|  |  |  |
| --- | --- | --- |
|  |  |  |



Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

Институт кибернетики   
Базовая кафедра №252 – информационной безопасности

**ОЗНАКОМИТЬЕЛЬНАЯ ПРАКТИКА**

**Тема:** Разработка безопасного аудиокодека с шифрованием трафика.

|  |  |  |
| --- | --- | --- |
| **Студенты учебной группы ККСО-01-21** |  | Ермакова А.В. Савосин Ю.А. |
|  |  |  |
| **Руководитель** |  | Трошков В. Е. |

2022

Содержание

[ВВЕДЕНИЕ 3](#_Toc104641212)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc104641213)

[1.1 Изучение способов передачи аудиоданных в информационнотелекоммуникационных сетях 4](#_Toc104641214)

[1.2 Изучение алгоритмов сжатия данных, в частности, сжатия аудиоинформации 7](#_Toc104641215)

[1.3.Изучение алгоритмов поточного и блочного шифрования данных 11](#_Toc104641216)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 16](#_Toc104641217)

[СПИСОК ЛИТЕРАТУРЫ. 17](#_Toc104641218)

# ВВЕДЕНИЕ

В современном мире крайне важно следить за безопасностью своих данных, особенно в сети. Сейчас существует множество алгоритмов шифрования различных типов данных, будь то изображение, текстовый документ или аудиофайл.

В этой работе мы хотим подробнее изучить как кодируются и декодируются аудиосигналы, какие алгоритмы при этом используются. Как происходит сжатие аудиофайла, как они передаются в сети и многое другое. Но и главная наша цель – это научиться писать простейшие аудиокодеки с внедрением алгоритма RC5.

# 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Изучение способов передачи аудиоданных в информационнотелекоммуникационных сетях

Для начала разберемся с такими понятиями как аналоговый и цифровой сигнал, определим разницу между процессами дискретизация и квантование.

[Аналоговый](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D0%BE%D0%B3%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB) (или континуальный) – это сигнал описывается непрерывной функцией времени, т.е. имеет непрерывную линию с непрерывным множеством возможных значений (рис. 1).

Цифровой сигнал – это сигнал, который можно представить как последовательность определенных цифровых значений. В любой момент времени он может принимать только одно определенное конечное значение (рис. 2).

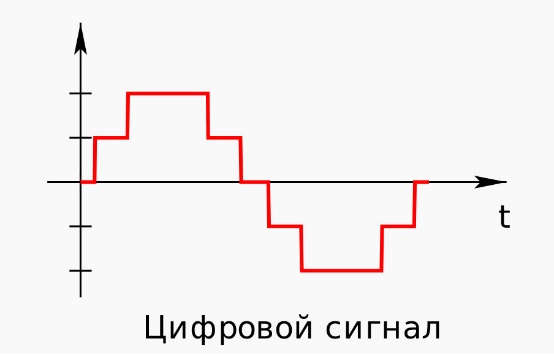


Рисунок 1 Рисунок 2

Квантование – это разбиение диапазона отсчётных значений сигнала на конечное число уровней и округление этих значений до одного из двух ближайших к ним уровней. Квантование приводит сигнал к заданным значениям, то есть округляет сигнал до ближайших к нему уровней (по горизонтали).

Дискретизация разбивает сигнал по временной составляющей (по вертикали).

Не следует путать квантование с дискретизацией (шаг квантования с частотой дискретизации). При дискретизации изменяющаяся во времени величина (сигнал) замеряется с заданной частотой (частотой дискретизации), таким образом, дискретизация разбивает сигнал по временной составляющей (на графике — по горизонтали). Квантование же приводит сигнал к заданным значениям, то есть округляет сигнал до ближайших к нему уровней (на графике — по вертикали).

Эти два процесса создают как бы координатную систему, которая позволяет описывать аудиосигнал определенным значением в любой момент времени.

