Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

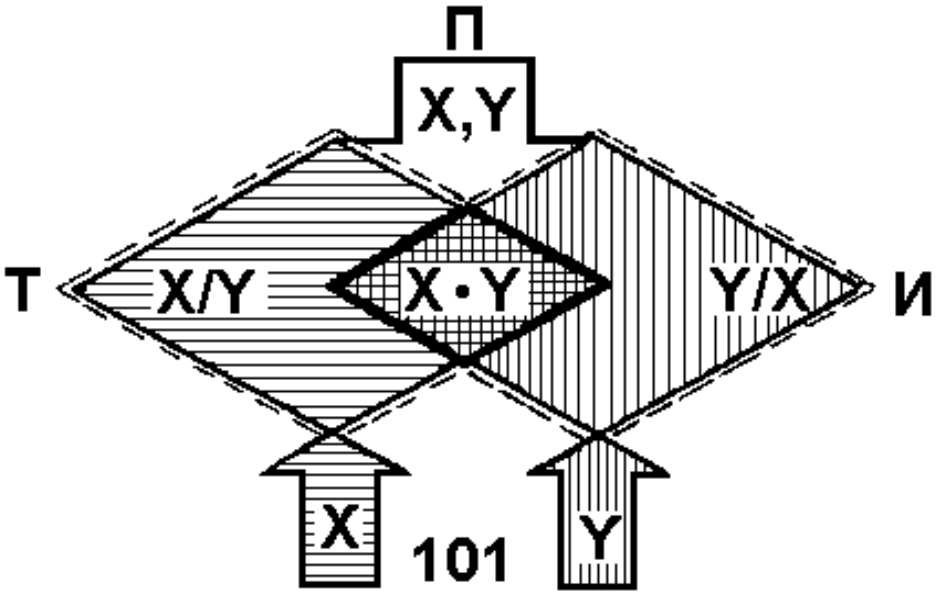
Кафедра систем управления

Н. И. Сорока, Г. А. Кривинченко

ТЕОРИЯ ПЕРЕДАЧИ ИНФОРМАЦИИ

Конспект лекций

для студентов специальности
1-53 01 07 «Информационные технологии и управление в технических системах»



Минск

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В.1. Определение информации

Деятельность людей связана с переработкой и использованием материалов, энергии и информации. Соответственно развиваются научно-технические дисциплины, отражающие вопросы технологии, энергетики и информатики. Условием успешной практической деятельности людей является эффективная организация обмена информацией (от латинского informatio-разьяснение, изложение). Таким образом, информационные наука и техника занимают одно из базовых положений. К информационной технике относятся средства, служащие для восприятия, подготовки, передачи, переработки, хранения и представления какой-либо информации, получаемой от человека, природы машины, вообще от какого-либо объекта наблюдения и управления. Комплексное применение этих средств приводит к созданию больших и сложных информационных систем.

Имеется множество определений понятия информации от наиболее общего философского – информация есть отражение реального мира до наиболее узкого практического – информация есть все сведения, являющиеся объектом хранения, передачи и преобразования.

В [11] информация определяется как содержательные сведения (данные), заключенные в том или другом сообщении, заранее не известные человеку или машине, принимающим сообщение.

Понятие информации связано с некоторыми моделями реальных вещей, отражающими их сущность в той степени, в какой это необходимо для практических целей. Это согласуется и с философской концепцией отражения вещей друг в друге и в живых организмах.

Таким образом, под информацией нужно понимать не сами предметы и процессы, а их представительные характеристики отражения или отображения в виде чисел, формул, описаний, чертежей, символов, образов и других абстрактных характеристик.

Сама по себе информация может быть отнесена к области абстрактных категорий, подобных, например, математическим формулам. Однако проявляется она всегда в материально-энергетической форме в виде сигналов. Схема образования сигнала показана на рис. В.1.

Различают две основные формы существования информации: статическая в виде различных записей на бумаге, пленке и других материалах и динамическая - при ее передаче.

С передачей и обработкой информации связаны действия любого автоматического устройства, поведение живого существа, творческая деятельность человека, развитие науки и техники, экономические и социальные преобразования в обществе и сама жизнь.

Теория информации в ее современном виде – это научная дисциплина, изучающая способы передачи и хранения информации наиболее надежным и

экономичным методом и ее знание поможет поразобраться с информационными с информационными процессами, протекающими в системах телемеханических, радиотехнических и системах связи (рис. В.2).

На этапе восприятия формируется образ объекта, производится его опознавание и оценка. При этом необходимо отделить полезную информацию от шумов, что в ряде случаев сопряжено со значительными трудностями.

На этапе подготовки информации производятся такие операции, как нормализация, квантование, кодирование и модуляция носителя. Иногда этот этап рассматривается как вспомогательный на этапе восприятия. В результате восприятия и подготовки получается сигнал в форме, удобной для передачи и обработки.

Передача информации состоит в переносе ее на расстояние посредством сигналов различной физической природы соответственно по электрическим, электромагнитным и оптическим каналам. Прием информации на другой стороне канала имеет характер вторичного восприятия со свойственными ему операциями борьбы с шумами.

