**Оглавление**

[ВВЕДЕНИЕ 5](#_Toc91269789)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_Toc91269790)

[1.1. Алгоритмы блочного шифрования 7](#_Toc91269791)

[1.2. Сеть Фейстеля 8](#_Toc91269792)

[1.3. ГОСТ 28147-89 10](#_Toc91269793)

[1.4. Режим электронной кодировочной книги (простой замены) 14](#_Toc91269794)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 16](#_Toc91269795)

[ОТЗЫВ 17](#_Toc91269796)

[ЗАКЛЮЧЕНИЕ 18](#_Toc91269797)

[СПИСОК ЛИТЕРАТУРЫ 19](#_Toc91269798)

[3. ПРИЛОЖЕНИЕ 20](#_Toc91269799)

[3.1. Исходный код 20](#_Toc91269800)

# ВВЕДЕНИЕ

В данной курсовой работе представлена реализация криптографического алгоритма блочного шифрования, утвержденного стандартом ГОСТ 28147-89 (в настоящий момент пересмотрен как ГОСТ 34.12-2018, также описан в RFC8891), в режиме электронной кодовой книги - ECB на языке программирования Java версии 17 (версия JDK 17.0.1). Указанным стандартом определено, что размер блока составляет 64 бита, а размер ключа 256 бит.

Согласно извещению ФСБ России о порядке использования алгоритма блочного шифрования ГОСТ 28147-89 от 01.07.2019 года данный алгоритм применяется в криптографических методах обработки и защиты информации, не содержащей сведений, составляющих государственную тайну.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Согласно принципам, изложенным Клодом Шенноном в статье "Теория связи в секретных системах", стойкость блочного шифра определяется характеристиками рассеивания и перемешивания, определенных следующим образом:

* рассеивание - распространение влияния одного знака открытого текста на множество знаков шифротекста, что позволяет скрыть статистические свойства открытого текста; если данное условие выполняется, то при шифровании двух блоков данных с минимальными отличиями между ними должны получаться совершенно непохожие друг на друга блоки шифротекста. Точно такая же картина должна иметь место и для зависимости шифротекста от ключа: один символ (бит) ключа должен влиять на несколько символов (битов) шифротекста;
* перемешивание - сокрытие взаимосвязи статистических свойств открытого и шифротекста; если шифр достаточно хорошо "перемешивает" биты исходного текста, то соответствующий шифротекст не содержит никаких статистических и тем более функциональных закономерностей для стороннего наблюдателя, обладающего лишь ограниченными вычислительными ресурсами.

## Алгоритмы блочного шифрования

Отличительная особенность алгоритмов блочного шифрования заключается в том, что в отличие от шифроблокнота, где для шифрования сообщения требуется равный ему по длине ключ, при преобразовании информации они используют блоки исходных данных фиксированной длины и получают равный по длине результирующий блок, применяя ключ сопоставимый по длине с размером 1 блока. Наиболее распространенные размерности блоков и ключей лежат в диапазоне от 64 до 256 бит и неизменны в рамках 1 алгоритма (варианта алгоритма).

То есть, принцип преобразования информации в алгоритмах блочного шифрования можно описать двумя функциями:

1. для шифрования - ,

2. для дешифрования - ,

где:

1. - исходное сообщение и результат дешифрования,

2. - результат шифрования,

3. - ключ.

## Сеть Фейстеля

Сеть Фейстеля - один из методов построения блочных шифров. Сеть представляет собой определённую многократно повторяющуюся (итерированную) структуру, называющуюся ячейкой Фейстеля. При переходе от одной ячейки к другой меняется ключ, причём выбор ключа зависит от конкретного алгоритма. Операции шифрования и дешифрования на каждом этапе очень просты, и при определённой доработке совпадают, требуя только обратного порядка используемых ключей.

В 1973 году Хорст Фейстель в журнале Scientific American опубликовал статью «Криптография и компьютерная безопасность» («Cryptography and Computer Privacy»), в которой раскрыл некоторые важные аспекты шифрования, а также ввел конструкцию, названную позже сетью Фейстеля. Эта схема была использована в проекте Lucifer фирмы IBM, над которым работал Фейстель и Дон Коперсмит. Этот проект был скорее экспериментальным, но стал базисом для DES - прародителя современных блочных шифров.

