Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА — Российский технологический университет»

базовая кафедра № 252

Разработка безопасного аудиокодека с шифрованием трафика

Реферат студентов 1 курса института искусственного интеллекта
Коровяковского Степана и Урвачева Романа

Москва 2022

Оглавление

Глава 1

Изучение способов передачи аудиоданных в информационно-телекоммуникационных сетях

1.1 Телекоммуникационные технологии

Каждому поколению свойственно разрабатывать новые технические средства, совершенствовать систему учета, обработки, передачи и хранения данных. Первыми телекоммуникационными средствами признан телеграф, телефон, телетайп, радиоприемник. Середина XIX столетия отмечена массовым использованием спутниковой связи, вычислительной техники, компьютерной сети. В результате это положительно отразилось на развитии новых телекоммуникационных технологий.

Современный мир невозможен без телекоммуникационных технологий, которые стирают государственные границы и расстояние между людьми, делают доступной мобильную и видеосвязь и позволяют решать множество задач в сфере управления, образования, коммерции. Каждый человек сталкивается с ними ежедневно, деля телефонные звонки, проверяя почту или покупая товары в интернет-магазинах.

1.1.1 Определение и понятие телекоммуникационных технологий

Общее понятие информационных и коммуникационных технологий включает в себя совокупность методов, процессов и устройств, позволяющих получать, собирать, накапливать, хранить, обрабатывать и передавать информацию, закодированную в цифровом виде или существующую в аналоговом виде.

В более узком смысле под телекоммуникационными технологиями понимается совокупность программных и аппаратных средств, позволяющих устанавливать связь без использования проводов и передавать пакеты информации.

1.1.2 Виды телекоммуникационных технологий

Телекоммуникационные технологии могут быть рассмотрены как сервисы, предоставляемые провайдерами различного уровня. По этому принципу можно выделить следующие виды телекоммуникационных технологий:

- телефонная связь, современная телефонная связь позволяет легко переключаться с аналогового стандарта на цифровой, подключать к интернет городские телефоны и соединять в одну сеть аналоговые и мобильные устройства;
- радиосвязь, которая сегодня превратилась в сотовую связь, телефон, перемещаясь в пределах сети, оказывается в зоне действия различных передающих устройств;
- спутниковая связь, которая используется провайдерами для создания систем мобильной связи и для государственных систем связи;
- интернет – наиболее распространенный вид телекоммуникационных технологий, при которых подключение к сети может осуществляться как проводным, так и беспроводным способом.

1.1.3 Основные типы информационно-телекоммуникационных сетей

Телекоммуникационные технологии, используемые в интернете, сейчас переживают этап бурного развития и роста. С каждым днём создаётся всо больше и больше новых сетей различных типов, среди которых основными являются:

- локальные сети компаний или учреждений (Local Area Network - LAN), связь между компьютерами в них осуществляется и проводным и беспроводным способом, количество пользователей этих сетей ограничено. Локальные сети могут быть корпоративными, в некоторых странах создаются и городские локальные сети;
- глобальные сети (Wide Area Network – WAN) представляют совокупность большого количества узлов-компьютеров, расположенных в разных странах мира и связанных между собой каналами оптово-волоконной связи. К этим сетям, представляющим услуги провайдеров, подключаются локальные сети.

1.1.4 Технические и программные средства телекоммуникационных технологий

Работоспособность интернета основана на использовании сетевых узлов и каналов связи. К узлам относятся как отдельные компьютеры, так и хостинги, предоставляющие IP-адреса и доменные имена. Каналы связи делятся на 4 типа:

- аналоговые телефонные сети;

- провода, по которым передается электричество;
- оптоволоконные каналы связи;
- беспроводные каналы связи, модемные или спутниковые.

К телекоммуникационным каналам связи относятся, в основном, третий и четвертый типы.

Среди коммуникаций, используемых для организации связи, можно отдельно отметить программы, обеспечивающие работу телекоммуникационного оборудования такого, как:

- IP-АТС;
- маршрутизаторы;
- компьютеры.

1.1.5 Основные задачи сетевых телекоммуникационных технологий

Различные сетевые телекоммуникационные технологии позволяют решать такие задачи, как:

- передачу информации в необходимых форматах;
- выстраивание коммуникаций;
- обеспечение взаимодействия различных участников сети.

Среди новых технологий особое место занимают программы, позволяющие работать в режиме нетворкинга, объединение CRM-систем с возможностями социальных сетей и многое другое.

Создание корпоративных сетей как офисных, компьютерных, так и телефонных, также попадает в область сетевых технологий, призванных обеспечить синергию за счет эффективной коммуникации пользователей.

Спектр возможностей использования телекоммуникационных технологий расширяется с каждым днем. Сложно сказать, что именно будет предложено завтра в этой области, чтобы сделать связь доступнее, а производственные процессы – проще.

1.2 Описание способов передачи аудиоданных

1.2.1 Передача аудиоданных по телефонным каналам связи

Взаимное проникновение вычислительной техники и технических средств связи оказало серьезное влияние как на структуру компьютеров, так и на структуру каналов связи.

Средства связи, предназначенные для передачи информации между людьми, имеют длительную историю, развитую структуру (в мировом масштабе), мощную научную и технологическую базу и, начиная с 60-х годов, стали использоваться для передачи данных, т.е. для передачи информации между техническими средствами вычислительной техники, что потребовало включения в каналы связи дополнительных технических устройств.

В настоящее время для распределенных вычислительных систем наиболее широко используются телефонные каналы.

На рис. 1.2.1.1 представлена упрощенная схема линии аналоговой междугородней телефонной связи.

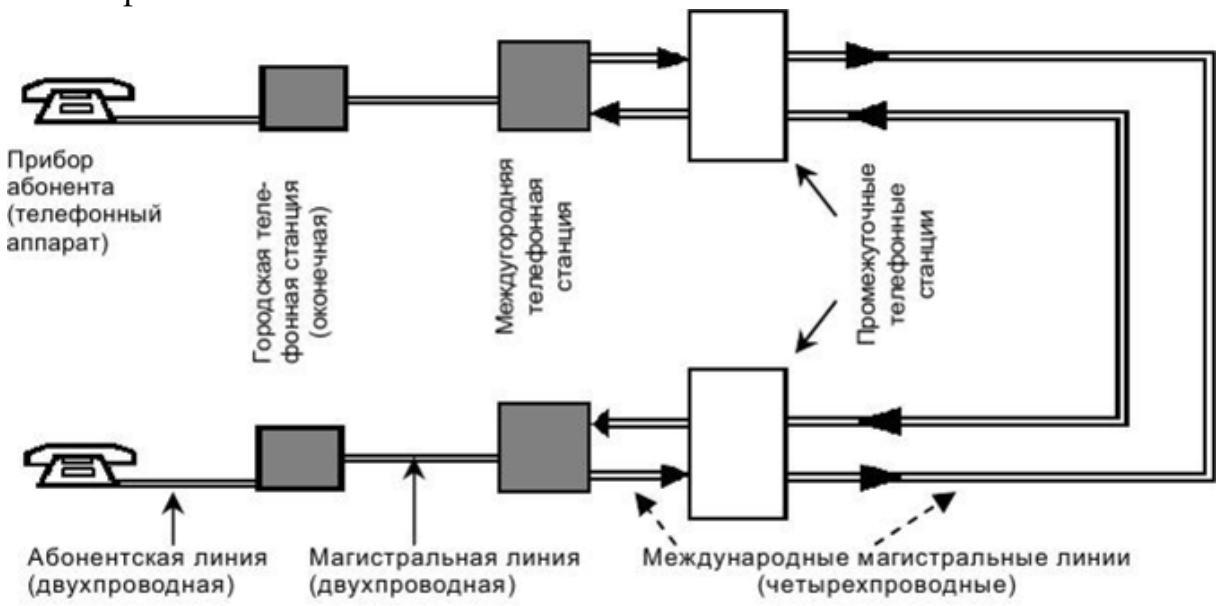

Рис. 1.2.1.1 Схема междугородней телефонной связи

На участке от телефонного аппарата до местной АТС происходит передача в первичной полосе частот « 200 - 3100 Гц (полоса частот человеческого голоса). При этом от каждого аппарата до АТС проводится двухпроводная электрическая линия для передачи этого сигнала, в дальнейшем происходит преобразование его в иную форму с целью уплотнения передачи. В каждом из последующих каналов идет очень большое количество передач. Существует два типа уплотнения: частотное и временное. В традиционных линиях связи, как правило, используется частотное уплотнение.

Описание типов каналов связи

В зависимости от типа передачи различают аналоговые (традиционно используемые, имеющие длительную историю развития) и цифровые каналы (систем ИКМ, ШОЫ и др.), являющиеся битовым трактом с цифровым импульсным сигналом на выходе и входе канала. Цифровые каналы отличаются рядом преимуществ перед аналоговыми, поэтому вновь создаваемые системы передачи данных стараются строить на основе цифровых каналов. Следует отметить, что цифровые каналы весьма успешно применяются не только для передачи данных, но и в средствах бытовой связи (звук, изображение и г.д.), при этом аналоговые сигналы

кодируются в цифровые перед передачей в канал.

Термины «аналоговый» и «цифровой» соответствуют непрерывным и дискретным процессам и используются при обсуждении коммуникационных систем в различных контекстах - данных, сигналов и передачи.

Аналоговые данные представляются физической величиной, которая может изменяться в непрерывном диапазоне значений. Величина прямо пропорциональна данным или является их функцией.

Цифровые данные принимают дискретные значения - текст, целые числа, двоичные данные.

Аналоговый сигнал - непрерывно изменяющаяся электромагнитная волна, распространяющаяся в различных средах.

Цифровой сигнал - дискретный (разрывной) сигнал, такой, как последовательность импульсов напряжения.

Возможны четыре вида передачи данных:

1) цифровые данные - цифровой сигнал, используется наиболее простое оборудование;

2) аналоговые данные - цифровой сигнал, необходимо преобразование аналоговых данных в цифровую форму, что позволяет использовать современное (высокоэффективное) оборудование передачи данных;

3) цифровые данные - аналоговый сигнал, необходимость преобразования связана с тем, что через некоторые среды (оптоволокно, беспроводные среды) может распространяться только аналоговый сигнал;

4) аналоговые данные - аналоговый сигнал, традиционная передача, аналоговые данные легко преобразуются в аналоговый сигнал.

Основные преимущества цифровой передачи данных

Среди преимуществ цифровой передачи необходимо отметить следующие.

- Быстрое развитие цифровых систем и уменьшение цены и размеров оборудования, цены и размеры аналогового оборудования остаются на прежнем уровне. Обслуживание цифровых систем намного дешевле аналоговых.

- Использование повторителей (в цифровых системах) вместо аналоговых усилителей позволяет передавать данные на большие расстояния по менее качественным линиям (нет накопления шумов) - сохранение целостности данных.

- Большая пропускная способность дает возможность более полно использовать пропускную способность оптоволокна и спутниковых средств связи. Временное разделение оказывается более эффективным, чем частотное.

- Используется интеграция, когда при обработке аналоговой и цифровой информации по цифровым технологиям все сигналы имеют одинаковую форму (вид). Эго позволяет сэкономить на оборудовании и трудозатратах при интеграции: голос, видео, цифровые данные.

Модемы

Термин «модем» применяется в настоящие время (в связи с распространением цифровых каналов) достаточно широко, при этом необязательно подразумевается какая-либо модуляция, а просто называются определенные операции преобразования сигналов, поступающих от ЭТЕ для их дальнейшей передачи по используемому каналу.

Существует очень много разновидностей модемов, отличающихся:

- по конструкции - внутренние (вставляемые в разъемы компьютера) и внешние, портативные, групповые и т.н.;
- по методу передачи - асинхронные, синхронные, синхронноасинхронные.

Асинхронный метод передачи (или стартстопный) - посимвольный режим передачи с контролем начала и конца символа, имеет низкую скорость и малую эффективность. Синхронный метод передачи осуществляет объединение большого количества символов или байт в отдельные блоки - кадры, которые передаются без задержек между восьмибитными элементами.

Режимы работы в зависимости от направления передачи

Очень важной характеристикой канала передачи являются режимы его работы в зависимости от направления возможной передачи данных:

- симплексный - передача осуществляется по линии связи только в одном направлении;
- полудуплексный - передача ведется в обоих направлениях, но попеременно во времени (технология Ethernet);
- дуплексный - передача ведется одновременно в двух направлениях.