Обработка информации заключается в решении задач, связанных с ее преобразованием, независимо от их функционального назначения.

Преобразование информации осуществляется либо средствами информационной техники, либо человеком. Если процесс обработки формализуем, он может выполняться техническими средствами. В современных сложных системах эти функции возлагаются на ЭВМ и микропроцессоры. Если процесс обработки не поддается формализации и требует творческого подхода, обработка информации осуществляется человеком.

Этап отображения информации должен предшествовать этапам, связанным с участием человека. Цель этапа отображения - представить человеку нужную ему информацию с помощью устройств, способных воздействовать на его органы чувств.

На этапе воздействия информация используется для осуществления необходимых изменений в системе.

В.2. Система передачи информации

Структурная схема одноканальной системы передачи информации приведена на рис. В.4. Информация поступает в систему в форме сообщений. Под сообщением понимают совокупность знаков или первичных сигналов, содержащих информацию. Источник сообщений в общем случае образует совокупность источника информации ИИ (исследуемого или наблюдаемого объекта) и первичного преобразователя ПП (датчика, человека-оператора и т.п.), воспринимающего информацию о его состоянии или о протекающем в нем процессе. Различают дискретные и непрерывные сообщения.

Дискретные сообщения формируются в результате последовательной выдачи источником отдельных элементов-знаков. Множество различных знаков называют алфавитом источника сообщений, а число знаков – объемом алфавита.