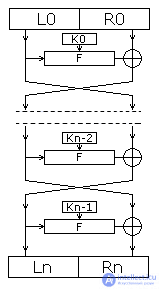


Рис. 1: Шифрование с использованием сети Фейстеля

Шифрование производится следующим образом:

1. Вся информация разбивается на блоки фиксированной длины. В случае, если длина входного блока меньше, чем размер, который шифруется заданным алгоритмом, то блок удлиняется каким-либо способом. Как правило длина блока является степенью двойки, например: 64 бита, 128 бит.

2. Выбранный блок делится на два равных подблока — «левый» (L\_0) и «правый» (R\_0).

3. «Левый подблок» L\_0 видоизменяется функцией в зависимости от раундового ключа K\_0, после чего он складывается по модулю 2 с «правым подблоком» R\_0.

4. Результат сложения присваивается новому левому подблоку L\_1, который будет половиной входных данных для следующего раунда, а «левый подблок» L\_0 присваивается без изменений новому правому подблоку R\_1 (см. схему), который будет другой половиной.

5. После чего операция повторяется N-1 раз, при этом при переходе от одного этапа к другому меняются раундовые ключи (K\_0 на K\_{1} и т. д.) по какому-либо математическому правилу, где N — количество раундов в заданном алгоритме.

Дешифрование же производится идентично с отличием в том, что порядок ключей инвертирован.

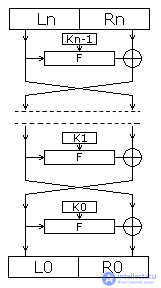


Рис. 2: Дешифрование с использованием сети Фейстеля

## ГОСТ 28147-89

В данном алгоритме сообщение разбивается на блоки длинной 64 бита, дополняясь при необходимости нулями (т.н. паддинг), если последний блок меньше по длине, чем необходимо.

В каждой итерации блок открытого текста делится на 2 длиной 32 бита и преобразовывается с использованием раундового ключа, поученного из исходного длиной 256 бит.

Во время каждой итерации, кроме 32, с правой и левой половиной зашифровываемого блока производится одно преобразование, основанное на сети Фейстеля. Сначала правая часть складывается по модулю 32 с текущим итерационным ключом, затем полученное 32-битное число делится на восемь 4-битных и каждое из них с использованием таблицы перестановки преобразуется в другое 4-битное число (нелинейное биективное преобразование).

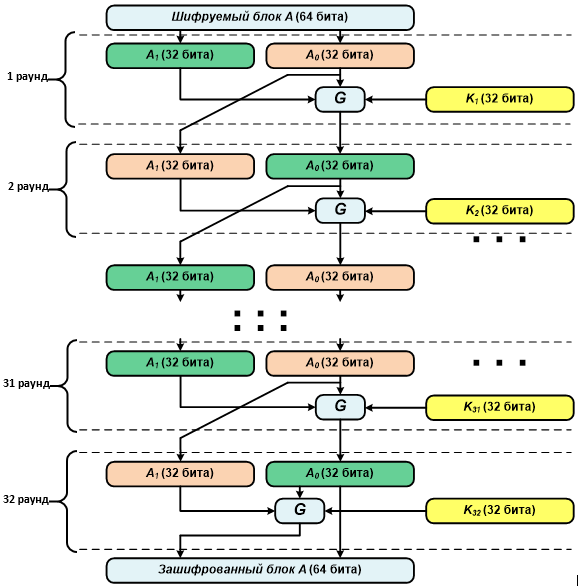


Рис. 3: Шифрование ГОСТ 28147-89

После этого преобразования полученное число циклически сдвигается влево на одиннадцать разрядов. Далее результат складывается по модулю 2 с левой половиной блока. Получившееся 32- битное число записывается в правую половину блока, а старое содержимое правой половины переносится в левую половину блока.