Дуплексный режим - наиболее универсальный и производительный. Самым простым вариантом организации дуплексного режима является использование двух независимых функциональных каналов (двух пар проводников или двух световодов) в кабеле, каждый из которых работает в симплексном режиме, т.е. передает данные в одном направлении. Такая организация дуплексного режима применяется во многих сетевых технологиях (Fast Ethernet, ATM и т.п.).

1.2.2 Передача аудиоданных по радиосвязи

Радиосвязь – быстрый и относительно надежный способ передачи данных на большие расстояния. При этом нет необходимости в использовании физического носителя, например проводов.

Свойства волн разной длины напрямую влияют на их применение для обеспечения радиосвязи. Кроме того, на качество передачи информации с их помощью влияют следующие факторы:

- высота приемной и передающей антенн;
- рельеф поверхности;
- солнечная активность, метеоусловия, время суток.

Процесс приема-передачи информации

Процесс приема-передачи информации с помощью радиоволн состоит из следующих основных этапов:

1) формирование сигнала;

2) выделение несущей частоты;

3) связывание передаваемой информации с несущей частотой (модуляция);

4) трансформация сигнала в дискретный вид, его кодирование (для цифровых систем);

5) передача в радиоэфир с помощью антенны;

6) прием сигнала;

7) декодировка и демодуляция;

8) преобразование сигнала в форму понятную абоненту.

Оборудование для осуществления передачи данных

Чтобы реализовать обмен информации необходимо чтобы у принимающей и передающей стороны в наличии было следующее оборудование:

- передатчик;
- антенна;
- ретрансляционное устройство – позволяет увеличить дальность передачи сигнала;
- принимающее устройство;

- оборудование модуляции-демодуляции, сжатия, оцифровки и кодирования;
- фильтры помех, усилители.

Способы передачи данных по радиосвязи

Применяется несколько способов радиосвязи, для каждого из которых используется специфическое оборудование. Три наиболее распространенных вида:

- сотовая связь;
- радиорелейная связь;
- спутниковая связь.

Сферы применения радиосвязи

Возможность практически мгновенной передачи информации на любые расстояния создает широкие возможности использования во всех сферах деятельности человека. Радиосвязь успешно применяется в следующих отраслях:

- телевизионное и радиовещание;
- качественная связь по безопасным линиям востребована в военной отрасли. Позволяет осуществлять управление и координацию боевых подразделений;
- в области транспорта – обеспечивается постоянная связь с поездами, морскими и речными судами, самолетами, грузовыми и легковыми автомобилям (полиция, скорая помощь, такси, курьерские службы);
- организация диспетчерских служб;
- обеспечение различных видов коммуникации: спутниковая, мобильная связь;
- беспроводное подключение к сети Интернет.

Также широкие возможности коммуникации являются неотъемлемым инструментом практически любого современного бизнеса. При помощи беспроводной связи можно успешно решать вопросы управления удаленными объектами.

1.2.3 Передача аудиоданных через системы спутниковой связи

Системы спутниковой связи (ССС) широко используются во многих регионах мира и стали неотъемлемой частью инфраструктуры телекоммуникаций большинства стран. Не только промышленно развитые страны с разнообразными современными сетями телекоммуникаций, но все чаще и развивающиеся страны успешно внедряют ССС.

Новые спутниковые приложения обеспечивают быстрое создание новых широковещательных служб и частных сетей.

Устройство спутниковой связи

Спутник - устройство связи, которое принимает сигналы от земной станции (ЗС), усиливает и транслирует в широковещательном режиме одновременно на все ЗС, находящиеся в зоне видимости спутника. Спутник не инициирует и не терминирует никакой пользовательской информации за исключением сигналов контроля и коррекции возникающих технических проблем и сигналов его позиционирования. Спутниковая передача начинается в некоторой ЗС, проходит через спутник, и заканчивается в одной или большем количестве ЗС.

ССС состоит из трех базисных частей: космического сегмента, сигнальной части и наземного сегмента (рис. 1.2.3.1). Космический сегмент охватывает вопросы проектирования спутника, расчета орбиты и запуска спутника. Сигнальная часть включает вопросы используемого спектра частоты, влияния расстояния на организацию и поддержание связи, источники интерференции сигнала, схем модуляции и протоколов передачи. Наземный сегмент включает размещение и конструкцию ЗС, типы антенн, используемых для различных приложений, схемы мультиплексирования, обеспечивающие эффективный доступ к каналам спутника.

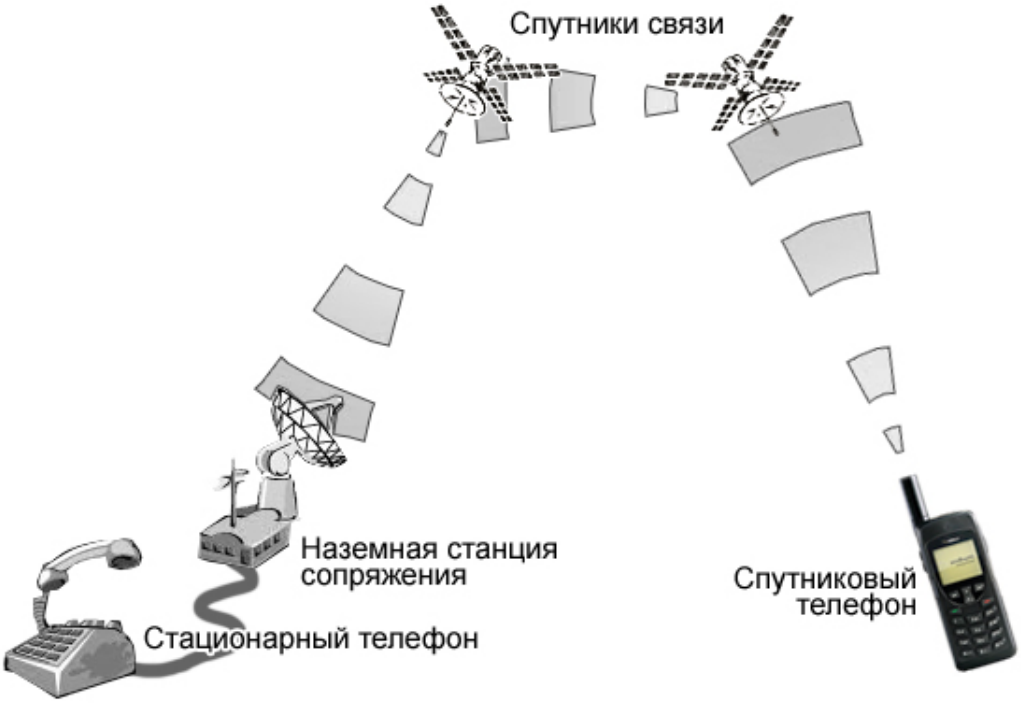

Рис. 1.2.3.1

На рис. 1.2.3.1: космический сегмент - спутники связи, сигнальная часть - наземная станция сопряжения, наземный сегмент - спутниковый телефон.

Преимущества и ограничения систем спутниковой связи

Системы спутниковой связи имеют уникальные особенности, отличающие их от других систем связи. Некоторые особенности обеспечивают преимущества, делающие спутниковую связь привлекательной для ряда приложений. Другие создают ограничения, которые неприемлемы при реализации некоторых прикладных задач.

ССС имеет ряд преимуществ:

- Устойчивые издержки. Стоимость передачи через спутник по одному соединению не зависит от расстояния между передающей и принимающей ЗС. Более того, все спутниковые сигналы - широковещательные. Стоимость спутниковой передачи, следовательно, остается неизменной независимо от числа принимающих ЗС.

- Широкая полоса пропускания.

- Малая вероятность ошибки. В связи с тем, что при цифровой спутниковой передаче побитовые ошибки весьма случайны, применяются эффективные и надежные статистические схемы их обнаружения и исправления.

Выделим также ряд ограничений в использовании ССС:

- Значительная задержка. Большое расстояние от ЗС до спутника на геосинхронной орбите приводит к задержке распространения, длиной почти в четверть секунды. Эта задержка вполне ощутима при телефонном соединении и делает чрезвычайно неэффективным использование спутниковых каналов при неадаптированной для ССС передаче данных.

- Размеры ЗС. Крайне слабый на некоторых частотах спутниковый сигнал, доходящий до ЗС (особенно для спутников старых поколений), заставляет увеличивать диаметр антенны ЗС, усложняя тем самым процедуру размещения станции.

- Защита от несанкционированного доступа к информации. Широковещание позволяет любой ЗС, настроенной на соответствующую частоту, принимать транслируемую спутником информацию. Лишь шифрование сигналов, зачастую достаточно сложное, обеспечивает защиту информации от несанкционированного доступа.

- Интерференция. Спутниковые сигналы, действующие в Ku- или Ka-полосах частот (о них ниже), крайне чувствительны к плохой погоде. Спутниковые сети, действующие в C-полосе частот, восприимчивы к микроволновым сигналам. Интерференция вследствие плохой погоды ухудшает эффективность передачи в Ku- и Ka-полосах на период от нескольких минут до нескольких часов. Интерференция в C-полосе ограничивает развертывание ЗС в районах проживания с высокой концентрацией жителей.

Влияние упомянутых преимуществ и ограничений на выбор спутниковых систем для частных сетей довольно значительно. Решение об использовании ССС, а не распределенных наземных сетей, всякий раз необходимо экономически обосновать.

Роль наземного сегмента

Технологическое развитие привело к значительному уменьшению размеров земных сегментов. На начальном этапе спутник не превышал нескольких сотен килограммов, а ЗС представляли собой гигантские сооружения с антеннами более 30 м в диаметре. Современные спутники весят несколько тонн, а антенны, зачастую не превышающие 1 м в диаметре, могут быть установлены в самых разнообразных местах. Тенденция уменьшения размеров ЗС вместе с упрощением установки оборудования приводит к снижению его стоимости. На сегодняшний день стоимость ЗС является, пожалуй, главной характеристикой, определяющей широкое распространение ССС. Преимущество спутниковой связи основано на обслуживании географически удаленных пользователей без дополнительных расходов на промежуточное хранение и коммутацию. Любые факторы, понижающие стоимость установки новой ЗС, однозначно содействуют развитию приложений, ориентированных на использование ССС.

Последние достижения технологии в области спутниковой связи говорят о больших потенциальных возможностях ССС в расширении пропускной способности каналов передачи, разработке и внедрении новых служб связи. Будущее ССС за широкополосными широковещательными приложениями и спутниковыми системами подвижной связи.

1.2.4 Передача аудиоданных через Интернет

Аудио через IP — распространение цифрового аудио по IP-сети, такой как Интернет. Все чаще используется для обеспечения высококачественной передачи звука на большие расстояния.

IP - телефония

IP-телефония — телефонная связь по протоколу IP. Под IP-телефонией подразумевается набор коммуникационных протоколов, технологий и методов, обеспечивающих традиционные для телефонии набор номера, дозвон и двустороннее голосовое общение, а также видеообщение по сети Интернет или любым другим IP-сетям. Сигнал по каналу связи передаётся в цифровом виде и, как правило, перед передачей преобразовывается, чтобы удалить избыточность информации и снизить нагрузку на сеть передачи данных.

Самая главная особенность рассматриваемой связи — это то, что передача информации происходит не по специально выделенным телефонным линиям, а через компьютерную сеть. Естественно, этот нюанс диктует основные принципы работы технологии.

Полученный сигнал изначально необходимо оцифровать и перевести в форму, соответствующую требованиям протокола TCP-IP. Как правило, эта обязанность возлагается на специальные инструменты — VoIP-шлюзы или, в случае с АТС,

DSP-процессор. Затем пакет передается абоненту, где данные переживают обратную трансформацию в понятный человеку вид (текст, картинка или звук).

Такой подход к организации связи позволяет получить сразу несколько преимуществ, ощутимых не только коммерческим клиентам, а и обычным пользователям:

- IP-телефония существенно снижает расходы на связь;
- Качество передаваемой информации не страдает вне зависимости от расстояния;
- Вариативность допускает легкую комбинацию несколько каналов;
- Программное управление заметно расширяет функционал;
- Оцифрованные данные можно закодировать.

К слову, одно из преимуществ рассматриваемого сервиса — это использование не физической, а виртуальной АТС.