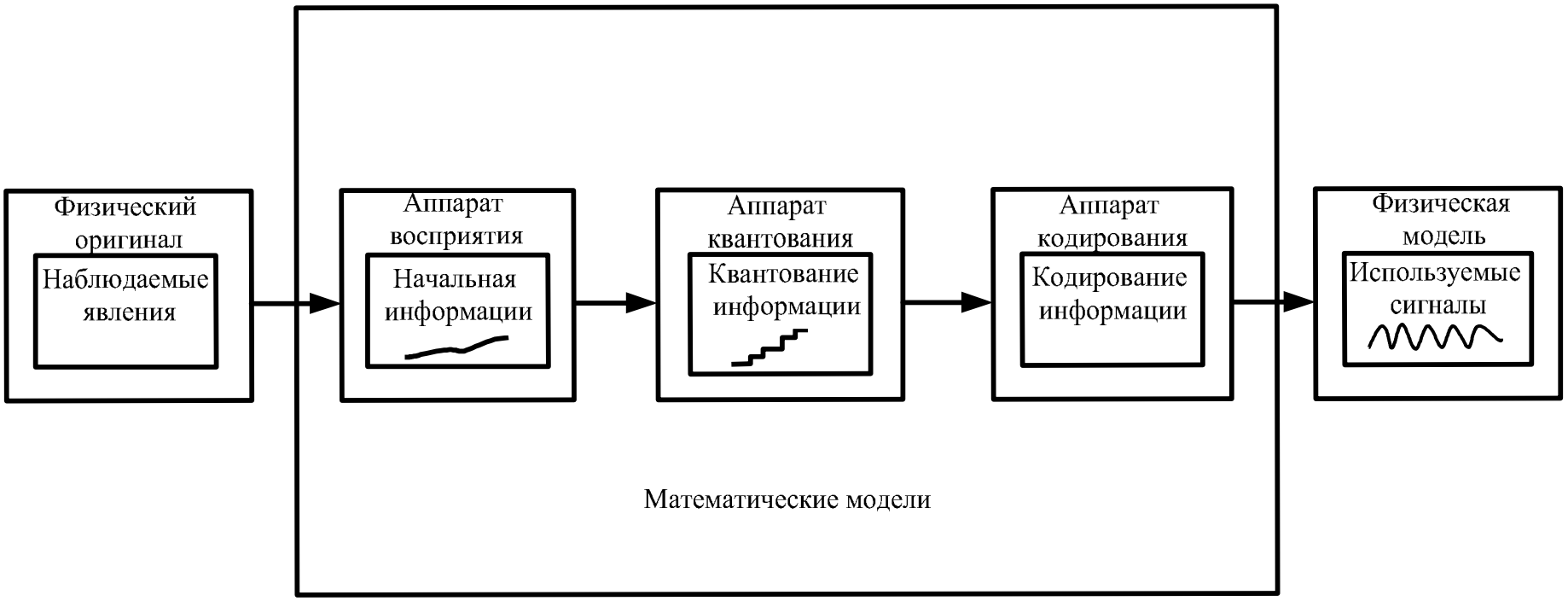


Рис. В.1. Методологическая схема образования сигнала

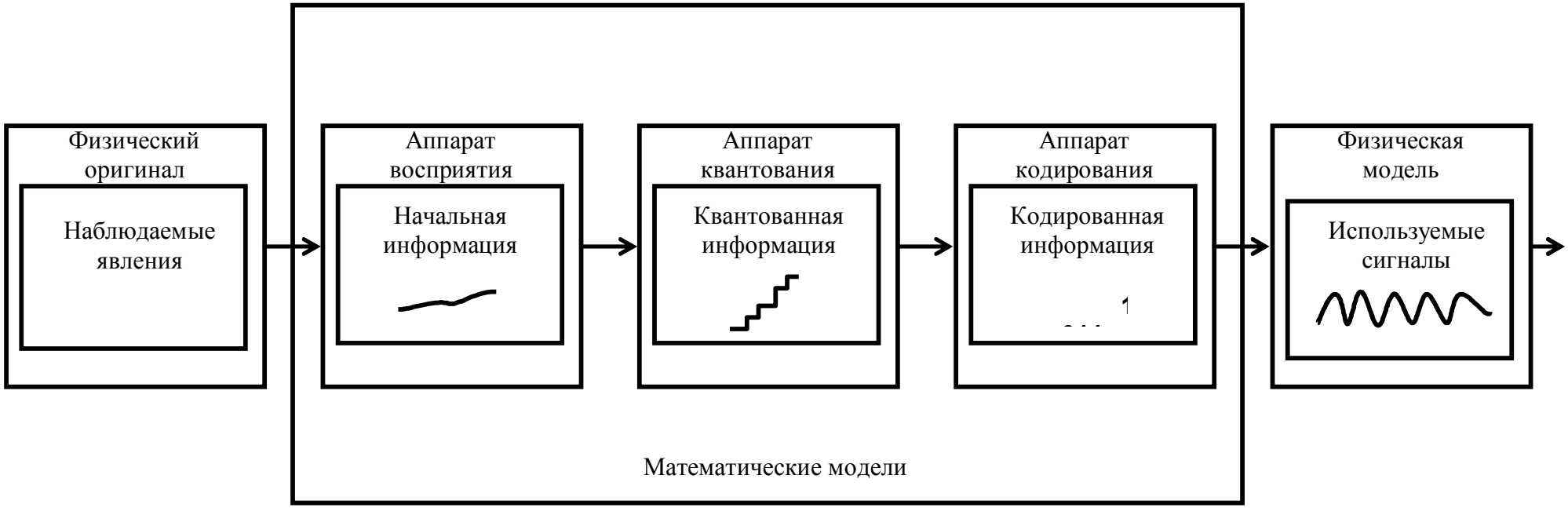


Рис. В.2. Методологическая схема образования сигнала

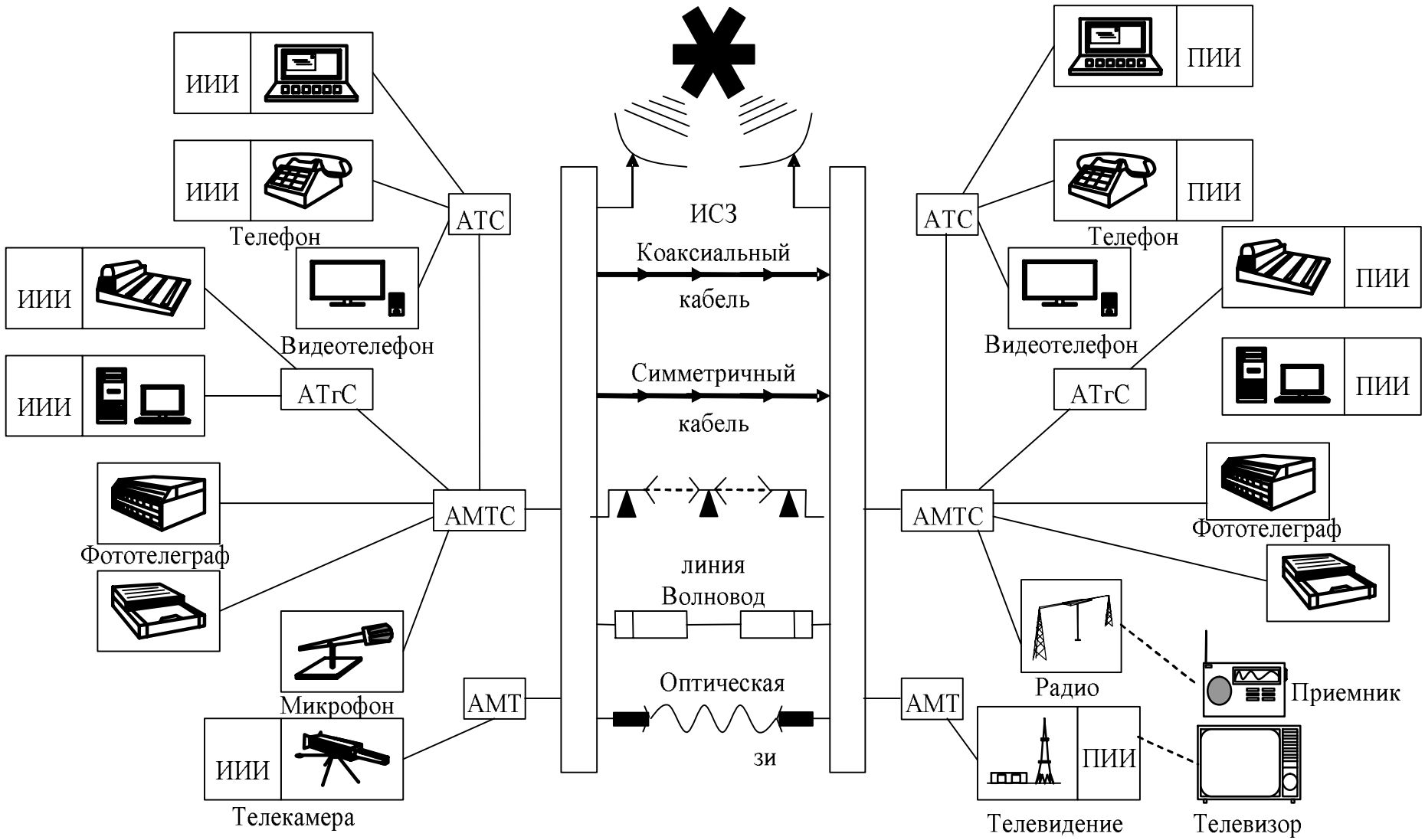


Рис. В.3. Основные виды электросвязи:
АТС - автоматическая телефонная станция; АТгС - автоматическая телеграфная станция; АМТС - аппаратная междугородней телефонной сети; АМТ - аппаратная междугороднего телевидения; ИИИ - источник измерительной информации; ПИИ - приемник измерительной информации

