить следующим образом:



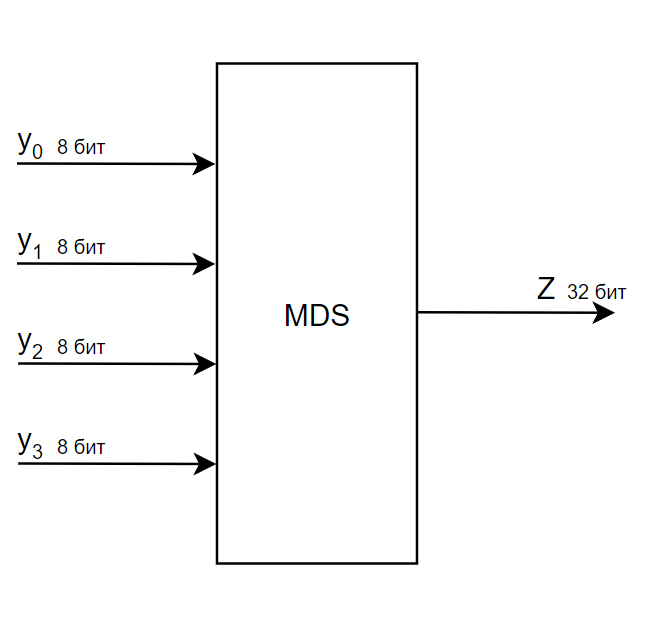
Таблицы поиска для q – перестановок представлены ниже.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| q0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| t0 | 8 | 1 | 7 | D | 6 | F | 3 | 2 | 0 | B | 5 | 9 | E | C | A | 4 |
| t1 | E | C | B | 8 | 1 | 2 | 3 | 5 | F | 4 | A | 6 | 7 | 0 | 9 | D |
| t2 | B | A | 5 | E | 6 | D | 9 | 0 | C | 8 | F | 3 | 2 | 4 | 7 | 1 |
| t3 | D | 7 | F | 4 | 1 | 2 | 6 | E | 9 | B | 3 | 0 | 8 | 5 | C | A |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| q1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| t0 | 2 | 8 | B | D | F | 7 | 6 | E | 3 | 1 | 9 | 4 | 0 | A | C | 5 |
| t1 | 1 | E | 2 | B | 4 | C | 3 | 7 | 6 | D | A | 5 | F | 9 | 0 | 8 |
| t2 | 4 | C | 7 | 5 | 1 | 6 | 9 | A | 0 | E | D | 8 | 2 | B | 3 | F |
| t3 | B | 9 | 5 | 1 | C | 3 | D | E | 6 | 4 | 7 | F | 2 | 0 | 8 | A |

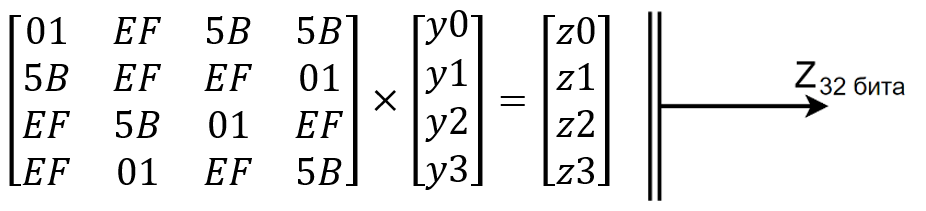
MDS-матрицы

Для того, чтобы шифр был стойким к атакам, в процессе шифрования должен создаваться лавинный эффект - свойство схемы, при котором изменение одного бита на входе значительно влияет на все биты на выходе. Лавинный эффект осуществляется за счёт перемешивания и рассеивания. Перемешивание обеспечивает как можно более сложную зависимость ключа и шифрованных данных и используется для того, чтобы усложнить задачу получения информации о ключе. Рассеивание - свойство алгоритма, при котором влияние одного бита открытого текста распространяется на большое количество битов шифротекста, скрывая статистические характеристики текста. MDS матрицы, умножение на которые используется в качестве линейного преобразования в блочных шифрах, является механизмом диффузии.