Цифровым называется сигнал, к которому применены дискретизация и квантование.

Для того чтобы передать аудиосигнал его сначала нужно оцифровать. Существуют различные понятия звукового формата. Формат представления звуковых данных в цифровом виде зависит от способа квантования аналогово–цифровым преобразователем (АЦП).

При оцифровке сигнала количество битов, кодирующих один уровень квантования, называют глубиной квантования (разрядность). Уровни квантования нумеруются и каждому уровню присваивается [двоичный код](https://ru.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4). Чем выше разрядность, тем больше уровней можно представить двоичным кодом.

Стоит отметить, что чем больше глубина квантования и частота дискретизации, тем точнее цифровой сигнал соответствует аналоговому. В случае равномерного квантования глубина квантования определяет динамический диапазон, измеряемый в децибелах.

В телекоммуникации существует два типа передачи данных:

1) Последовательная.

2) Параллельная.

Цифровая последовательная передача — это последовательная отправка [битов](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82) по одному проводу, частоте или оптическому пути. Так как это требует меньшей [обработки сигнала](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB%D0%BE%D0%B2) и меньше вероятность ошибки, чем при параллельной передаче, то скорость передачи данных по каждому отдельному пути может быть быстрее. Этот механизм может использоваться на более дальних расстояниях, потому что легко может быть передана контрольная цифра или [бит чётности](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82_%D1%87%D1%91%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%B8).

В цифровой связи параллельной передачей называется одновременная передача соответствующих элементов сигнала по двум или большему числу путей. Используя множество электрических проводов можно передавать несколько бит одновременно, что позволяет достичь более высоких скоростей передачи, чем при последовательной передаче. Этот метод применяется внутри компьютера, например, во внутренних [шинах данных](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D0%BD%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), а иногда и во внешних устройствах, таких, как [принтеры](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%82%D0%B5%D1%80). Основной проблемой при этом является «перекос», потому что провода при параллельной передаче имеют немного разные свойства, поэтому некоторые биты могут прибыть раньше других, что может повредить сообщение. [Бит чётности](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82_%D1%87%D1%91%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%B8) может способствовать сокращению ошибок.

Тем не менее электрический провод при параллельной передаче данных менее надёжен на больших расстояниях, поскольку передача нарушается с гораздо более высокой вероятностью.

## Изучение алгоритмов сжатия данных, в частности, сжатия аудиоинформации

Методы сжатия данных можно разделить на два типа:

1. Неискажающие (loseless) методы сжатия (называемые также методами сжатия без потерь) гарантируют, что декодированные данные будут в точности совпадать с исходными;

2. Искажающие (lossy) методы сжатия (называемые также методами сжатия с потерями) могут искажать исходные данные, например за счет удаления несущественной части данных, после чего полное восстановление невозможно.

Первый тип сжатия применяют, когда данные важно восстановить после сжатия в неискаженном виде, это важно для текстов, числовых данных и т. п. Полностью обратимое сжатие ничего не удаляет из исходных данных. Сжатие достигается только за счет иного, более экономичного, представления данных.

Второй тип сжатия применяют, в основном, для видео изображений и звука. За счет потерь может быть достигнута более высокая [степень сжатия](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_classification_01.html#ratio). В этом случае потери при сжатии означают несущественное искажение изображения (звука) которые не препятствуют нормальному восприятию, но при сличении оригинала и восстановленной после сжатия копии могут быть замечены.

Существует несколько методов сжатия данных: общего назначения и специальные методы.

Общего назначения (general-purpose), как правило, ориентированы на сжатие текстов, исполняемых программ, объектных модулей и библиотек и т. д., т. е. данных, которые в основном и хранятся в ЭВМ.

Специальные (special) методы сжатия, которые ориентированы на сжатие данных известной природы, например, звука, изображений и т. д. И за счет знания специфических особенностей данных достигают существенно лучшего качества и/или скорости сжатия, чем при использовании методов общего назначения.