АТС или автоматическая телефонная станция — система устройств, обеспечивающая автоматическое (без участия оператора или телефонисток) соединение и поддержание телефонной связи между абонентами этой АТС.

1.2.5 Интернет-радио

Ещё одной разновидностью передачи аудиоданных по сети Интернет является интернет-радио.

Интернет-радио или веб-радио — группа технологий передачи потоковых аудиоданных через сеть Интернет для осуществления широковещательных передач. Также, в качестве термина интернет-радио или веб-радио может пониматься радиостанция, использующая для вещания технологию потокового вещания в глобальной сети Интернет.

Принцип работы

В технологической основе системы лежит три элемента:

- Станция — генерирует аудиопоток (либо из списка звуковых файлов, либо прямой оцифровкой с аудиокарты, либо копируя существующий в сети поток) и направляет его серверу. (Станция потребляет минимум трафика, потому что создаёт один поток)
- Сервер (повторитель потока) — принимает аудиопоток от станции и перенаправляет его копии всем подключённым к серверу клиентам, по сути является репликатором данных. (Трафик сервера пропорционален количеству слушателей + 1)

- Клиент — принимает аудиопоток от сервера и преобразует его в аудиосигнал, который и слышит слушатель интернет-радиостанции. Можно организовывать каскадные системы радиовещания, используя в качестве клиента повторитель потока (клиент, как и станция, потребляет минимум трафика; трафик клиента-сервера каскадной системы зависит от количества слушателей такого клиента.)

В качестве станции могут выступать обычная программа-аудиоплеер со специальным плагином-кодеком или специализированная программа (например — ICes, EzStream, SAM Broadcaster, RadioShure), а также аппаратное устройство, преобразующее аналоговый аудиопоток в цифровой.

Существует большое количество серверов интернет-вещания. Широко распространён сервер Shoutcast компании Nullsoft, разработанный специально для своего проигрывателя Winamp. Совместимый с Shoutcast сервер Icecast обладает гораздо большей функциональностью, распространяется свободно (на условиях GNU GPL) и бесплатно. В отличие от Shoutcast, Icecast способен передавать несколько аудиопотоков и требует меньше ресурсов на аудиопоток, чаще обновляется, поддерживает UTF-теги и разные форматы аудио, но он намного сложнее в настройке. Также сервера могут различаться по форматам аудиоданных, например: MP3, Ogg/Vorbis, RealAudio.

В качестве клиента можно использовать любой медиаплеер, поддерживающий потоковое аудио и способный декодировать формат, в котором вещает радио.

На рисунке 1.2.5.1 представлена упрощенная схема интернет-радио.

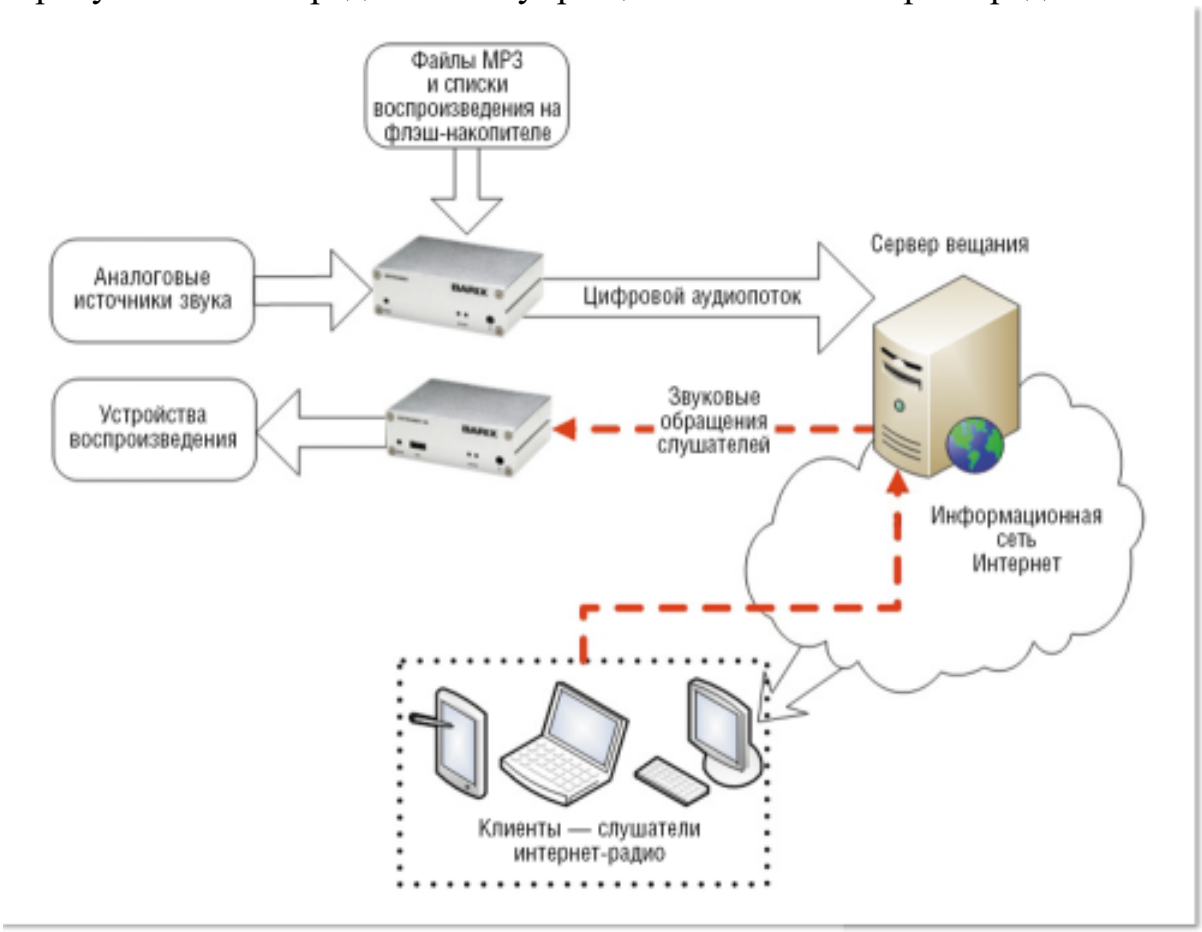

Рис. 1.2.5.1 Схема работы интернет-радио

Глава 2

Изучение алгоритмов сжатия данных, в частности, сжатия аудиоинформации.

2.1 Определение

Сжатие данных — алгоритмическое обратимое преобразование данных, производимое с целью уменьшения занимаемого ими объёма. Применяется для более рационального использования устройств хранения и передачи данных.

Сжатие основано на устранении избыточности, содержащейся в исходных данных. Простейшим примером избыточности является повторение в тексте фрагментов (например, слов естественного или машинного языка). Подобная избыточность обычно устраняется заменой повторяющейся последовательности ссылкой на уже закодированный фрагмент с указанием его длины. Другой вид избыточности связан с тем, что некоторые значения в сжимаемых данных встречаются чаще других. Сокращение объёма данных достигается за счёт замены часто встречающихся данных короткими кодовыми словами, а редких — длинными (энтропийное кодирование). Сжатие данных, не обладающих свойством избыточности (например, случайный сигнал или белый шум, зашифрованные сообщения), принципиально невозможно без потерь.

2.2 Принципы сжатия данных

В основе любого способа сжатия лежит модель источника данных, или, точнее, модель избыточности. Иными словами, для сжатия данных используются некоторые априорные сведения о том, какого рода данные сжимаются. Не обладая такими сведениями об источнике, невозможно сделать никаких предположений о преобразовании, которое позволило бы уменьшить объём сообщения. Модель избыточности может быть статической, неизменной для всего сжимаемого сообщения, либо строиться или параметризоваться на этапе сжатия (и восстановления). Методы, позволяющие на основе входных данных изменять модель избыточности

информации, называются адаптивными. Неадаптивными являются обычно узкоспециализированные алгоритмы, применяемые для работы с данными, обладающими хорошо определёнными и неизменными характеристиками. Подавляющая часть достаточно универсальных алгоритмов является в той или иной мере адаптивной.

Все методы сжатия данных делятся на два основных класса:

- Сжатие без потерь
- Сжатие с потерями

При использовании сжатия без потерь возможно полное восстановление исходных данных, сжатие с потерями позволяет восстановить данные с искажениями, обычно несущественными с точки зрения дальнейшего использования восстановленных данных. Сжатие без потерь обычно используется для передачи и хранения текстовых данных, компьютерных программ, реже — для сокращения объёма аудио- и видеоданных, цифровых фотографий и т. п., в случаях, когда искажения недопустимы или нежелательны. Сжатие с потерями, обладающее значительно большей, чем сжатие без потерь, эффективностью, обычно применяется для сокращения объёма аудио- и видеоданных и цифровых фотографий в тех случаях, когда такое сокращение является приоритетным, а полное соответствие исходных и восстановленных данных не требуется.

2.3 Основные характеристики алгоритмов сжатия

2.3.1 Коэффициент сжатия

Коэффициент сжатия — основная характеристика алгоритма сжатия. Она определяется как отношение объёма исходных несжатых данных к объёму сжатых данных, то есть: $k = \frac{S_o}{S_c}$, где k - коэффициент сжатия, S_o - объём исходных данных, а S_c - объём сжатых. Таким образом, чем выше коэффициент сжатия, тем алгоритм эффективнее. Следует отметить:

- если k = 1, то алгоритм не производит сжатия, то есть выходное сообщение оказывается по объёму равным входному;
- если k < 1, то алгоритм порождает сообщение большего размера, нежели несжатое, то есть, совершает «вредную» работу.

2.3.2 Допустимость потерь

Основным критерием различия между алгоритмами сжатия является описанное выше наличие или отсутствие потерь. В общем случае алгоритмы сжатия без потерь универсальны в том смысле, что их применение безусловно возможно для данных любого типа, в то время как возможность применения сжатия с потерями должна быть обоснована. Для некоторых типов данных искажения не допустимы в принципе. В их числе:

- символические данные, изменение которых неминуемо приводит к изменению их семантики: программы и их исходные тексты, двоичные массивы и т.п.;

- жизненно важные данные, изменения в которых могут привести к критическим ошибкам: например, получаемые с медицинской измерительной аппаратуры или контрольных приборов летательных, космических аппаратов и т.п.;

- многократно подвергаемые сжатию и восстановлению промежуточные данные при многоэтапной обработке графических, звуковых и видеоданных.

2.3.3 Системные требования алгоритмов

Различные алгоритмы могут требовать различного количества ресурсов вычислительной системы, на которых они реализованы:

- оперативной памяти (под промежуточные данные);

- постоянной памяти (под код программы и константы);

- процессорного времени.

В целом, эти требования зависят от сложности и «интеллектуальности» алгоритма. Общая тенденция такова: чем эффективнее и универсальнее алгоритм, тем большие требования к вычислительным ресурсам он предъявляет. Тем не менее, в специфических случаях простые и компактные алгоритмы могут работать не хуже сложных и универсальных. Системные требования определяют их потребительские качества: чем менее требователен алгоритм, тем на более простой, а следовательно, компактной, надёжной и дешёвой системе он может быть реализован.

Так как алгоритмы сжатия и восстановления работают в паре, имеет значение соотношение системных требований к ним. Нередко можно, усложнив один алгоритм, значительно упростить другой. Таким образом, возможны три варианта:

Алгоритм сжатия требует больших вычислительных ресурсов, нежели алгоритм восстановления.

Это наиболее распространённое соотношение, характерное для случаев, когда однократно сжатые данные будут использоваться многократно. В качестве примера можно привести цифровые аудио- и видеопроигрыватели.

Алгоритмы сжатия и восстановления требуют приблизительно равных вычислительных ресурсов.

Наиболее приемлемый вариант для линий связи, когда сжатие и восстановление происходит однократно на двух её концах (например, в цифровой телефонии).

Алгоритм сжатия существенно менее требователен, чем алгоритм восстановления.

Такая ситуация характерна для случаев, когда процедура сжатия реализуется простым, часто портативным, устройством, для которого объём доступных ресурсов весьма критичен, например, космический аппарат или большая распределённая сеть датчиков. Это могут быть также данные, распаковка которых требуется в очень малом проценте случаев, например запись камер видеонаблюдения.