Непрерывные сообщения неразделимы на элементы и описываются функциями времени, принимающими непрерывное множество значений. В ряде случаев непрерывные сообщения с целью повышения качества передачи преобразуются в дискретные.

В.3. Этапы обращения информации

Информация в автоматических и автоматизированных системах используется для выработки управляющих воздействий. При этом различают этапы [1], представленные на рис. В.3. Поскольку материальным носителем информации

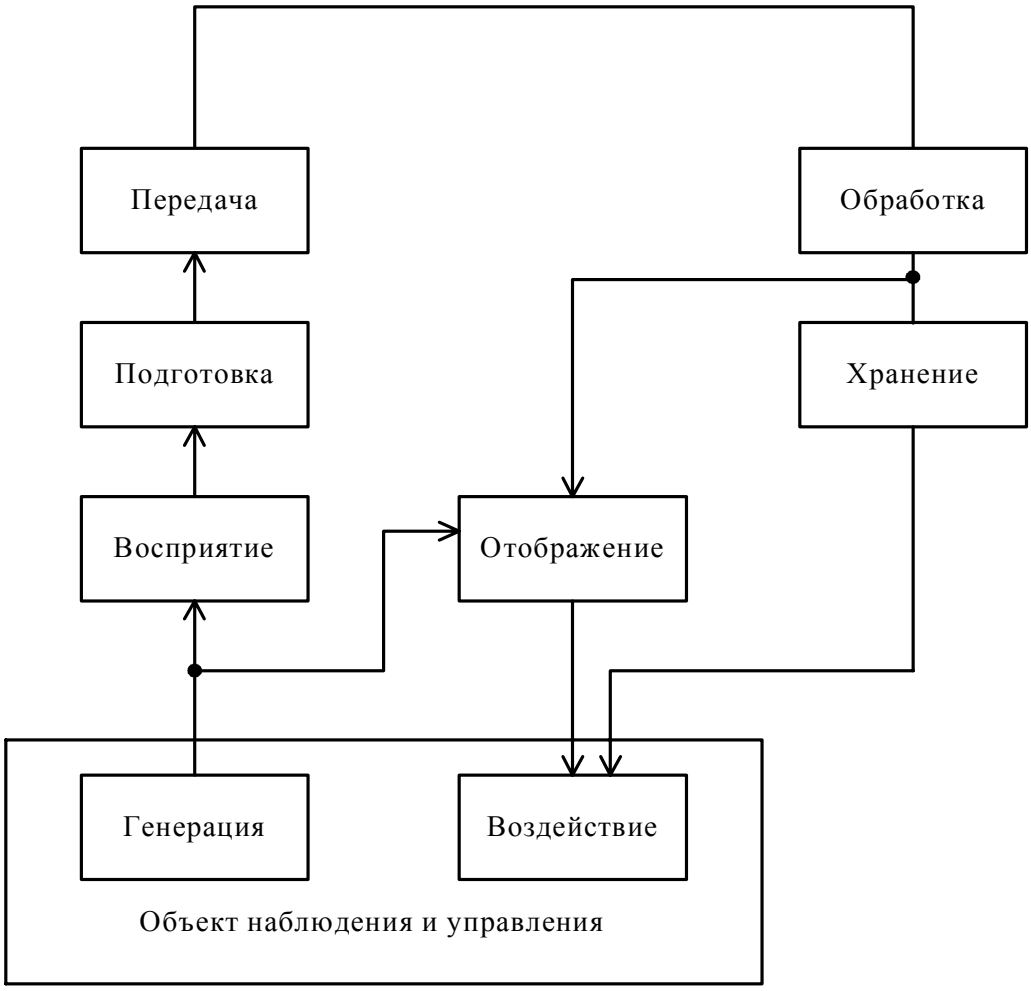


Рис. В.3. Этапы обращения информации

является сигнал, то реально это будут этапы обращения и преобразования сигналов [2].