Методы сжатия общего назначения являются [неискажающие](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_classification_01.html#loseless_def); [искажающими](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_classification_01.html#lossy_def) могут быть только [специальные](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_classification_01.html#special_def) методы сжатия. Как правило, искажения допустимы только при обработке всевозможных сигналов (звука, изображения, данных с физических датчиков).

Поговорим о сжатии аудиоданных.

Форматы хранения звуковой информации – Layer-1, Layer-2 и Layer-3. Все три формата похожи друг на друга, но используют различные уровни компромисса между сжатием и сложностью.

Уровень Layer-1 - наиболее простой, не требует значительных затрат на сжатие, но и дает незначительную степень сжатия.

MPEG 1 Layer ll является широко используемым форматом. Это типичный пример алгоритма кодирования на основе психоакустического восприятия, который анализирует входящий сигнал и сравнивает его с теоретической моделью для того, чтобы определить, какая частота и информация о временной области могут быть потеряны. Необходимость анализа аудиосигнала дает обязательную задержку, как правило, свыше 30 мс.

Уровень Layer-3 – наиболее трудоемкий и обеспечивает самое лучшее сжатие. В последнее время этот формат завоевал огромную популярность. Его часто называют MP3. Такое название связано с расширением звуковых файлов, хранящихся в этом формате.

Теоретически можно достичь высоких степеней сжатия, но даже при относительно низком сжатии MPEG может серьезно снизить качество звука.

Основанная идея, на которой основаны все методики [сжатия](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_introduction.html#compression) аудио сигнала с [потерями](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_lossy.html#lossy), – пренебрежение тонкими деталями звучания оригинала, лежащие вне пределов которые воспринимает человеческое ухо.

Звуковое сжатие базируется на простом факте – если человек находиться рядом с громко воющей сиреной, то вряд ли он услышит разговор стоящих неподалеку людей. Причем это происходит не оттого, что человек обращает большое внимание на громкий звук, а в большей степени оттого, что человеческое ухо фактически теряет звуки, лежащие в том же диапазоне частот, что и более громкий звук. Этот эффект носит название маскирующего, он изменяется с различием в громкости и частоте звука.

Вторым моментом является деление полосы звуковых частот на подполосы, каждая из которых далее обрабатывается отдельно. Программа кодирования выделяет самые громкие звуки в каждой полосе и использует эту информацию для определения приемлемого уровня шума для этой полосы. Лучшие программы кодирования учитывают также влияние соседних полос. Очень громкий звук в одной полосе может повлиять на маскирующий эффект и на близлежащие полосы.

Еще одним моментом кодирования является использование психоакустической модели, опирающейся на особенности человеческого восприятия звука. [Сжатие](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_introduction.html#compression) с использованием этой модели основано на удалении заведомо неслышимых частот с более тщательным сохранением звуков, хорошо различаемых человеческим ухом. К сожалению, здесь не может быть точных математических формул. Восприятие звука человеком – сложный, до конца не изученный процесс, поэтому выбор методов сжатия выполняется на основе прослушивания и сравнения по-разному сжатых звуков группами экспертов. Зато здесь имеются практически неограниченные возможности в сфере улучшения психоакустических моделей.

Еще одним приемом сжатия является использование совмещенного стерео. Известно, что слуховой аппарат человека может определить направление лишь средних частот – высокие и низкие звучат словно отдельно от источника. Значит, эти фоновые частоты можно кодировать в моносигнал.

На последней стадии сжатия используется алгоритм сжатия [Хаффмана](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_huffman.html#huffman). Этот процесс позволяет улучшить [степень](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_classification_01.html#ratio) сжатия для относительно однородных сигналов, которые плохо сжимаются с помощью описанных выше приемов. На основе описанных идей строятся алгоритмы сжатия, позволяющие достигать [степени](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_classification_01.html#ratio) компрессии 10:1 (или выше) практически без потери в качестве звучания. При кодировании задают требуемый уровень компрессии, а алгоритмы сжатия добиваются требуемого значения уровня [сжатия](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_introduction.html#compression) за счет потери качества. Требуемый уровень сжатия обычно указывают в виде величины потока данных ([bit rate](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_souvid.html" \l "bitrate)), измеряемого в Кбит/сек.