Это умножение матрицы в Twofish обеспечивают свойство рассеивания для противодействия атакам на основе статистических характеристик. Свойство MDS здесь гарантирует, что количество измененных входных байтов плюс количество измененных выходных байтов составляет не менее пяти. Другими словами, любое изменение в одном входном байте гарантированно изменит все четыре выходных байта, любое изменение в любых двух входных байтах гарантированно изменит по крайней мере три выходных байта и т.д. Существует более 2127 таких матриц MDS, но матрица MDS Twofish также тщательно выбирается со свойством сохранения количества измененных байтов, даже после преобразований в функции F.

В Twofish четыре байта, выводимые из четырех S-блоков, представляются как вектор и умножаются на матрицу MDS 4 на 4. Умножение выполняется в поле Галуа GF(28).



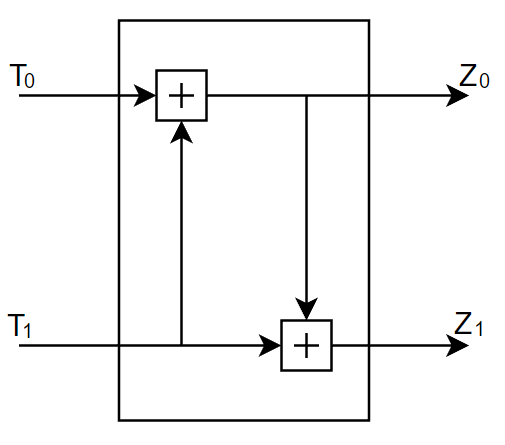
Псевдо-преобразование Адамара

Псевдо-преобразование Адамара (PHT) - обратимое преобразование битовых строк, используемое в криптографии для обеспечения свойства рассеивания для противодействия атакам на основе статистических характеристик. Получая два входа, T0 и T1 в виде 32-разрядных строк, в результате действия псевдо-преобразования Адамара получим две строки Z0 и Z1, значения которых вычисляются по следующим формулам:

Z0 = T0 + T1 mod 232

Z1 = T0 + 2T1 mod 232

Схематично псевдо-преобразование Адамара изображается следующим образом



Обратное псевдо-преобразование Адамара получается очень просто

T1 = Z1 - Z0 mod 232

T0 = 2Z0 - Z1 mod 232

Псевдо-преобразование Адамара может быть представлено в матричной форме. Представим T и Z в векторной форме

T = , Z =

Псевдо-преобразование Адамара будет равносильно умножению на матрицу

H =

по модулю 232.

Ключевое расписание

Ключевое расписание – средство, с помощью, которого из битов ключа получаются раундовые ключи, которое уже используются на протяжении всего алгоритма шифрования Twofish в различных местах. Благодаря этому на вход шифру требуется меньшее количество ключевых битов, чем требуется для шифра, но при этом на каждом раунде используются уникальные раундовые ключи.

Входной ключ M имеет длину N равную 128, 192, 256 бит. Из него строится два набора ключей K и S. Набор ключей S состоит из двух ключей S0 и S1, которые используются в S-боксах. Набор ключей K состоит из 40 ключей K0, K1,…,K39, каждый из которых имеет длину 32 бита:

* 4 ключа для входного отбеливания
* 4 ключа для выходного отбеливания
* 2 ключа на каждом раунде, что в сумме дает 2\*16 = 32 ключа