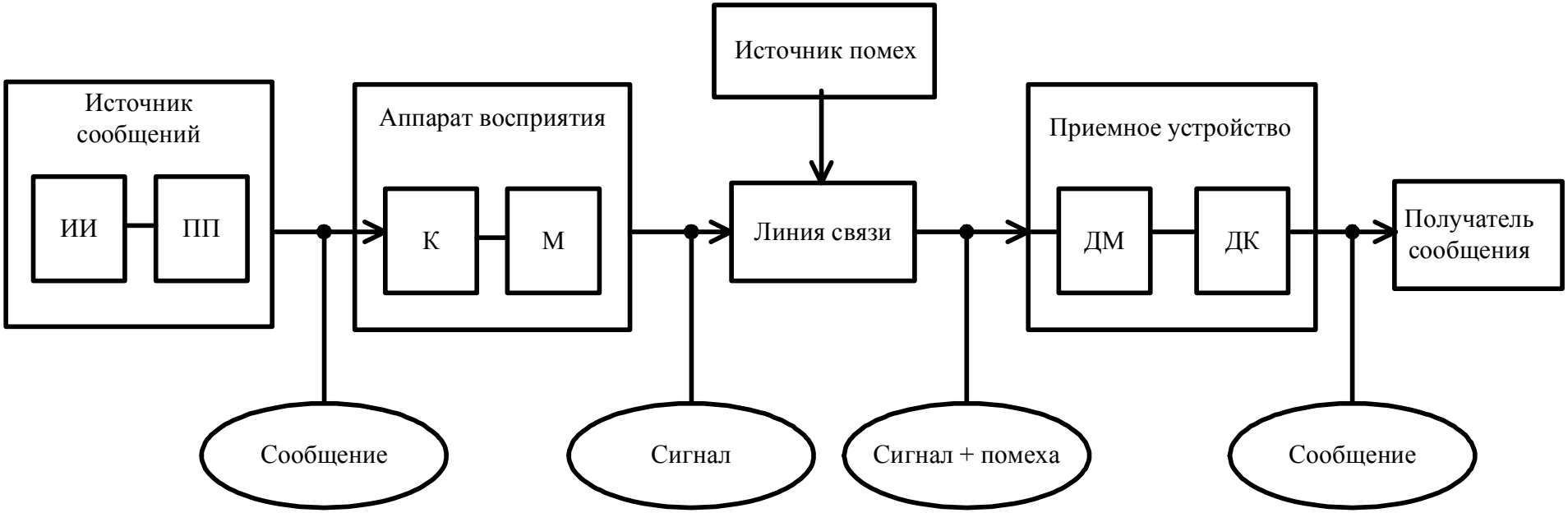


Рис. В.4. Структурная схема передачи информации

Сообщение может иметь форму, не приспособленную для передачи, хранения и обработки. В связи с этим применяют различные способы преобразования сообщения в сигнал. К ним относятся дискретизация, кодирование и модуляция.

Под кодированием понимается процесс преобразования дискретных или квантованных непрерывных сообщений в сложный дискретный сигнал, представляющий собой набор элементарных сигналов.

Под модуляцией понимается процесс изменения параметров носителя под действием сообщения.

В информационных системах под сигналом понимается физический процесс, несущий сообщение.

Операцию восстановления сообщения по принятому сигналу называют демодуляцией и декодированием, техническая реализация которых осуществляется демодулятором ДМ и декодером ДК соответственно.

Под линией связи понимают физическую среду, обеспечивающую поступление сигналов от передающего устройства к приемному. Сигналы на выходе линий связи могут отличаться от переданных вследствие затухания, искажения и воздействия помех. Помехами называют сторонние возмущения, искажающие полезный сигнал. Эффект воздействия помех на различные блоки системы стараются учесть эквивалентным изменением характеристик линии связи. Поэтому источник помех условно относят к линии связи.

Меру соответствия принятого сообщения посланному называют достоверностью передачи.

Принятое сообщение с выхода системы связи поступает к получателю.

Совокупность технических средств, предназначенных для передачи сообщений, называют каналом связи.

В.4. Уровни проблем передачи информации

Обмен информацией предполагает использование некоторой системы знаков, например, естественного или искусственного (формального) языка. В связи с этим проблемы передачи информации разделяют на проблемы синтактического, семантического и прагматического уровней.

Проблемы синтактического уровня касаются создания теоретических основ построения систем связи, основные показатели функционирования которых были бы близки к предельно возможным, а также совершенствования существующих систем с целью повышения эффективности их использования. Это чисто технические проблемы совершенствования методов передачи сообщений и сигналов. На этом уровне интересуют проблемы доставки получателю сообщений как совокупности знаков, при этом полностью абстрагируются от их смыслового и прагматического содержания.