Термин «bit rate» обозначает количество битов передаваемой информации в секунду. Иногда для звуковых файлов этот же параметр называют [степенью](https://mf.grsu.by/UchProc/livak/po/comprsite/theory_classification_01.html#ratio) сжатия файла. Например, говорят, что файл сжат до 128 Кбит/с. В+еличина битрейта напрямую связана с физическим размером звукового файла в расчете на одну секунду звучания.

## 1.3.Изучение алгоритмов поточного и блочного шифрования данных

Алгоритм блочного шифрования данных.

Современный блочный шифр – это шифр с симметричным ключом, разбивающий перед шифрованием открытый текст на n -битовые блоки и далее шифрующий сообщение блоками.

Если сообщение содержит менее n битов, то его дополняют дополнительными данными (чаще нулями) до n -битового размера; если текст имеет больше, чем n бит, то его делят на n -битовые блоки. Чаще всего блочные шифры обрабатывают блоки длиной n = 64, 128, 256, 512 бит.

Блочный шифр можно спроектировать так, чтобы он действовал и как шифр подстановки, и как шифр перестановки:

1) Если шифр спроектирован как шифр подстановки, каждый бит открытого текста может быть заменен на 0 или 1, следовательно, исходный текст и шифротекст могут иметь различное число единиц.

2) Если шифр спроектирован как шифр перестановки, то биты открытого текста только меняются местами.

В любом случае, число возможных n-битовых открытых текстов равно числу шифротекстов и равно 2^n (так как каждый из n битов блока может быть равен 0 или 1).

Стоит отметить, что современные блоковые шифры спроектированы как шифры подстановки, потому что использование только перестановки делает шифр уязвимым к методу полного перебора (из-за сохранения числа единиц или нулей).

К современным блочным шифрам выдвигают следующие требования:

1. Рассеивание информации – распространение влияния одного знака открытого текста на почти все знаки шифротексты. Это затрудняет статистический анализ и не позволяет восстановить ключ по частям;

2. Перемешивание информации для усложнения зависимости между ключом и шифрованным текстом.

Схемы, решающие такие задачи, называют SP-сетями (от первых букв англ. substitution – подстановка и permutation – перестановка).

Основные операторы блоковых шифров:

1. операторы перестановки, называемые P-боксами;

2) операторы подстановки, называемые S -боксами;

3) операция исключающего ИЛИ (XOR);

4) циклический сдвиг;

5) замена;

6) разбиение и объединение блока. Комбинация этих операторов в современных шифрах позволяет рассеять и перемешать информацию.

P-бокс (блок перестановки) подобен традиционному шифру перестановки (переставляемые символы – биты). Возможны 3 типа P - боксов: прямые P-боксы (простая перестановка символов, n входов и n выходов, всего возможно n! отображений); P-боксы расширения; P-боксы сжатия.

Прямой P-бокс обычно задают с помощью таблицы перестановок, в которой номера ячеек задают номера битов входящего блока, а числа записанные в ячейки указывают новые позиции бита.

P-бокс сжатия – это P-бокс с n входами и m выходами, где m < n. Некоторые из информационных входов блокированы и не связаны с выходом. Задаются таблицей перестановки с m ячейками (входы), в которых позиции битов от 1 до n, (биты, которые блокированы в таблице отсутствуют).

P-бокс расширения – это P-бокс с n входами и m выходами, где m > n. Некоторые из входов связаны больше чем с одним выходом.