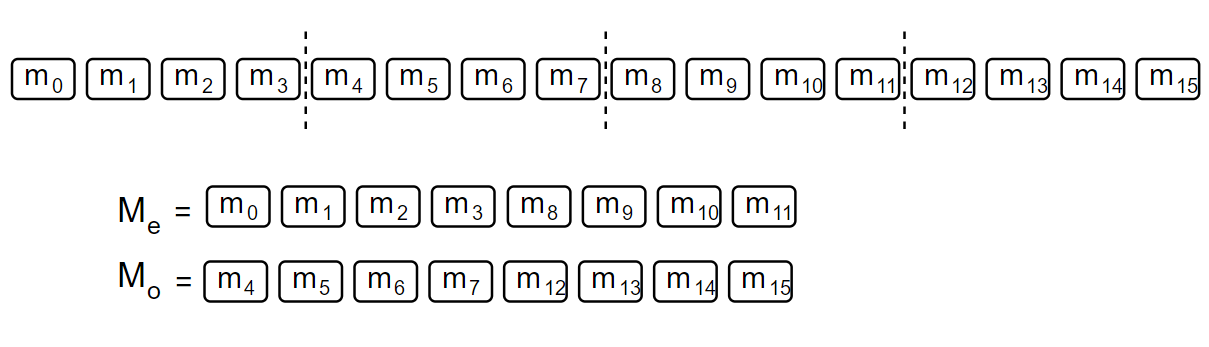
В начале алгоритм находит k = N/64. Получается что M состоит из 8k байтов m0,…,m8k-1. Ключ M делится на 2k частей по 32 бита следующим образом

Mi =

И эти части распределяются по двум векторам

Me  = (M0, M2, … , M2k-2)

Mo  = (M1, M3, … , M2k-1)



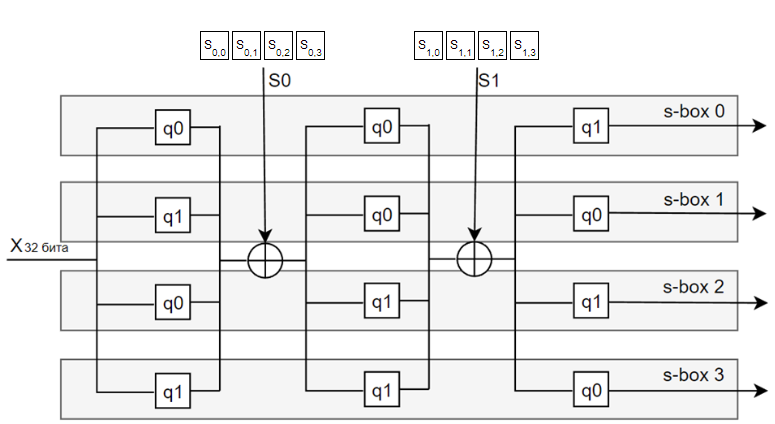
Так же еще нужно найти третий вектор длиной k. Это делается путем взятия байтов ключа в группах по 8, интерпретируя их как вектор на GF(28) и умножая на RS матрицу размера 4 на 8. Результат умножения интерпретируется как 32-битные слова, которые и составляют 3-ий вектор S.

для i = 0, … , k-1

S = (Sk-1, Sk-2, … , S0)

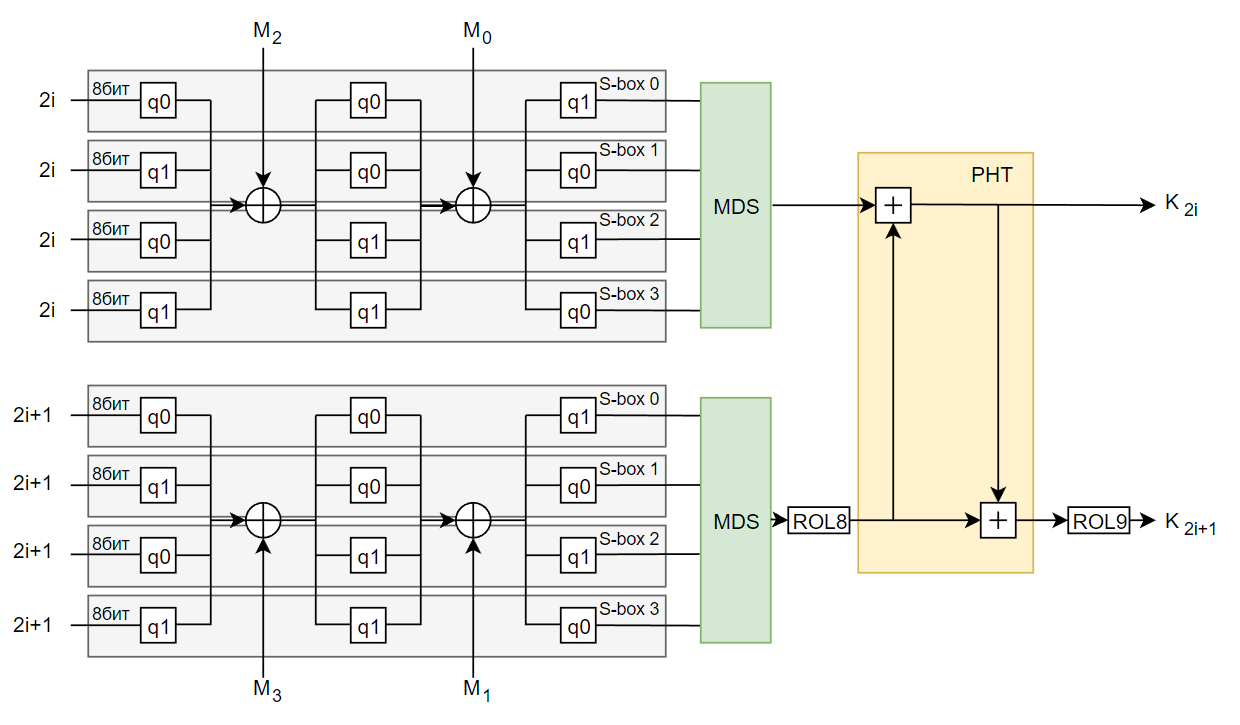
Эти три вектора Me, Mo и S составляют основу для расписания ключей.