Основу интересующей нас теории информации составляют результаты решения ряда проблем именно этого уровня. Она опирается на понятие количе-

ство информации, являющееся мерой частоты употребления знаков, которая никак не отражает ни смысла, ни важности передаваемых сообщений. В связи с этим иногда говорят, что теория информации находится на синтактическом уровне.

Проблемы семантического уровня связаны с формализацией смысла передаваемой информации, например, введением количественных оценок близости информации к истине, т.е. оценок ее качества. Эти проблемы чрезвычайно сложны, так как смысловое содержание информации больше зависит от получателя, чем от семантики сообщения, представленного в каком-либо языке.

Следует отметить, что мы еще не умеем измерять семантическую информацию. Имевшие место подходы к ее измерению пока носили весьма частный характер.

На прагматическом уровне интересуют последствия от получения и использования данной информации абонентом. Проблемы этого уровня – это проблемы эффективности. Основная сложность здесь состоит в том, что ценность или потребительская стоимость информации может быть совершенно различной для различных получателей. Кроме того, она существенно зависит от истинности информации, своевременности ее доставки и использования. В направлении количественного определения прагматического содержания информации сделаны лишь первые шаги, которые еще недостаточно конструктивны, чтобы найти широкое практическое применение. В связи с созданием информационно-вычислительных сетей ведутся интенсивные исследования в области оценки старения информации, то есть потери ее ценности в процессе доставки.

Контрольные вопросы

1. Что понимается под термином "информация"?
2. Приведите схему образования сигнала и поясните ее.
3. Дайте определение теории информации, как научной дисциплине.
4. Назовите этапы обращения информации.
5. Приведите структурную схему системы передачи информации и поясните ее.
6. Что понимается под сообщением и сигналом?
7. В чем отличие дискретных и непрерывных сообщений?
8. Что понимается под кодированием и модуляцией?
9. Дайте определение линии связи и канала связи.
10. Что понимается под достоверностью передачи?
11. Назовите уровни проблем передачи информации и дайте характеристику каждому уровню?
12. Назовите форму существования информации.

1. ОБЩИЕ СВЕДЕНИЯ О СИГНАЛАХ

1.1. Система передачи информации

В системах автоматики и телемеханики, проводной и радиосвязи сигнал передается на более или менее далекое расстояние чаще всего в виде электромагнитного возмущения. Поэтому физической величиной, определяющей характер сигнала, обычно является напряжение (или ток), изменяющееся во времени по определенному закону, отображающему передаваемое сообщение. В теоретических исследованиях сигнал, независимо от его физической природы, заменяется математическим представлением в виде некоторой функции времени, описывающей закон изменения во времени, заложенный в реальном сигнале.

Сигнал будем называть регулярным, если его математическим представлением является заранее заданная функция времени $f(t)$. Другими словами, регулярный сигнал соответствует известному сообщению.

Изучение свойств различного вида регулярных сигналов, связанных с их передачей, позволяет перейти к исследованию более сложных сигналов, имеющих характер случайных процессов.

Выражение регулярного сигнала определенной функцией времени называют временным представлением сигнала. Форма записи функции может быть различной. В частности, при некоторых ограничениях, функция времени, заданная на некотором отрезке времени, может быть представлена в виде тригонометрического ряда, каждый член которого является простейшей гармонической функцией времени (косинус, синус). Эти функции называются гармониками, и каждой из них принадлежат определенные амплитуда, частота и фаза. Множество амплитуд, частот и фаз называют спектром рассматриваемого сигнала. Функция времени находится в однозначном соответствии с принадлежащим ей спектром. На этом основании временное представление сигнала может быть заменено так называемым частотным представлением. Оба представления адекватны. Выбор того или иного представления зависит от физических и математических особенностей рассматриваемой задачи.

К основным типам регулярных сигналов относятся: периодический, почти периодический и непериодический.

Периодический сигнал представляется функцией времени, удовлетворяющей условию

$$f(t) = f(t+T), \qquad (1.1)$$

где t – любой момент времени на интервале $-\infty \le t \le +\infty$, а T – некоторая постоянная.

Наименьший конечный промежуток времени T, удовлетворяющий условию (1.1), называется периодом.

Периодический сигнал физически неосуществим, так как реальный сигнал не может продолжаться вечно; он всегда имеет начало и конец. Однако абстрактный смысл периодического сигнала не мешает его широкому использованию в теоретических исследованиях и получению результатов, соответствующих наблюдаемым в действительности. Дело в том, что регулярный сигнал, воздействующий на какое-либо устройство, можно считать существовавшим бесконечно долго, если рассматривается только установившийся режим, который не зависит от начальных условий.