2.4 Кодирование аудиоданных

2.4.1 Принципы оцифровки звука

Цифровой звук — это аналоговый звуковой сигнал, представленный посредством дискретных численных значений его амплитуды.

Оцифровка звука — технология осуществления замеров амплитуды звукового сигнала с определенным временным шагом и последующей записи полученных значений в численном виде. Другое название оцифровки звука — аналогово-цифровое преобразование звука.

Оцифровка звука включает в себя два процесса:

- процесс дискретизации (осуществление выборки) сигнала по времени
- процесс квантования по амплитуде.

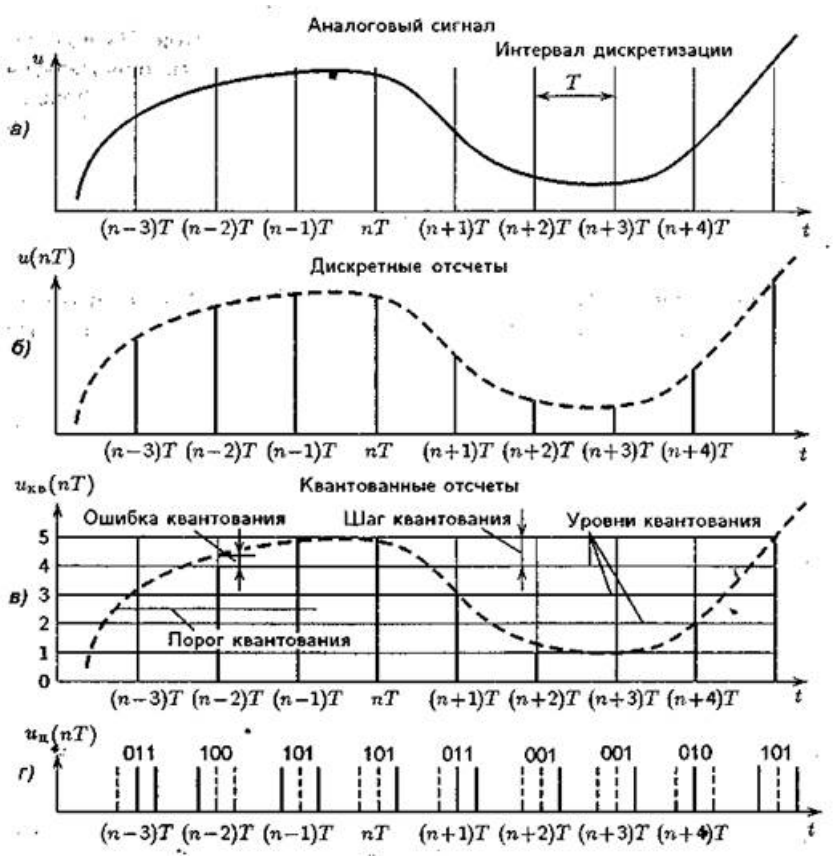

Рис. 2.4.1.1 Схема оцифровки звука

Процесс дискретизации по времени

Процесс дискретизации по времени — процесс получения значений сигнала, который преобразуется с определенным временным шагом — шагом дискретизации . Количество замеров величины сигнала, осуществляемых в единицу времени, называют частотой дискретизации или частотой выборки. Чем меньше шаг дискретизации, тем выше частота дискретизации и тем более точное представление о сигнале будет получено. Это подтверждается теоремой Котельникова. Согласно ей, аналоговый сигнал с ограниченным спектром точно описуем дискретной последовательностью значений его амплитуды, если эти значения берутся с частотой, как минимум вдвое превышающей наивысшую частоту спектра сигнала. То есть, аналоговый сигнал, в котором находится частота спектра равная F_m, может быть точно представлен последовательностью дискретных значений амплитуды, если для частоты дискретизации F_d выполняется: $F_d > 2F_m$.

На практике это означает, что для того, чтобы оцифрованный сигнал содержал информацию о всем диапазоне слышимых частот исходного аналогового сигнала (20 Гц — 20 кГц) необходимо, чтобы выбранное значение частоты дискретизации составляло не менее 40 кГц.

Основная трудность оцифровки заключается в невозможности записать измеренные значения сигнала с идеальной точностью.

Линейное квантование амплитуды

Отведём для записи одного значения амплитуды сигнала в памяти компьютера N бит. Значит, с помощью одного N-битного слова можно описать $2N$ разных положений. Пусть амплитуда оцифровываемого сигнала колеблется в пределах от -1 до 1 некоторых условных единиц. Представим этот диапазон изменения амплитуды — **динамический диапазон сигнала** — в виде $2N-1$ равных промежутков, разделив его на $2N$ уровней — **квантов**. Теперь, для записи каждого отдельного значения амплитуды, его необходимо округлить до ближайшего уровня квантования. Этот процесс носит название **квантования по амплитуде**.

Квантование по амплитуде — процесс замены реальных значений амплитуды сигнала значениями, приближенными с некоторой точностью. Каждый из 2^N возможных уровней называется **уровнем квантования**, а расстояние между двумя ближайшими уровнями квантования называется **шагом квантования**. Если амплитудная шкала разбита на уровни линейно, квантование называют линейным (однородным). Точность округления зависит от выбранного количества $2N$ уровней квантования, которое, в свою очередь, зависит от количества бит N, отведенных для записи значения амплитуды. Число N называют **разрядностью квантования**, а полученные в результате округления значений амплитуды числа — **отсчетами** или **семплами**. Принимается, что погрешности квантования, являющиеся результатом квантования с разрядностью 16 бит, остаются для слушателя почти незаметными. Этот способ оцифровки сигнала — дискретизация сигнала

во времени в совокупности с методом однородного квантования — называется **импульсно-кодовой модуляцией, ИКМ (англ. Pulse Code Modulation — PCM)**.

Оцифрованный сигнал в виде набора последовательных значений амплитуды уже можно сохранить в памяти компьютера. В случае, когда записываются абсолютные значения амплитуды, такой формат записи называется PCM (Pulse Code Modulation). Стандартный аудио компакт-диск (CD-DA), применяющийся с начала 80-х годов, хранит информацию в формате PCM с частотой дискретизации 44.1 кГц и разрядностью квантования 16 бит.

Аналогово-Цифровые Преобразователи

Вышеописанный процесс оцифровки звука выполняется **аналогово-цифровыми преобразователями (АЦП)**. Это преобразование включает в себя следующие операции:

1 **Ограничение полосы частот** производится при помощи фильтра нижних частот для подавления спектральных компонент, частота которых превышает половину частоты дискретизации.

2 **Дискретизация по времени**

3 **Квантование по уровню**

4 **Кодирование или оцифровка**, в результате которой значение каждого квантованного отсчета представляется в виде числа, соответствующего порядковому номеру уровня квантования.

2.4.2 Кодирование оцифрованного звука

Для хранения цифрового звука существует много различных способов. Оцифрованный звук являет собой набор значений амплитуды сигнала, взятых через определенные промежутки времени.

- Блок оцифрованной аудио информации можно записать в файл без изменений, то есть последовательностью чисел — значений амплитуды. В этом случае существуют два способа хранения информации.
  - Первый — PCM (Pulse Code Modulation — импульсно-кодовая модуляция) — способ цифрового кодирования сигнала при помощи записи абсолютных значений амплитуд. (В таком виде записаны данные на всех аудио CD.)
  - Второй — ADPCM (Adaptive Delta PCM — адаптивная относительная импульсно-кодовая модуляция) — запись значений сигнала не в абсолютных, а в относительных изменениях амплитуд (приращениях).
- Можно сжать данные так, чтобы они занимали меньший объем памяти, нежели в исходном состоянии. Тут тоже есть два способа.

– Кодирование данных без потерь (lossless coding) — способ кодирования аудио, который позволяет осуществлять стопроцентное восстановление данных из сжатого потока. К нему прибегают в тех случаях, когда сохранение оригинального качества данных особо значимо. Существующие сегодня алгоритмы кодирования без потерь (например, Monkeys Audio) позволяют сократить занимаемый данными объем на 20-50%, но при этом обеспечить стопроцентное восстановление оригинальных данных из полученных после сжатия.

– Кодирование данных с потерями (lossy coding). Здесь цель — добиться схожести звучания восстановленного сигнала с оригиналом при как можно меньшем размере сжатого файла. Это достигается путём использования алгоритмов, «упрощающих» оригинальный сигнал (удаляющих из него «несущественные», неразличимые на слух детали). Это приводит к тому, что декодированный сигнал перестает быть идентичным оригиналу, а является лишь «похоже звучащим». Методов сжатия, а также программ, реализующих эти методы, существует много. Наиболее известными являются MPEG-1 Layer I,II,III (последним является всем известный MP3), MPEG-2 AAC (advanced audio coding), Ogg Vorbis, Windows Media Audio (WMA), TwinVQ (VQF), MPEGPlus, TAC, и прочие. В среднем, коэффициент сжатия, обеспечиваемый такими кодерами, находится в пределах 10-14 (раз). В основе всех lossy-кодеров лежит использование так называемой психоакустической модели. Она занимается этим самым «упрощением» оригинального сигнала. Степень сжатия оригинального сигнала зависит от степени его «упрощения» — сильное сжатие достигается путём «воинственного упрощения» (когда кодером игнорируются множественные нюансы). Такое сжатие приводит к сильной потере качества, поскольку удалению могут подлежать не только незаметные, но и значимые детали звучания.

2.4.3 Сжатие аудиоданных без потерь

Сокращение статистической избыточности основано на учёте свойств самих звуковых сигналов. Она определяется наличием корреляционной связи между соседними отсчетами цифрового звукового сигнала, устранение которой позволяет сокращать объём передаваемых данных на 15-25 % по сравнению с их исходной величиной. Для передачи сигнала необходимо получить более компактное его представление, что возможно осуществить с помощью ортогонального преобразования. Важными условиями применения такого метода преобразования являются:

- возможность восстанавливать исходный сигнал без искажений
- способность обеспечивать наибольшую концентрацию энергии в небольшом числе коэффициентов преобразования
- быстрый вычислительный алгоритм

Этим требованиям отвечает модифицированное дискретно-косинусное преобразование (МДКП).

Уменьшить скорость цифрового потока позволяют методы кодирования, учитывающие статистику звуковых сигналов, например, вероятности появления уровней разной величины. Одним из таких методов является код Хаффмана, где наиболее вероятным значениям сигнала приписываются более короткие кодовые слова, а значения отсчетов, вероятность появления которых мала, кодируются кодовыми словами большей длины. Именно в силу этих двух причин в наиболее эффективных алгоритмах компрессии цифровых аудиоданных кодированию подвергаются не сами отсчеты звукового сигнала, а коэффициенты МДКП.

Подобные методы применяются при архивации файлов.

2.4.4 Сжатие аудиоданных с потерями

Сжатие аудиоданных с потерями основывается на несовершенстве человеческого слуха при восприятии звуковой информации. Неспособность человека в определённых случаях различать тихие звуки в присутствии более громких, называемая эффектом маскировки, была использована в алгоритмах сокращения психоакустической избыточности. Эффекты слухового маскирования зависят от спектральных и временных характеристик маскируемого и маскирующего сигналов и могут быть разделены на две основные группы:

- частотное (одновременное) маскирование
- временное (неодновременное) маскирование

Эффект маскирования в частотной области связан с тем, что в присутствии больших звуковых амплитуд человеческое ухо нечувствительно к малым амплитудам близких частот. То есть, когда два сигнала одновременно находятся в ограниченной частотной области, то более слабый сигнал становится неслышимым на фоне более сильного.

Маскирование во временной области характеризует динамические свойства слуха, показывая изменение во времени относительного порога слышимости (порог слышимости одного сигнала в присутствии другого), когда маскирующий и маскируемый сигналы звучат не одновременно. При этом следует различать явления послемаскировки (изменение порога слышимости после сигнала высокого уровня) и предмаскировки (изменение порога слышимости перед приходом сигнала максимального уровня). Более слабый сигнал становится неслышимым за 5 − 20 мс до включения сигнала маскирования и становится слышимым через 50 − 200 мс после его включения.