Ключи вектора S используются в функции g



Вектора Me, Mo поступают на вход функции h, которая генерирует ключи K0, K1,…,K39.

Функция h для k = 4 схематично изображена ниже.



Достоинства и недостатки

[Согласно анализу IEEE](https://ieeexplore.ieee.org/document/5966408), алгоритм Twofish отстает по скорости **шифрования текста и изображений по сравнению с сегодняшним стандартом шифрование AES**. Но при достаточном увеличении объема оперативной памяти алгоритм Twofish был **быстрее для шифрования текста** и на одном уровне с AES для шифрования изображений. Это указывает на требовательность к ресурсам у Twofish. Алгоритм Twofish по своей конструкции является сложным и использует **16 раундов** независимо от используемой длины ключа. Все это приводит к тому, что в большинстве случаев он **медленный и неэффективный**. Преимущество AES также в том, что высокопроизводительные современные процессоры x86 и ARM включают аппаратное ускорение для него, что делает его несправедливым по отношению к практически любому другому шифрованию.

На Twofish нет практических атак, даже для сокращенных вариантов. [Единственные атаки являются теоретическими](http://www.drdobbs.com/security/the-twofish-encryption-algorithm/184410744), но вычислительно невозможными из-за их **большой временной сложности**.

Для большинства приложений алгоритм AES, вероятно, является лучшим вариантом, поскольку он достаточно **быстрый и** [безопасный](https://cloudstorageinfo.org/how-secure-is-cloud-storage). Но если у вас есть конфиденциальная информация, которую вы хотите защитить, и **производительность не является проблемой**, алгоритм Twofish – хороший вариант.

Применение шифра

Любой может использовать Twofish без ограничений, поскольку его создатели не запатентовали его и сделали доступным для общественности. Фактически, это один из немногих шифров, включенных в стандарт OpenPGP (RFC 4880), наиболее широко используемый на сегодняшний день стандарт шифрования электронной почты.

Алгоритм Twofish может использоваться в различных направлениях и широко применяться в системах с разными уровнями требований к безопасности, так как стойкость данного алгоритма достаточно высока, а сам алгоритм не столь популярен, что увеличивает надежность его применения.

Twofish помимо своего прямого назначени – сокрытия информации от [неавторизованных](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) лиц с предоставлением в это же время [авторизованным](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) пользователям доступа к ней, может быть использован иначе:

* в построении односторонних хеш-функций, диапазон применения которых очень широк: проверка целостности сообщений и файлов, верификация пароля, аутентификационные сообщения, цифровая подпись…
* лежать в основе генератора псевдослучайных чисел.