Простейшим и наиболее распространенным периодическим сигналом является гармонический сигнал (рис. 1.1), выраженный косинусоидальной (или синусоидальной) функцией времени.

$$U(t) = U_m \cos(\Omega_1 t + \varphi_1), \text{ или } U(t) = U_m \sin(\Omega_1 t + \psi_1), \quad (1.2)$$

где U(t)– мгновенное значение напряжения; U_m – его амплитуда; $\Omega_1 = 2\pi/T$ – угловая частота; T – период; ψ_1 – начальная фаза; $\varphi_1 = \psi_1 - 90°$.

На рис. 1.2 показан график периодического несинусоидального напряжения, которое получается при непрерывно повторяющейся зарядке конденсатора от источника напряжения U_0 и его разрядке через активное сопротивление.

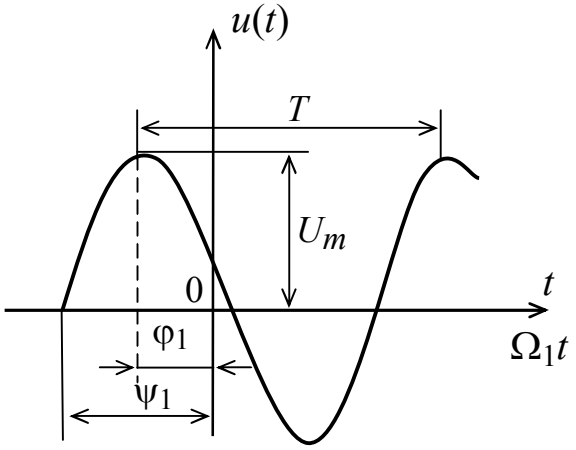


Рис. 1.1. Синусоидальное напряжение

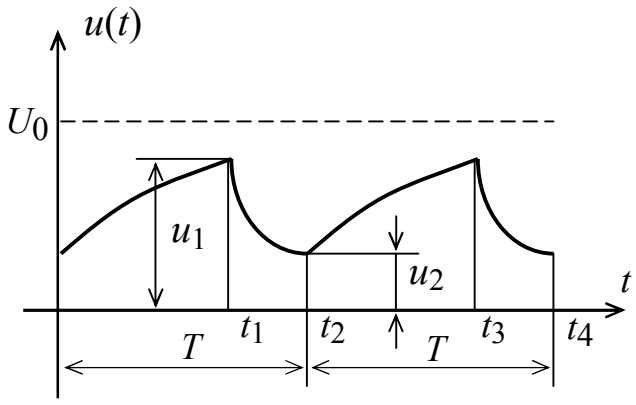


Рис. 1.2. Периодическое несинусоидальное напряжение

Функция, описывающая данный процесс, имеет вид

$$u(t) = \begin{cases} U_0 - (U_0 - U_2)e^{-\alpha_1 t} & \text{при} \quad 0 \le t \le t_1; \\ U_1 e^{-\alpha_2 (t - t_1)} & \text{при} \quad t_1 \le t \le T. \end{cases} \quad (1.3)$$

Коэффициенты α_1 и α_2 показывают скорость зарядки и разрядки и зависят от емкости конденсатора и величин активных сопротивлений цепей зарядки и разрядки.

В общем виде это напряжение, как и другие периодические функции $f(t)$, можно записать так:

$$f(t) = f(t + nT), \tag{1.4}$$

где n – любое целое положительное или отрицательное число; T – период.

В математике функция, представляемая в виде суммы гармонических составляющих с произвольными частотами, получила название **почти периодической функции**. Почти периодические функции обладают многими замечательными свойствами, и их исследованиям отведено большое место в современной теории функций. Одно из основных свойств заключается в том, что для данных функций может быть определен приближенный период (почти-период). В системах телемеханики встречаются сигналы, частоты гармоник которых не находятся в простых кратных соотношениях. Подобные сигналы называют **почти периодическими**.

Непериодическим называется регулярный сигнал, определяемый непериодической функцией, т.е. функцией, которая не удовлетворяет условию (1.1) на всем интервале времени $-\infty \le t \le +\infty$. Такой сигнал представляется функцией, заданной в пределах конечного ($t_1 \le t \le t_2$) или полубесконечного ($t_1 \le t < \infty$) промежутка времени, вне которого она принимается тождественно равной нулю. Форма сигнала может быть практически любой и, в частности, обладать периодичностью в пределах времени своего существования (например, конечный или полубесконечный отрезок синусоиды).

В зависимости от структуры информационных параметров различают сигналы:

1) непрерывные по множеству и времени, или просто непрерывные (рис.1.3,а);

2) дискретные по множеству и времени, или просто дискретные (рис. 1.3,б);

3) непрерывные по времени и дискретные по множеству (рис. 1.3,в);

4) непрерывные по множеству и дискретные по времени (рис. 1.3,г).