Наилучшим методом кодирования звука, учитывающим эффект маскирования, оказывается полосное кодирование. Сущность его заключается в следующем. Группа отсчетов входного звукового сигнала, называемая кадром, поступает на блок фильтров который разделяет сигнал на частотные поддиапазоны. На выходе каждого фильтра оказывается та часть входного сигнала, которая попадает в

полосу пропускания данного фильтра. Далее, в каждой полосе с помощью психоакустической модели, анализируется спектральный состав сигнала и оценивается, какую часть сигнала следует передавать без сокращений, а какая лежит ниже порога маскирования и может быть переквантована на меньшее число бит. Для сокращения максимального динамического диапазона определяется максимальный отсчет в кадре и вычисляется масштабирующий множитель, который приводит этот отсчет к верхнему уровню квантования. Эта операция аналогична компандированию в аналоговом вещании. На этот же множитель умножаются и все остальные отсчеты. Масштабирующий множитель передается к декодеру вместе с кодированными данными для коррекции коэффициента передачи последнего. После масштабирования производится оценка порога маскирования и осуществляется перераспределение общего числа битов между всеми полосами.

Очевидно, что после устранения психоакустической избыточности звуковых сигналов их точное восстановления при декодировании оказывается уже невозможным. Методами устранения психофизической избыточности можно обеспечить сжатие цифровых аудиоданных в $10 - 12$ раз без существенных потерь в качестве.

2.4.5 Структура кодера сжатия аудиоданный с потерями

- Исходный цифровой звуковой сигнал разделяется на частотные поддиапазоны и сегментируется по времени в блоке временной и частотной сегментации.

- Длина кодируемой выборки зависит от формы временной функции звукового сигнала. При отсутствии резких выбросов по амплитуде используется так называемая длинная выборка, обеспечивающая высокое разрешение по частоте. В случае же резких изменений амплитуды сигнала длина кодируемой выборки резко уменьшается, что дает более высокое разрешение по времени. Решение об изменении длины кодируемой выборки принимает блок психоакустического анализа, вычисляя значение психоакустической энтропии сигнала.

- После сегментации сигналы частотных поддиапазонов нормируются, квантуются и кодируются. В наиболее эффективных алгоритмах компрессии кодированию подвергаются не сами отсчеты выборки звукового сигнала, а соответствующие им коэффициенты МДКП.

- Учёт закономерностей слухового восприятия звукового сигнала выполняется в блоке психоакустического анализа. Здесь по специальной процедуре для каждого частотного поддиапазона рассчитывается максимально допустимый уровень искажений (шумов) квантования, при котором они ещё маскируются полезным сигналом данного поддиапазона.

- Блок динамического распределения бит в соответствии с требованиями психоакустической модели для каждого поддиапазона кодирования выделяет такое минимально возможное их количество, при котором уровень искажений,

вызванных квантованием, не превышал порога их слышимости, рассчитанного психоакустической моделью.

- Также могут использоваться:
  - матрицирование стерео — сложение и вычитание левого и правого канала для устранения повторяющейся информации
  - специальные процедуры итерационных циклов, позволяющие управлять величиной энергии искажений квантования в поддиапазонах при недостаточном числе доступных для кодирования бит
  - процедуры линейного и обратного адаптивного предсказаний
  - техника сглаживания переходных шумов во временной области (Temporal Noise Shaping — TNS), позволяющая управлять микроструктурой искажений квантования внутри каждого поддиапазона кодирования

Многие другие приёмы могут послужить способом сократить объём данных звуковой информации. Даже простое сужение полосы частот сигнала вместе с уменьшением динамического диапазона может уже называться сжатием аудиоданных. Например, в стандарте сжатия звука в сотовой связи используется и то и другое. Стремясь удалить избыточность из звука, кодек при плохом качестве сигнала становится избирателен к определённым словам, упорно проглатывая их.

Глава 3

Изучение алгоритмов блочного и поточного шифрования данных

3.1 Алгоритмы блочного шифрования данных

3.1.1 Определение и основные свойства блочного шифра

Блочный шифр — разновидность симметричного шифра, оперирующего группами бит фиксированной длины — блоками, характерный размер которых меняется в пределах 64–256 бит. Если исходный текст (или его остаток) меньше размера блока, перед шифрованием его дополняют. Фактически, блочный шифр представляет собой подстановку на алфавите блоков, которая, как следствие, может быть моно- или полиалфавитной. Блочный шифр является важной компонентой многих криптографических протоколов и широко используется для защиты данных, передаваемых по сети.

Блочный шифр способен зашифровать одним ключом одно или несколько сообщений суммарной длиной больше, чем длина ключа. От поточных шифров работа блочного отличается обработкой бит группами, а не потоком. При этом блочные шифры медленнее поточных. Симметричные системы обладают преимуществом над асимметричными в скорости шифрования, что позволяет им оставаться актуальными, несмотря на более слабый механизм передачи ключа (получатель должен знать секретный ключ, который необходимо передать по уже налаженному зашифрованному каналу. В то же время, в асимметричных шифрах открытый ключ, необходимый для шифрования, могут знать все, и нет необходимости в передаче ключа шифрования).

К достоинствам блочных шифров относят сходство процедур шифрования и расшифрования, которые, как правило, отличаются лишь порядком действий. Это упрощает создание устройств шифрования, так как позволяет использовать одни и те же блоки в цепях шифрования и расшифрования.

Гибкость блочных шифров позволяет использовать их для построения других криптографических примитивов: генератора псевдослучайной последовательности, поточного шифра, имитовставки и криптографических хешей.

Структура блочного шифра

Блочный шифр состоит из двух парных алгоритмов: шифрования и расшифрования. Оба алгоритма можно представить в виде функций. Функция шифрования E (encryption) на вход получает блок данных M (message) размером n бит и ключ K (key) размером k бит и на выходе отдает блок шифротекста C (cipher) размером n бит:

$$E_K(M) := E(K, M) : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n.$$

Для любого ключа K, E_K является биективной функцией (перестановкой) на множестве n-битных блоков. Функция расшифрования D (decryption) на вход получает шифротекст C, ключ K и на выходе отдает M:

$$D_K(C) := D(K, C) : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n,$$

являясь, при этом, обратной к функции шифрования:

$$D = E^{-1},$$
$$\forall K : \mathrm{D}_K(E_K(M)) = M$$
$$E_K(D_K(C)) = C.$$

Заметим, что ключ, необходимый для шифрования и дешифрования, один и тот же — следствие симметричности блочного шифра.

3.1.2 Алгоритм шифрования DES

Алгоритм шифрования данных DES (Data Encryption Standard) был опубликован в 1977 году. Является официальным стандартом США. Блочный симметричный алгоритм DES пока остается наиболее распространенным алгоритмом, используемым в системах защиты коммерческой информации.

Принцип работы DES

Алгоритм DES состоит из чередующейся последовательности перестановок и подстановок. Алгоритм DES осуществляет шифрование 64-битовых блоков данных с помощью 64-битового ключа, в котором значащими являются 56 бит (остальные 8 – проверочные биты для контроля четности).

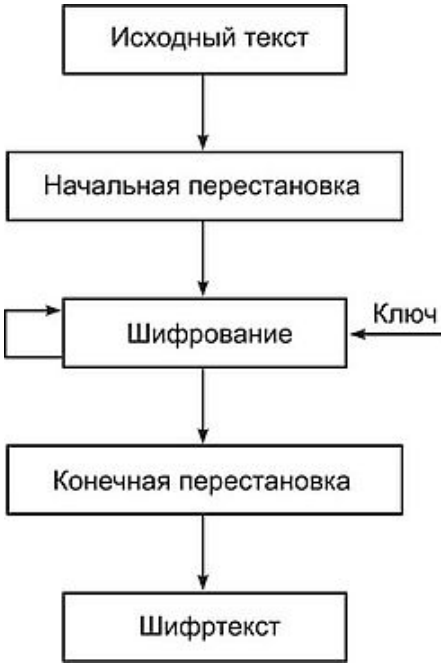

Рис. 3.1.2.1 Обобщенная схема шифрования в алгоритме DES

Процесс шифрования в блочном алгоритме DES (рис. 3.1.2.1) заключается в начальной перестановке битов 64-битового блока, шестнадцати циклах (раундах) шифрования и, наконец, в конечной перестановке битов. Расшифрование в DES является операцией, обратной шифрованию, и выполняется путем повторения операций шифрования в обратной последовательности.

Процесс зашифрования

Исходный текст T (блок 64 бит) преобразуется с помощью начальной перестановки IP которая определяется таблицей 3.1.2.1:

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Таблица 3.1.2.1 Начальная перестановка IP

По таблице первые 3 бита результирующего блока $IP(T)$ после начальной перестановкиявляются битами 58, 50, 42 входного блока , а его 3 последние бита являются битами 23, 15, 7 входного блока.

Полученный после начальной перестановки 64-битовый блок $IP(T)$ участвует в 16 циклах преобразования Фейстеля.

Преобразования Фейстеля- это преобразование над векторами (блоками), представляющими собой левую и правую половины регистра сдвига. В алгоритме DES используются прямое преобразование сетью Фейстеля в шифровании и обратное

преобразование сетью Фейстеля в расшифровании.

16 циклов преобразования Фейстеля:

Разбить IP(T) на две части L_0, R_0, где L_0, R_0 — соответственно 32 старших битов и 32 младших битов блока $T_0IP(T) = L_0R_0$

Пусть $T_{i-1} = L_{i-1}R_{i-1}$ результат (i-1) итерации, тогда результат i-ой итерации $T_i = L_iR_i$ определяется:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

Левая половина L_i равна правой половине предыдущего вектора $L_{i-1}R_{i-1}$. А правая половина R_i — это битовое сложение L_{i-1} и $f(R_{i-1}, k_i)$ по модулю 2.

В 16-циклах преобразования Фейстеля функция f играет роль шифрования.

Аргументами функции f являются 32-битовый вектор R_{i-1} и 48-битовый ключ k_i, который является результатом преобразования 56-битового исходного ключа шифра k. Для вычисления функции f последовательно используются

1 функция расширения E

2 сложение по модулю 2 с ключом k_i

3 преобразование S, состоящее из 8 преобразований S-блоков S_1,S_2,S_3… S_8

4 перестановка P

Генерирование ключей

Ключи k_i получаются из начального ключа k (56 бит = 7 байтов или 7 символов в ASCII) следующим образом. Добавляются биты в позиции 8, 16, 24, 32, 40, 48, 56, 64 ключа k таким образом, чтобы каждый байт содержал нечетное число единиц. Это используется для обнаружения ошибок при обмене и хранении ключей. Затем делают перестановку для расширенного ключа (кроме добавляемых битов 8, 16, 24, 32, 40, 48, 56, 64).

Процесс расшифрования

При расшифровании данных все действия выполняются в обратном порядке. В 16 циклах расшифрования, в отличие от шифрования с помощью прямого преобразования сетью Фейстеля, здесь используется обратное преобразование сетью Фейстеля.

$$R_{i-1} = L_i$$
$$L_{i-1} = R_i \oplus f(L_i, k_i)$$

Ключ k_i, i=16,…,1, функция f, перестановка IP и IP^{-1} такие же, как и в процессе шифрования. Алгоритм генерации ключей зависит только от ключа пользователя, поэтому при расшифровании они идентичны.

Режимы использования DES

DES может использоваться в четырёх режимах.

1 Режим электронной кодовой книги (ECB — Electronic Codebook): обычное использование DES как блочного шифра. Шифруемый текст разбивается на блоки, при этом каждый блок шифруется отдельно, не взаимодействуя с другими блоками (см. рис. 3.1.2.2)

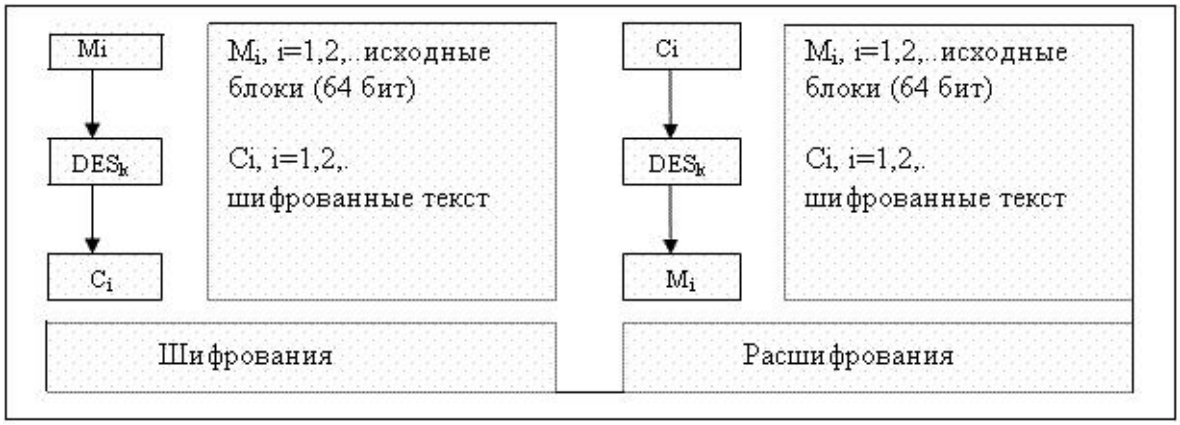

Рис. 3.1.2.2 Режим электронной кодовой книги — ECB

2 Режим сцепления блоков шифротекста (CBC — Cipher Block Chaining)(см. рис. 3.1.2.3). Каждый очередной блок M_i i>=1, перед зашифровыванием складывается по модулю 2 с предыдущим блоком зашифрованного текста C_{i-1}. Вектор C_0 — начальный вектор, он меняется ежедневно и хранится в секрете.