**Продукты, использующие шифрование Twofish:**

* **PGP** [Pretty Good Privacy (PGP):](https://www.techslang.com/definition/what-is-pretty-good-privacy/)  [компьютерная программа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), также библиотека функций, позволяющая выполнять операции шифрования и цифровой подписи сообщений, файлов и другой информации. Эта программа держит в своем арсенале алгоритм Twofish.
* **GnuPG:** GnuPG - реализация стандарта IETF OpenPGP, позволяет вам шифровать и подписывать ваши данные и сообщения; он оснащен универсальной системой управления ключами, а также модулями доступа ко всем видам каталогов открытых ключей. GnuPG, также известный как GPG, представляет собой инструмент командной строки с функциями для легкой интеграции с другими приложениями. Доступно множество [интерфейсных приложений](https://www.gnupg.org/software/frontends.html) и [библиотек](https://www.gnupg.org/software/libraries.html).
* VeraCrypt**:** VeraCrypt - бесплатное программное обеспечение с открытым исходным кодом для создания и поддержки на лету зашифрованного тома (устройства хранения данных). Шифрование на лету означает, что данные автоматически шифруются непосредственно перед сохранением и расшифровываются сразу после загрузки без какого-либо вмешательства пользователя. Никакие данные, хранящиеся на зашифрованном томе, не могут быть прочитаны (расшифрованы) без использования правильных паролей / файлов ключей или правильных ключей шифрования. Вся файловая система зашифрована
* **KeePass:** KeePass - это программное обеспечение с открытым исходным кодом для хранения и [управления паролями](https://www.techslang.com/definition/what-is-password-management/), которое использует Twofish для шифрования паролей своих баз данных.. Он также использует Twofish для создания паролей для своих пользователей.

Глава 2. Практическая часть

Заключение

Список использованных источников

1. Н.Р.Спиричева Алгоритмы блочной криптографии : учебно-методическое пособие / Н. Р. Спиричева. – Екатеринбург : Изд-во Урал. Ун-та, 2013. – 78, [2] c.
2. Twofish: A 128-Bit Block Cipher / Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson – Minneapolis : [б. и.], 2015. – 68 p.
3. P. Preneel, V. Rijmen, and A. Bosselaers: “Principles and Performance of Cryptographic Algorithms” Dr.Dobb’s Journal, v. 23, n. 12, 1998, pp 126 – 131.
4. «Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard (AES)» (англ.). Department of Commerce -- National Institute of Standards and Technology -- Federal Register: September 12, 1997
5. Nechvatal J., Barker E., Bassham L., Burr W., Dworkin M., Foti J., Roback E. «Report on the Development of the Advanced Encryption Standard (AES)» (англ.). -- National Institute of Standards and Technology.
6. N. Ferguson, J. Kelsey, B. Schneier, D. Whiting «A Twofish Retreat: Related-Key Attacks Against Reduced-Round Twofish» (англ.) Twofish Technical Report #6, February 14, 2000
7. Fauzan Mirza, Sean Murphy «An Observation on the Key Schedule of Twofish» (англ.) -- Information Security Group, Royal Holloway, University of London -- January 26, 1999
8. ГОСТ 34.13-2018 «Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров»
9. Гатченко Н.А., Исаев А.С., Яковлев А.Д. «Криптографическая защита информации» – СПб: НИУ ИТМО, 2012 – 142 с.
10. Основы криптографии: учебное пособие / В. И. Коржик, В. А. Яковлев. – СПб., ИЦ Интермедия, 2016 – 296 с. : илл.
11. Шнайер, Б. Прикладная криптография / Б. Шнайер. – М. : Триумф, 2002.
12. Защита информации. Блочные шифры. — URL: <https://www.youtube.com/watch?v=PbDOmI1iM64&t=725s>
13. Pseudo-Hadamard transform. — URL: <https://en.wikipedia.org/wiki/Pseudo-Hadamard_transform>
14. Feistel cipher. — URL: <https://en.wikipedia.org/wiki/Feistel_cipher>
15. Twofish vs AES Encryption. — URL: <https://cloudstorageinfo.org/twofish-vs-aes-encryption>
16. Twofish. — URL: <https://www.techtarget.com/searchsecurity/definition/Twofish>