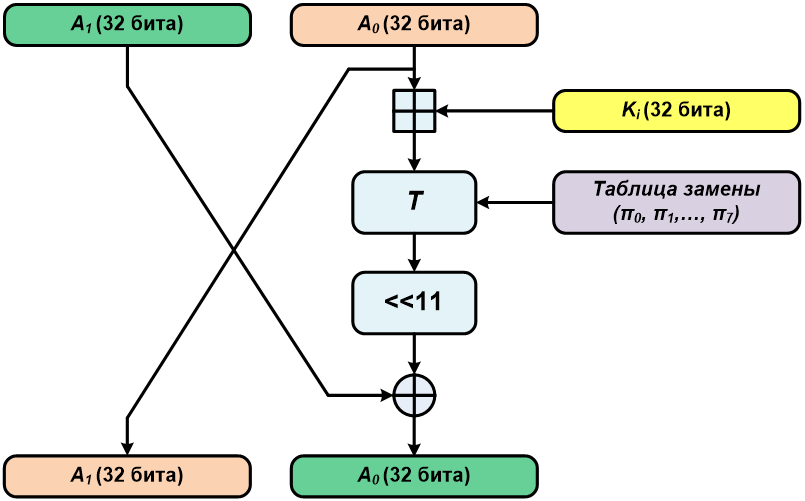


Рис. 4: Раунд шифрования ГОСТ 28147-89

В ходе последней (тридцать второй) итерации так же, как описано выше, преобразуется правая половина, после чего полученный результат пишется в левую часть исходного блока, а правая половина сохраняет свое значение. Итерационные ключи получаются из исходного 256-битного ключа. Исходный ключ делится на восемь 32-битных раундовых ключей, и далее они используются в следующем порядке: три раза с первого по восьмой и один раз с восьмого по первый.

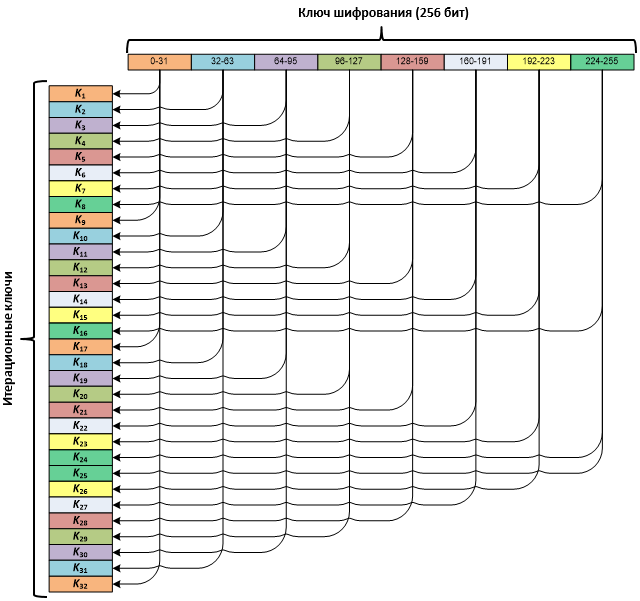


Рис. 5: Преобразование ключа по раундам

При дешифровании, как указано ранее, механизм идентичен, но порядок ключей инвертирован:

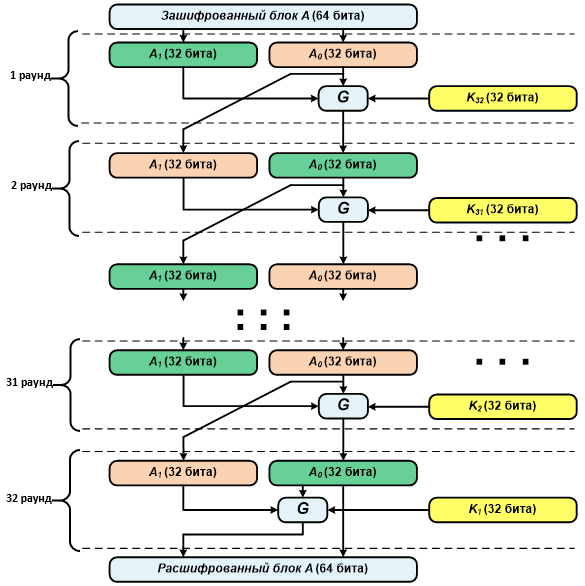


Рис. 6: Дешифрование ГОСТ 28147-89

При преобразовании шифр может быть использован в режиме:

1. ECB (Electronic Code Book) - электронной кодировочной книги,

2. CBC (Cipher Block Chaining) - сцепления блоков шифротекста,

3. CFB (Cipher Feedback) - обратной связи по шифротексту или

4. OFB (Output Feedback) - обратной связи по выходу.

Отличия режимов заключаются в следующем:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Характеристика** | **ECB** | **CBC** | **CFB** | **OFB** |
| Блоки, от которых зависит шифрование блока | Текущий | Все предыдущие | Все предыдущие | Позиция блока в файле |
| Результат искажения одного бита при передаче | Порча всего текущего блока | Порча всего текущего и всех последующих блоков | Порча одного бита текущего блока и всех последующих блоков | Порча одного бита текущего блока |
| Возможность кодирования без дополнения числа байтов, некратных блоку | Нет | Нет | Да | Да |
| Поступление на выход криптосистемы | Выход криптоалгоритма | Выход криптоалгоритма | XOR-маска с исходным текстом | XOR-маска с исходным текстом |

## Режим электронной кодировочной книги (простой замены)

Согласно стандарту, суть режима заключается в том, что каждый блок шифруемых данных обрабатывается алгоритмом шифрования отдельно и независимо от других блоков. Аналогично выполняется и дешифрование – блоки шифротекста обрабатываются поочередно и независимо. Режим ECB имеет следующие особенности:

* замены и перестановки отдельных блоков в шифротексте не нарушают корректности расшифрования;
* шифрование на одном ключе одинаковых блоков открытого текста дает одинаковые блоки шифротекста независимо от их положения в сообщении.