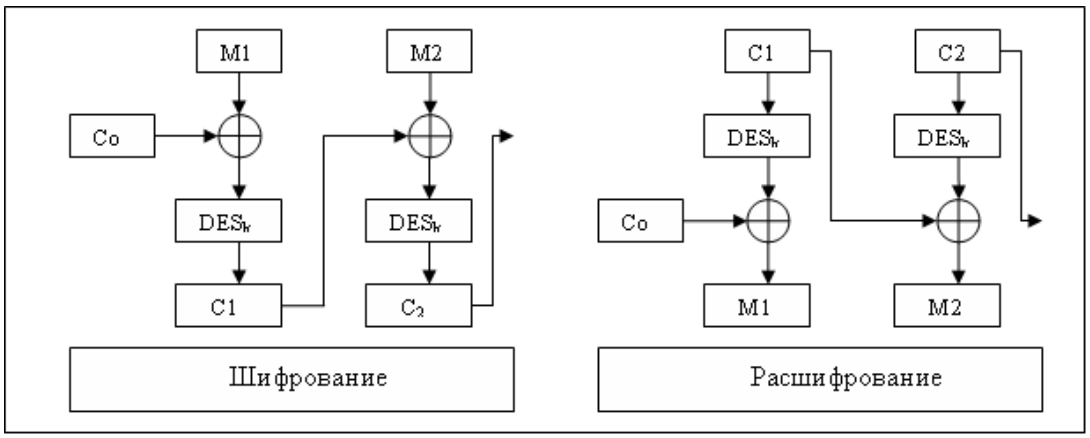

Рис. 3.1.2.3 Режим сцепления блоков — CBC

3 Режим обратной связи по шифротексту (Cipher Feedback) (см. рис. 3.1.2.4). В режиме CFB вырабатывается блочная «гамма» $Z_0, Z_1, ..., Z_i = DES_k(C_{i-1})$ $C_i = M_i \oplus Z_i$. Начальный вектор C_0 является синхропосылкой и предназначен для того, чтобы разные наборы данных шифровались по-разному с использованием одного и того же секретного ключа. Синхропосылка посылается получателю в открытом виде вместе с зашифрованным файлом. Алгоритм DES, в отличие от предыдущих режимов, используется только как шифрование (в обоих случаях).

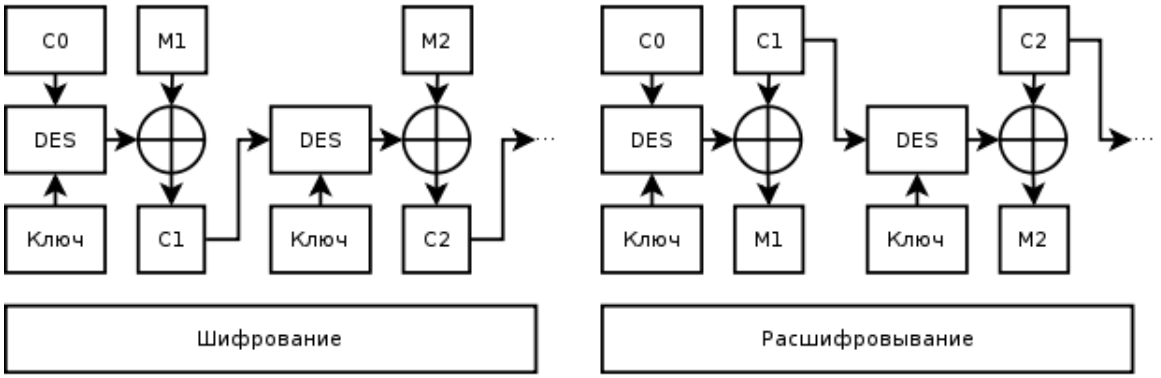

Рис.3.1.2.4 Режим обратной связи по шифротексту — CFB

4 Режим обратной связи по выходу (Output Feedback) (см. Рис. 3.1.2.5). В режиме OFB вырабатывается блочная «гамма» $Z_0, Z_1, ..., Z_i = DES_k(Z_{i-1})C_i = M_i \oplus Z_i$, i>=1. Режим также использует DES только как шифрование (в обоих случаях).

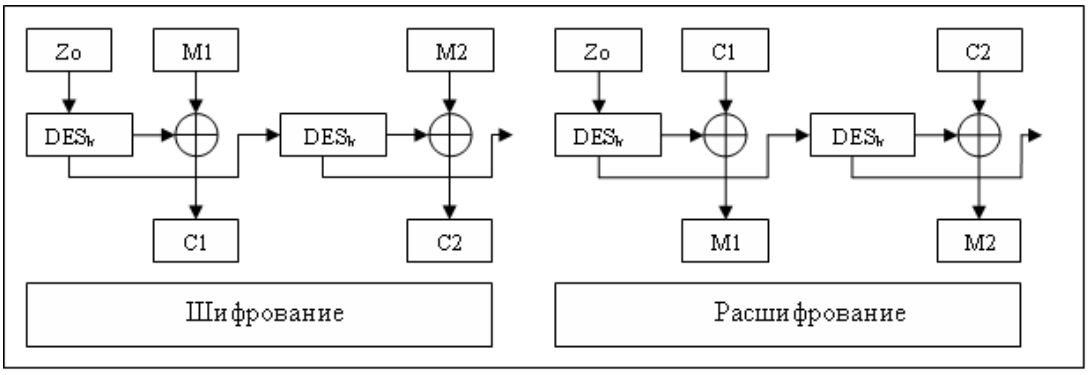

Рис. 3.1.2.5 Режим обратной связи по выходу — OFB

Достоинства и недостатки DES

Основные достоинства алгоритма шифрования DES:

- относительная простота алгоритма обеспечивает высокую скорость обработки;

- для расшифровки сообщения, зашифрованного с помощью одного пакета программ, можно использовать любой другой пакет программ, соответствующий алгоритму DES;
- криптостойкость алгоритма достаточна для защиты коммерческой информации.

Также алгоритм DES имеет ряд существенных недостатков. Основные из них

- битовые операции в узлах замены неэффективно реализуются программным путем;
- короткая длина ключа (56 битов), что помогает организовать полный перебор;

3.1.3 Алгоритм шифрования IDEA

IDEA (англ. International Data Encryption Algorithm, международный алгоритм шифрования данных) — симметричный блочный алгоритм шифрования данных, запатентованный швейцарской фирмой Ascom. Известен тем, что применяется в пакете программ шифрования PGP и в его свободной альтернативе GnuPG.

Первую версию алгоритма разработали в 1990 году Сюэцзя Лай и Джеймс Мэсси из Швейцарского института ETH Zürich в качестве замены DES и назвали ее PES (англ. Proposed Encryption Standard, предложенный стандарт шифрования). Затем, после публикации работ Бихама и Шамира по дифференциальному криптоанализу PES, алгоритм был улучшен с целью усиления криптостойкости и назван IPES (англ. Improved Proposed Encryption Standard, улучшенный предложенный стандарт шифрования). Через год его переименовали в IDEA.

Принцип работы

IDEA использует 128-битный ключ и 64-битный размер блока. Открытый текст разбивается на блоки по 64 бит, если такое разбиение невозможно, используются различные режимы шифрования.Каждый исходный незашифрованный 64-битный блок делится на четыре подблока по 16 бит каждый, так как все алгебраические операции, использующиеся в процессе шифрования, совершаются над 16-битными числами. Для шифрования и расшифрования IDEA использует один и тот же алгоритм.

Фундаментальным нововведением в алгоритме является использование операций из разных алгебраических групп, а именно:

- сложение по модулю 2^{16}
- умножение по модулю $2^{16} + 1$
- побитовое исключающее ИЛИ (XOR).

Эти три операции несовместимы в том смысле, что:

- никакие две из них не удовлетворяют дистрибутивному закону, то есть $a * (b + c) \ <> \ (a * b) + (a * c)$
- никакие две из них не удовлетворяют ассоциативному закону, то есть $a + (b \oplus c) \ <> \ (a + b) \oplus c$

Применение этих трех операций затрудняет криптоанализ IDEA по сравнению с DES, который основан исключительно на операции исключающее ИЛИ, а также позволяет отказаться от использования S-блоков и таблиц замены. IDEA является модификацией сети Фейстеля.

Шифрование

Процесс шифрования состоит из восьми одинаковых раундов шифрования и одного выходного преобразования. Исходный незашифрованный текст делится на блоки по 64 бита. Каждый такой блок делится на четыре подблока по 16 бит каждый. На рисунке эти подблоки обозначены D_1, D_2, D_3, D_4. В каждом раунде используются свои подключи согласно таблице подключей. Над 16-битными подключами и подблоками незашифрованного текста производятся следующие операции:

- умножение по модулю $2^{16} + 1 = 65537$, причем вместо нуля используется 2^{16}
- сложение по модулю 2^{16}
- побитовое исключающее ИЛИ

В конце каждого раунда шифрования имеется четыре 16-битных подблока, которые затем используются как входные подблоки для следующего раунда шифрования. Выходное преобразование представляет собой укороченный раунд, а именно, четыре 16-битных подблока на выходе восьмого раунда и четыре соответствующих подключа подвергаются операциям:

- умножение по модулю $2^{16} + 1$
- сложение по модулю 2^{16}

После выполнения выходного преобразования конкатенация подблоков D'_1, D'_2, D'_3 и D'_4 представляет собой зашифрованный текст. Затем берется следующий 64-битный блок незашифрованного текста и алгоритм шифрования повторяется. Так продолжается до тех пор, пока не зашифруются все 64-битные блоки исходного текста.

Структура алгоритма IDEA показана на рисунке 3.1.3.1.

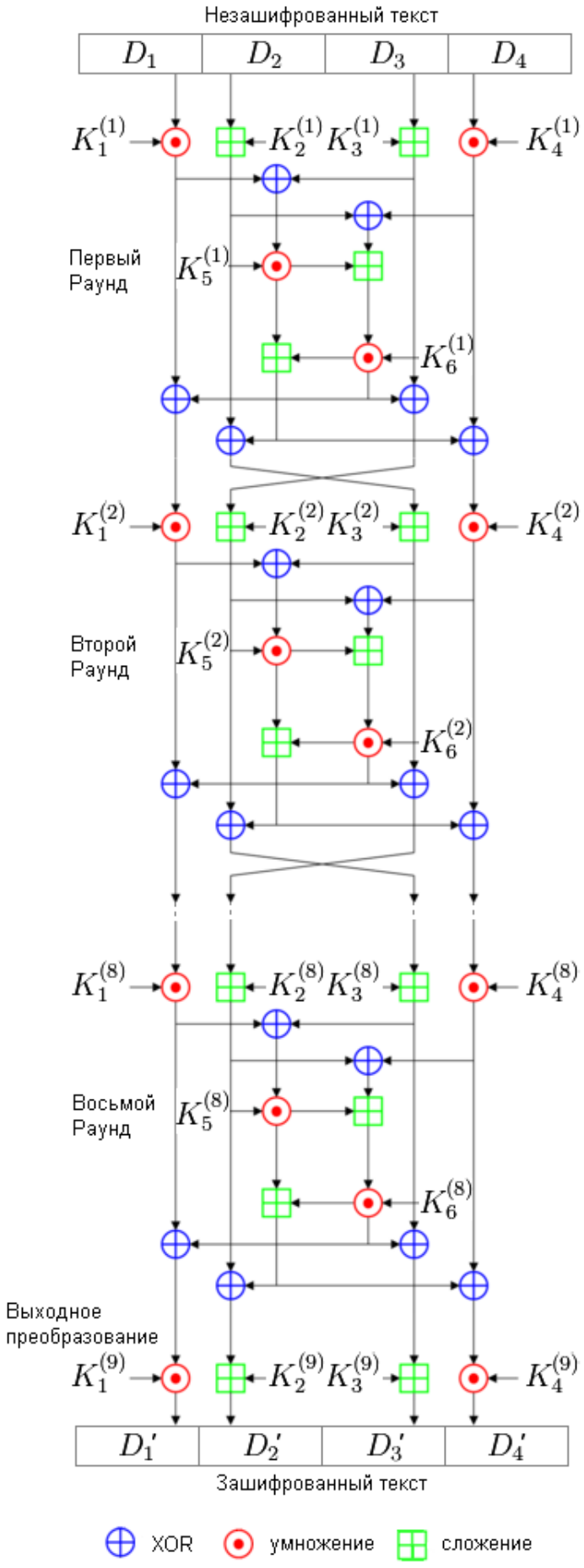

Рис. 3.1.3.1 Схема шифрования IDEA

Расшифровка

Метод вычисления, использующийся для расшифровки текста по существу такой же, как и при его шифровании. Единственное отличие состоит в том, что для расшифровки используются другие подключи.