S -бокс (блок подстановки) – это миниатюрный шифр подстановки. На вход в S -бокс может подаваться n -битовый блок, а на выходе выйти уже m- битовый блок, где не всегда m = n. (говорят бокс с n входами и m выходами).

Операция XOR – побитовое сложение, сложения по модулю 2. Свойства операции XOR:

1. Замкнутость: если x y, – n -битовые слова, то x ⊕ y = z, где z – n -битовое слово.

2. Ассоциативность.

3. Коммутативность.

4. Существование нулевого элемента с условием x ⊕ x = (00...0) .

5. Существование инверсии – операция XORслова с самим собой дает нулевой элемент x ⊕ x = (00...0).

Циклический правый сдвиг сдвигает каждый бит в n -битовом слове на k позиций вправо; самые правые k бит справа удаляются и становятся крайними левыми. Аналогично работает левый сдвиг.

Свойства циклического сдвига:

1. Cмещение по модулю n. Другими словами, если k = 0 или k = n, никакого смещения не происходит. Если k > n, то входная информация сдвигается на k mod n бит.

2. Если смещение делается неоднократно, то вновь может появиться исходное n -битовое слово (сдвиг является групповой операцией).

Замена – частный случай операции циклического сдвига на k = n/2 битов (здесь n – четное). Сдвиг влево на n / 2 – то же самое, что сдвиг на n / 2 вправо, следовательно, операция является обратимой.

Разбиение разделяет n -битовое слово пополам, создавая два слова равной длины. Объединение связывает два слова равной длины, чтобы создать n -битовое слово. Эти две операции инверсны друг другу: если одна используется для шифрования, то другая – для дешифрования.

Алгоритм поточного шифрования данных.

Поточный (потоковый) шифр – это [симметричный шифр](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80), в котором каждый символ открытого текста преобразуется в символ шифрованного текста в зависимости не только от используемого ключа, но и от его расположения в потоке открытого текста. Поточный шифр реализует другой подход к симметричному шифрованию, нежели [блочные шифры](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).

Простейшая реализация блочного шифрования такова: Генератор гаммы выдаёт ключевой поток (гамму): k1, k2, …, kL {\displaystyle k\_{1},k\_{2},k\_{3},\dots ,k\_{L}}. Обозначим поток битов открытого текста m1, m2, …, mL{\displaystyle m\_{1},m\_{2},m\_{3},\dots ,m\_{L}}. Тогда поток битов шифротекста получается с помощью применения операции исключающего или (XOR): c1, c2, …, cL, где ci = mi ⊕ ki.

Расшифрование производится операцией XOR между той же самой гаммой и зашифрованным текстом mi = ci ⊕ ki.

Очевидно, что если последовательность битов гаммы не имеет периода и выбирается случайно, то взломать шифр невозможно. Но у данного режима шифрования есть и отрицательные особенности. Так ключи, сравнимые по длине с передаваемыми сообщениями, трудно использовать на практике. Поэтому обычно применяют ключ меньшей длины (например, 128 бит).

С помощью него генерируется псевдослучайная гаммирующая последовательность, удовлетворяющая следующим принципам (постулаты Голомба):

1) Количество «1» в каждом периоде должно отличаться от количества «0» не более, чем на единицу.

2) В каждом периоде половина серий (из одинаковых символов) должна иметь длину один, одна четверть должна иметь длину два, одна восьмая должна иметь длину три и т.д. Более того, для каждой из этих длин должно быть одинаковое количество серий из «1» и «0».

3) Предположим, у нас есть две копии одной и той же последовательности периода p, сдвинутые относительно друг друга на некоторое значение d. Тогда для каждого d (0<=d<=p-l), мы можем подсчитать количество согласованностей между этими двумя последовательностями Ad, и количество несогласованностей Dd. Коэффициент автокорреляции для каждого d определяется соотношением (Ad - Dd)/p и эта функция автокорреляции принимает различные значения по мере того, как d проходит все допустимые значения. Тогда для любой последовательности, удовлетворяющей правилу 3, автокорреляционная функция (АКФ) должна принимать лишь два значения.