В режиме CBC перед шифрованием значение блока открытого текста суммируется по модулю 2 (операция XOR) со старшими битами значения счетчика CTR, реализованного на основе двоичного регистра сдвига длины m. Длина регистра должна быть кратной числу n битов в блоке открытого текста (n = 64 бита для «Магмы»). Сдвиг осуществляется на блок в сторону старших разрядов. При этом n последних (младших) битов регистра заполняются предыдущим блоком шифротекста.

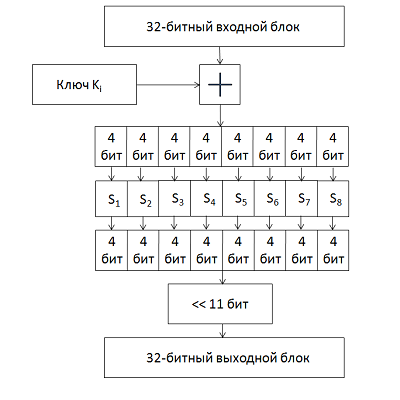


Рис. 6: Шифрование ГОСТ 28147-89 в режиме простой замены

При использовании алгоритма блочного шифрования в данном режиме информация может преобразовываться с использованием распараллеливания на необходимое (или возможное) число потоков шифрования (дешифрования), т.к. преобразования блоков открытого текста в блоки зашифрованного текста между собой никак не связаны. Тем самым можно существенно повысить производительность работы алгоритмов блочного шифрования в режиме простой замены. Однако данный режим имеет значительный недостаток, заключающийся в том, что одинаковые блоки открытого текста преобразуются в одинаковые блоки зашифрованного текста. Таким образом, в зашифрованном тексте могут встречаться повторяющиеся блоки, свидетельствующие об одинаковых блоках открытого текста, что очевидным образом раскрывает его структуру.

# ПРАКТИЧЕСКАЯ ЧАСТЬ

# ОТЗЫВ

Я, Бацаева Анна Павловна, студент 3-го курса группы ККСО-01-19, в период с 01.09.21 по 22.12.21 проходила производственную практику на Базовой кафедре №252 “Информационная безопасность”.

Во время производственной практики мною были выполнены поставленные преподавателем задачи и цели следующего характера:

* Изучение теоретического материала по алгоритму шифрования «Магма»;
* Изучение принципов работы алгоритма шифрования «Магма» на языке Java;
* Подготовка материала для написания отчета по производственной практике;
* Написание подробного отчета о проделанной мной работе.

Хотела бы выразить благодарность базовой кафедре за предоставление возможности применения теоретических навыков на реальных задачах, где были обеспечены благоприятные условия прохождения практики для студентов.

Студент-практикант « » \_\_\_\_\_\_\_ 2021 \_\_\_\_\_\_\_\_/Бацаева А.П.

# ЗАКЛЮЧЕНИЕ

В данной работе я реализовала алгоритм шифрования ГОСТ 28147−89. Шифр является устойчивым к атакам путём полного перебора.

# СПИСОК ЛИТЕРАТУРЫ

* 1. ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования»

2. Панасенко А.П., Зубков А.Ю., Кузьмин А.С., Черемушкин А.В. – «Основы криптографии. Учебное пособие, 2-ое издание»

3. Шнайер Б. – Прикладная криптография. Протоколы, алгоритмы и исходные тексты на языке С – М.: Издательский дом «Вильямс» – 2016. – 1040 с.

4. ГОСТ Р 34.12-2015 «Криптографическая защита информации. Блочные шифры»

5. Саломаа А. Криптография с открытым ключом. - М., 1995.

6. Романец Ю.В., Панасенко С.П., Заботин И.А., Петров С.В., Ракитин В.В., Дударев Д.А., Сырчин В.К., Салманова Ш.А. Глава 3. История создания алгоритма ГОСТ 28147-89 и принципы, заложенные в его основу // Фирма

# ПРИЛОЖЕНИЕ

## 3.1. Исходный код