В процессе расшифровки подключи должны использоваться в обратном порядке. Первый и четвёртый подключи i-го раунда расшифровки получаются из первого и четвёртого подключа (10-i)-го раунда шифрования мультипликативной инверсией. Для 1-го и 9-го раундов второй и третий подключи расшифровки получаются из второго и третьего подключей 9-го и 1-го раундов шифрования аддитивной инверсией. Для раундов со 2-го по 8-й второй и третий подключи расшифровки получаются из третьего и второго подключей с 8-го по 2-й раундов шифрования аддитивной инверсией. Последние два подключа i-го раунда расшифровки равны последним двум подключам (9-i)-го раунда шифрования.

Мультипликативная инверсия подключа K обозначается 1/K и $(1/K) * K = 1\ mod\ (2^{16}+1)$. Так как $2^{16}+1$ — простое число, каждое целое не равное нулю K имеет уникальную мультипликативную инверсию по модулю $2^{16}+1$. Аддитивная инверсия подключа K обозначается -K и $-K+K=0\ mod\ (2^{16})$.

Достоинства и недостатки IDEA

Преимущества алгоритма IDEA:

- В программной реализации на Intel486SX по сравнению с DES IDEA в два раза быстрее, что является существенным повышением скорости, длина ключа у IDEA имеет размер 128 бит, против 56 бит у DES, что является хорошим улучшением против полного перебора ключей;
- Использование IDEA в параллельных режимах шифрования на процессорах Pentium III и Pentium MMX позволяет получать высокие скорости;
- Хорошая изученность и устойчивость к общеизвестным средствам криптоанализа.

Недостатки алгоритма IDEA:

- IDEA значительно медленнее, почти в два раза, чем Blowfish;
- IDEA не предусматривает увеличение длины ключа.

3.1.4 Алгоритм шифрования RC5

RC5 (Ron's Code 5 или Rivest's Cipher 5) — это блочный шифр, разработанный Роном Ривестом из компании RSA Security с переменным количеством раундов, длиной блока и длиной ключа. Он отличается простотой, быстротой (за счет использования только примитивных компьютерных операций, таких как XOR, shift

и т. Д.) И использует меньше памяти. Это расширяет сферу использования и упрощает переход на более сильный вариант алгоритма.

Существует несколько различных вариантов алгоритма, в которых преобразования в «пол-раундах» классического RC5 несколько изменены. В классическом алгоритме используются три примитивных операции и их инверсии:

- сложение по модулю 2^w
- побитовое исключающее ИЛИ (XOR)
- операции циклического сдвига на переменное число бит $(x \lll y)$.

Основным нововведением является использование операции сдвига на переменное число бит, не использовавшиеся в более ранних алгоритмах шифрования. Эти операции одинаково быстро выполняются на большинстве процессоров, но в то же время значительно усложняют дифференциальный и линейный криптоанализ алгоритма.

Шифрование по алгоритму RC5 состоит из двух этапов. Процедура расширения ключа и непосредственно шифрование. Для расшифрования выполняется сначала процедура расширения ключа, а затем операции, обратные процедуре шифрования. Все операции сложения и вычитания выполняются по модулю 2^w.

Параметры алгоритма

Так как алгоритм RC5 имеет переменные параметры, то для спецификации алгоритма с конкретными параметрами принято обозначение **RC5-W/R/b**, где:

- W — половина длины блока в битах, возможные значения 16, 32 и 64. Для эффективной реализации величину W рекомендуют брать равным машинному слову. Например, для 32-битных платформ оптимальным будет выбор W=32, что соответствует размеру блока 64 бита.
- R — число раундов, возможные значения от 0 до 255. Увеличение числа раундов обеспечивает увеличение уровня безопасности шифра. Так, при R=0 информация шифроваться не будет. Также алгоритм RC5 использует таблицу расширенных ключей размера $2(R + 1)$ слов, которая получается из ключа заданного пользователем.
- b — длина ключа в байтах, возможные значения от 0 до 255.

Расширение ключа

Перед непосредственно шифрованием или расшифрованием данных выполняется процедура расширения ключа. Процедура генерации ключа состоит из четырёх этапов:

- Генерация констант

- Разбиение ключа на слова
- Построение таблицы расширенных ключей
- Перемешивание

Генерация констант

Для заданного параметра W генерируются две псевдослучайные величины используя две математические константы: e (экспонента) и f (Золотое сечение).

$Q_w \leftarrow \text{Odd}((f-1) \cdot 2^w)$

$P_w \leftarrow \text{Odd}((e-2) \cdot 2^w)$

Odd() - округление до ближайшего нечетного целого.

Разбиение ключа на слова

На этом этапе происходит копирование ключа $K_0 \ldots K_{b-1}$ в массив слов $L_0 \ldots L_{c-1}$, где $c = b/u$, где $u = W/8$, то есть, количество байт в слове.

Если b не кратен $W/8$, то L дополняется нулевыми битами до ближайшего большего размера c, кратного $W/8$.

В случае если $b = c = 0$, то мы устанавливаем значение $c = 1$, а $L_0 = 0$.

Построение таблицы расширенных ключей

На этом этапе происходит построение таблицы расширенных ключей $S_0 \ldots S_{2 \square (R+1)-1}$, которое выполняется следующим образом:

$S_0 = P_w$

$S_{i+1} = S_i + Q_w$

Перемешивание

Циклически N раз выполняются следующие действия:

$G = S_i = (S_i + G + H) \lll 3$

$H = L_j = (L_j + G + H) \lll (G + H)$

$i = (i + 1) \mod (2(R + 1))$

$j = (j + 1) \mod c$,

G, H, i, j — временные переменные, начальные значения которых равны 0. Количество итераций цикла N — это максимальное из двух значений $3 \square c$ и $(3 \cdot 2 \cdot (R + 1))$

Шифрование

Перед первым раундом выполняются операции наложения расширенного ключа на шифруемые данные:

$A = (A + S_0) \mod 2^w$

$B = (B + S_1) \mod 2^w$

В каждом раунде выполняются следующие действия:

$A = ((A \oplus B) \lll B) + S_{2i}$

$B = ((B \oplus A) \lll A) + S_{2i+1}$

Расшифрование

Для Расшифрования данных используются обратные операции, то есть для $i = R, R-1, ..., 1$ выполняются следующие раунды:

$B = ((B - S_{2i+1}) \ggg A) \oplus A$

$A = ((A - S_{2i}) \ggg B) \oplus B$

После выполнения всех раундов, исходное сообщение находится из выражения:

$B = (B - S_1) \bmod 2^w$

$A = (A - S_0) \bmod 2^w$

3.2 Алгоритмы поточного шифрования данных

3.2.1 Определение и основные свойства поточного шифра

Поточный или потоковый шифр — это симметричный шифр, 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 (тот же принцип и для шифрования аудиоданных).

Характерной особенностью поточных шифров есть побитная обработка информации. При этом шифрование и дешифрование может обрываться в произвольный момент времени. И как только связь восстановлена можно продолжать процедуру без проблем.

Поточные шифры называют шифрами гаммирования. Также само шифрование есть методом защиты информации. Эти шифры в разы быстрее своих конкурентов — блочных шифров, если оно реализовано аппаратно. Если же реализация программная, здесь скорость может быть даже меньше блоковых шифров. Функция которая формирует гамму, руководствуется тремя компонентами:

- ключ;
- номер текущего шага шифрования;
- ближние биты исходного или зашифрованного текста от текущей позиции.

На рисунке 3.2.1.1 представлен режим гаммирования потоковых шифров.

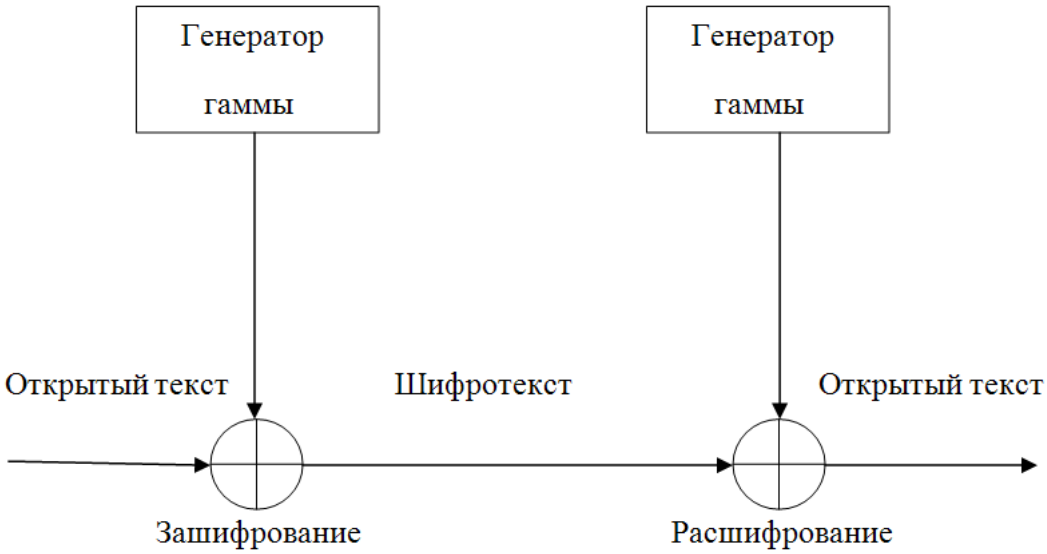

Рис. 3.2.1.1 Режим гаммирования для поточных шифров

Классификация потоковых шифров

Для предотвращения потери информации решают проблему синхронизации шифрования и расшифрования текста. По способу решения этой проблемы шифросистемы подразделяются на синхронные и системы с самосинхронизацией.

Синхронные потоковые шифры

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.

Обычно синхронизация производится вставкой в передаваемое сообщение специальных маркеров. В результате этого пропущенный при передаче знак приводит к неверному расшифрованию лишь до тех пор, пока не будет принят один из маркеров.

Самосинхронизирующиеся потоковые шифры

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

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

3.2.2 Алгоритм шифрования RC4

RC4 (Rivest cipher 4 или Ron's code), также известен как ARC4 или ARCFOUR) — потоковый шифр, широко применяющийся в различных системах защиты информации в компьютерных сетях (например, в протоколах SSL и TLS, алгоритмах обеспечения безопасности беспроводных сетей WEP и WPA).

Алгоритм RC4 строится на основе генератора псевдослучайных битов. На вход генератора записывается ключ, а на выходе читаются псевдослучайные биты. Длина ключа может составлять от 40 до 2048 бит. Генерируемые биты имеют равномерное распределение. Генерируемые биты имеют равномерное распределение.

Принцип работы

Ядро алгоритма поточных шифров состоит из функции — генератора псевдослучайных битов (гаммы), который выдаёт поток битов ключа (ключевой поток, гамму, последовательность псевдослучайных битов).