Естественно, псевдослучайность гаммы может быть использована при атаке на потоковый шифр.

Допустим, что при передаче по каналу связи произошло искажение одного знака шифротекста. В таком случае все знаки, принятые без искажения, будут расшифрованы правильно. Но если при шифровании один знак будет потерян, то неправильно будет расшифрован весь текст после него. Чтобы этого не допустить нужно решить проблему синхронизации шифрования и расшифрования текста. По способу решения этой проблемы шифросистемы подразделяются на синхронные и самосинхронизирующиеся.

Синхронные потоковые шифры (СПШ) – шифры, в которых поток ключей генерируется независимо от открытого текста и шифротекста. При шифровании генератор потока ключей выдаёт биты потока ключей, которые идентичны битам потока ключей при дешифровании. Потеря знака шифротекста приведёт к нарушению синхронизации между этими двумя генераторами и невозможности расшифрования оставшейся части сообщения. Очевидно, что в этой ситуации отправитель и получатель должны повторно синхронизоваться для продолжения работы.

Плюсы СПШ:

1) Отсутствие эффекта распространения ошибок, только искажённый бит будет расшифрован неверно.

2) Они предохраняют от любых вставок и удалений шифротекста, так как они приведут к потере синхронизации и будут обнаружены.

Минусы СПШ:

1) Они уязвимы для изменения отдельных бит шифрованного текста. Если злоумышленнику известен открытый текст, он может изменить эти биты так, чтобы они расшифровывались, как ему надо.

Самосинхронизирующиеся потоковые шифры (асинхронные потоковые шифры (АПШ)) – шифры, в которых ключевой поток создается функцией ключа и фиксированного числа знаков шифротекста.

Итак, внутреннее состояние генератора потока ключей является функцией предыдущих N битов шифротекста. Поэтому расшифрующий генератор потока ключей, приняв N битов, автоматически синхронизируется с шифрующим генератором.

Реализация этого режима происходит следующим образом: каждое сообщение начинается случайным заголовком длиной N битов; заголовок шифруется, передаётся и расшифровывается; расшифровка является неправильной, зато после этих N бит оба генератора будут синхронизированы.

Плюсы АПШ:

1) Размешивание статистики открытого текста. Так как каждый знак открытого текста влияет на следующий шифротекст, статистические свойства открытого текста распространяются на весь шифротекст. Следовательно, АПШ может быть более устойчивым к атакам на основе избыточности открытого текста, чем СПШ.

Минусы АПШ:

1) Распространение ошибки (каждому неправильному биту шифротекста соответствуют N ошибок в открытом тексте).

2) АПШ чувствительны к вскрытию повторной передачей.

# 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

Ознакомиться с нашим кодеком вы можете на GitHab’e. Ссылка прикреплена вместе с документом.

# СПИСОК ЛИТЕРАТУРЫ.

1. [HerrFelix](https://habr.com/ru/users/HerrFelix/), «Цифровое представление аналогового аудиосигнала. Краткий ликбез» // Статья на Habr. [электронный ресурс], URL: <https://habr.com/ru/post/503786/>

2. «Алгоритм работы RC5» // Википедия. [электронный ресурс], URL: <https://ru.wikipedia.org/wiki>

3. «Алгоритмы сжатия» // Статься в Интернете [электронный ресурс] URL: <https://mf.grsu.by/UchProc/livak/po/comprsite/theory_classification_01.html>

4. Сушко С.А., «Блочные шифры» // Лекция. [электронный ресурс] URL: <https://bit.nmu.org.ua/ua/student/metod/cryptology>

5. «Потоковый шифр» // Статься в Интернете [электронный ресурс] URL: <https://mf.grsu.by/UchProc/livak/po/comprsite/theory_classification_01.html>