Алгоритм шифрования

1 Функция генерирует последовательность битов k_i;

2 Затем последовательность битов посредством операции «суммирование по модулю два» (xor) объединяется с открытым текстом m_i. В результате получается шифрограмма c_i: $c_i = m_i \oplus k_i$.

Алгоритм расшифровки

1 Повторно создаётся (регенерируется) поток битов ключа (ключевой поток) k_i;

2 Поток битов ключа складывается с шифрограммой c_i операцией «xor». В силу свойств операции «xor» на выходе получается исходный (незашифрованный) текст m_i: $m_i = c_i \oplus k_i = (m_i \oplus k_i) \oplus k_i$

RC4 — фактически класс алгоритмов, определяемых размером блока (в дальнейшем S-блока). Параметр n является размером слова для алгоритма и определяет длину S-блока. Обычно, n = 8, но в целях анализа можно уменьшить его. Однако для повышения безопасности необходимо увеличить эту величину. В алгоритме нет противоречий на увеличение размера S-блока . При увеличении n, допустим, до 16 бит, элементов в S-блоке становится 65 536 и соответственно время начальной итерации будет увеличено. Однако, скорость шифрования возрастёт.

Внутреннее состояние RC4 представляется в виде массива размером 2n и двух счётчиков. Массив известен как S-блок, и далее будет обозначаться как S. Он всегда содержит перестановку 2n возможных значений слова. Два счётчика обозначены через i и j.

Инициализация RC4 состоит из двух частей:

1 инициализация S-блока;

2 генерация псевдослучайного слова K.

Инициализация S-блока

Алгоритм также известен как «key-scheduling algorithm» или «KSA». Этот алгоритм использует ключ, подаваемый на вход пользователем, сохранённый в Key, и имеющий длину L байт. Инициализация начинается с заполнения массива S, далее этот массив перемешивается путём перестановок, определяемых ключом. Так как только одно действие выполняется над S, то должно выполняться утверждение, что S всегда содержит один набор значений, который был дан при первоначальной инициализации (S[i] := i).

```
for i from 0 to 255
     S[i] := i
endfor
j := 0
for i from 0 to 255
     j := ( j + S[i] + Key[ i mod L ] ) mod 256
endfor
```

Генерация псевдослучайного слова K

Эта часть алгоритма называется генератором псевдослучайной последовательности (англ. pseudo-random generation algorithm, PRGA). Генератор ключевого потока RC4 переставляет значения, хранящиеся в S. В одном цикле RC4 определяется одно n-битное слово K из ключевого потока. В дальнейшем ключевое слово будет сложено по модулю два с исходным текстом, которое пользователь хочет зашифровать, и получен зашифрованный текст.

```
i := 0
j := 0
while GenCycle:
     i := ( i + 1 ) mod 256
     j := ( j + S[i] ) mod 256
     t := ( S[i] + S[j] ) mod 256
     K := S[t]
endwhile
```

3.2.3 Алгоритм шифрования VMPC

VMPC (Variably Modified Permutation Composition) — это потоковый шифр, применяющийся в некоторых системах защиты информации в компьютерных сетях. Шифр разработан криптографом Бартошем Жултаком в качестве усиленного варианта популярного шифра RC4. Алгоритм VMPC строится как и любой потоковый шифр на основе параметризованного ключом генератора псевдослучайных битов. Основные преимущества шифра, как и RC4 — высокая скорость работы, переменный размер ключа и вектора инициализации (от 128 до 512 бит включительно), простота реализации (буквально несколько десятков строк кода).

Основа шифра - генератор псевдослучайных чисел, базой которого является односторонняя необратимая функция VMPC.

Приниципп работы

Ключевое расписание

Алгоритм преобразования ключа и (дополнительно) вектора инициализации в 256-элементную перестановку P. Инициализация глобальной переменной s.

C : Длина ключа в байтах $(16 \leq c \leq 64)$
K : Ключ
z : Длина вектора инициализации в байтах $(16 \leq z \leq 64)$
V : Вектор инициализации
i : 8-разрядная переменная
j : 16-разрядная переменная
s : 8-разрядная глобальная переменная
P : таблица из 256 байт для хранения перестановок

```
1.  s = 0
2.  for i = 0 to 255: P[i] = i

3.  for j = 0 to 767
4.  i = j mod 256
5.  s = P[(s + P[i] + K[j mod c]) mod 256]
6.  Temp = P[i]
       P[i] = P[s]
       P[s] = Temp
for j = 0 to 767
7.  i = j mod 256
8.  s = P[(s + P[i] + V[j mod z]) mod 256]
9. Temp = P[i]
    P[i] = P[s]
    P[s] = Temp
```

Алгоритм зашифрования

Генерация выходной ключевой последовательности. Для генерации L байт выходного ключевого потока выполняются следующие операции:

L : длина ключевой последовательности в байтах

```
1. i = 0
3. s = P[(s + P[i]) mod 256]
4. Output = P[(P[P[s]] + 1) mod 256]
5. Temp = P[i]
      P[i] = P[s]
      P[s] = Temp
6. i = (i + 1) mod 256
```

Особенности алгоритма VMPC

Вероятность получения двух последовательных одинаковых результатов при генерации ключевой последовательности при использовании шифра VMPC равна 2^{-N} что совпадает с соответствующей вероятностью идеального генератора случайной последовательности. N - число разрядов внутреннего состояния генератора псевдослучайной последовательности, обычно равно 8. В 2005 году А. Максимов показал, что на основании 2^{40} выходных бит возможно отличить последовательность генератора VMPC от случайного потока.

Эксперименты, проведенные Б.Жултаком, показали, что не наблюдается статистически значимого отклонения вероятности появления в выходной последовательности:

- каждого из возможных 2^8 значений $P = 1/256$ для $2^{41.85}$ байт выходной последовательности);
- каждой из возможных 2^{16} пар последовательных значений $P = 1/65536$ для $2^{40.1}$ байт выходной последовательности);
- каждой из возможных 2^{24} троек последовательных значений $P = 1/16777216$ для $2^{41.6}$ байт выходной последовательности)

3.2.4 Алгоритм шифрования A5

A5 — это поточный алгоритм шифрования, используемый для обеспечения конфиденциальности передаваемых данных между телефоном и базовой станцией в европейской системе мобильной цифровой связи GSM (Groupe Spécial Mobile).

Шифр основан на побитовом сложении по модулю два (булева операция «исключающее или») генерируемой псевдослучайной последовательности и шифруемой информации. В A5 псевдослучайная последовательность реализуется на основе трёх линейных регистров сдвига с обратной связью. Регистры имеют длины 19, 22 и 23 бита соответственно. Сдвигами управляет специальная схема, организующая на каждом шаге смещение как минимум двух регистров, что приводит

к их неравномерному движению. Последовательность формируется путём операции «исключающее или» над выходными битами регистров.

Принцип работы

Алгоритм A5 в настоящее время — это целое семейство шифров. Для описания возьмем A5/1 как родоначальника этого семейства.

Потоковое шифрование В этом алгоритме каждому символу открытого текста соответствует символ шифротекста. Текст не делится на блоки (как в блочном шифровании) и не изменяется в размере. Для упрощения аппаратной реализации и, следовательно, увеличения быстродействия используются только простейшие операции: сложение по модулю 2 (XOR) и сдвиг регистра.

Формирование выходной последовательности происходит путём сложения потока исходного текста с генерируемой последовательностью (гаммой). Особенность операции XOR заключается в том, что применённая чётное число раз, она приводит к начальному значению. Отсюда, декодирование сообщения происходит путём сложения шифротекста с известной последовательностью.

В реальных системах создаётся ключ заданного размера, который без труда передаётся по закрытому каналу. Последовательность генерируется на его основе и является псевдослучайной. Большой класс поточных шифров (в том числе A5) составляют шифры, генератор псевдослучайной последовательности которой основан на регистрах сдвига с линейной обратной связью.

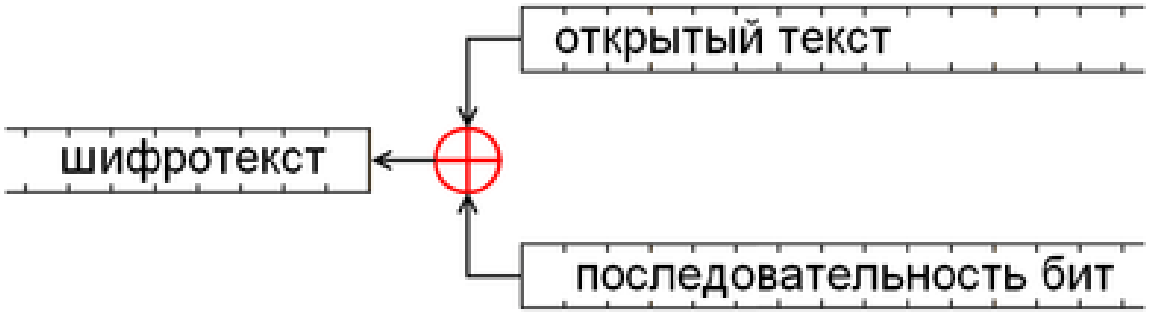

Рис. 3.2.4.1 Схема поточного шифра: сложение открытого текста и последовательности бит даёт шифротекст

Система РСЛОС в A5

Регистр сдвига с линейной обратной связью состоит из собственно регистра (последовательности бит заданной длины) и обратной связи. На каждом такте происходит следующие действия: крайний левый бит (старший бит) извлекается, последовательность сдвигается влево и в опустевшую правую ячейку (младший бит) записывается значение функции обратной связи. Эта функция является суммированием по модулю два определённых битов регистра и записывается в виде многочлена, где степень указывает номер бита. Извлечённые биты формируют выходную последовательность.

Сам по себе РСЛОС легко поддаётся криптоанализу и не является достаточно надёжным для использования в шифровании. Практическое применение имеют системы регистров переменного тактирования с различными длинами и функциями обратной связи.

Структура алгоритма A5 выглядит следующим образом:

- Три регистра(R1, R2, R3) имеют длины 19, 22 и 23 бита,
- Многочлены обратных связей:
  - $X^{19} + X^{18} + X^{17} + X^{14} + 1$ для R1,
  - $X^{22} + X^{21} + 1$ для R2 и
  - $X^{23} + X^{22} + X^{21} + X^8 + 1$ для R3,
- Управление тактированием осуществляется специальным механизмом:
  - в каждом регистре есть биты синхронизации: 8 (R1), 10 (R2), 10 (R3),
  - вычисляется функция $F = x\&y|x\&z|y\&z$, где & — булево AND, | - булево OR, а x, y и z — биты синхронизации R1, R2 и R3 соответственно,
  - сдвигаются только те регистры, у которых бит синхронизации равен F,
  - фактически, сдвигаются регистры, синхробит которых принадлежит большинству,
- Выходной бит системы — результат операции XOR над выходными битами регистров.

Функционирование алгоритма

Рассмотрим особенности функционирования алгоритма на основе известной схемы. Передача данных осуществляется в структурированном виде — с разбивкой на кадры (114 бит). Перед инициализацией регистры обнуляются, на вход алгоритма поступают сеансовый ключ (K — 64 бита), сформированный A8, и номер кадра (Fn — 22 бита). Далее последовательно выполняются следующие действия:

- Инициализация:
  - 64 такта, при которых очередной бит ключа XOR-ится с младшим битом каждого регистра, регистры при этом сдвигаются на каждом такте,
  - аналогичные 22 такта, только операция XOR производится с номером кадра,
  - 100 тактов с управлением сдвигами регистров, но без генерации последовательности,
- $228(114 + 114)$ тактов рабочие, происходит шифрование передаваемого кадра (первые 114 бит) и дешифрование (последние 114 бит) принимаемого,
- далее инициализация производится заново, используется новый номер кадра.

Литература

[1] https://ru.wikipedia.org

[2] https://www.sviaz-expo.ru/ru/ui/17142/

[3] https://www.osp.ru/nets/1996/07/141827

[4] http://asvagroup.com/2020/12/rasprostranenie-radiovoln-v-srede-i-peredacha-dannyh/

[5] http://www.comprice.ru/articles/detail.php?ID=40210

[6] https://ru.wikipedia.org/wiki/Сжатие_данных

[7] https://ru.wikipedia.org/wiki/Кодирование_звуковой_информации