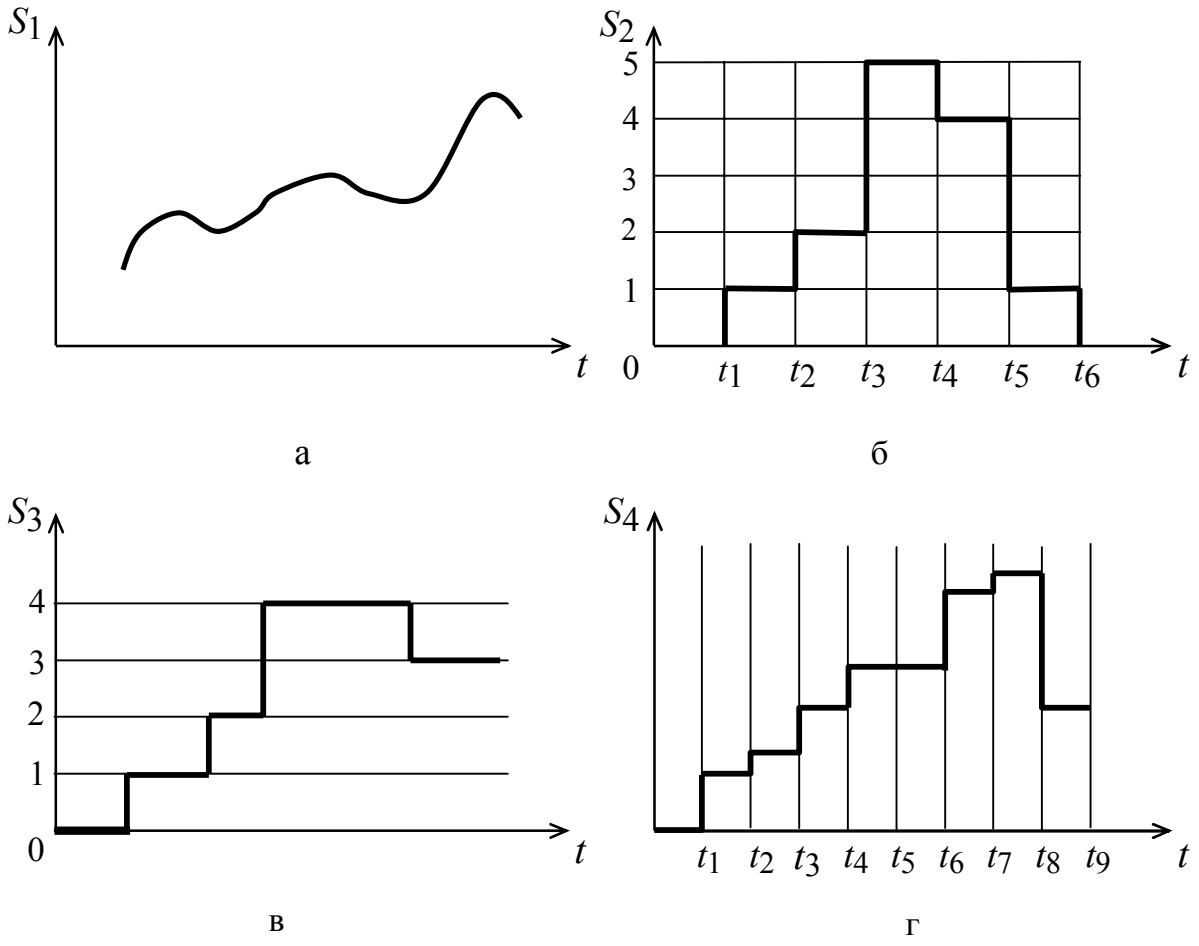


Рис. 1.3. Виды сигналов в системах телемеханики

1.2. Периодические сигналы

Представление периодического сигнала суммой гармонических составляющих осуществляется с помощью разложения в ряд Фурье функции (1.1), которая является временным представлением сигнала. Если функция *f*(t)задана на интервале времени $t_1 \le t \le t_2$ и повторяется с периодом $T=2\pi/\Omega_1 = t_2 - t_1$, то тригонометрическая форма ряда Фурье для нее может быть записана следующим образом:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos k\Omega_1 t + b_k \sin k\Omega_1 t) =$$

$$= \frac{A_0}{2} + \sum_{k=1}^{\infty} A_k \cos(k\Omega_1 t - \psi_k), \; k = 1, 2, \ldots. \qquad (1.5)$$

Амплитуды косинусоидальных и синусоидальных членов в разложении (1.5) определяются выражениями:

$$a_k = \frac{2}{T}\int_0^T f(t)\cos(k\Omega_1 t)dt\,; \tag{1.6}$$

$$b_k = \frac{2}{T}\int_0^T f(t)\sin(k\Omega_1 t)dt\,. \tag{1.7}$$

Слагаемое

$$\frac{a_0}{2} = \frac{A_0}{2} = \frac{1}{T}\int_0^T f(t)dt \tag{1.8}$$

является постоянной составляющей сигнала, которая, как это следует из (1.8), равна среднему значению функции $f(t)$за период.

Амплитуда A_k и фаза ψ_k k-й гармоники, как это следует из (1.5), связаны с величинами a_k и b_k соотношениями:

$$A_k = \sqrt{a_k^2 + b_k^2}\,,\ \ a_k = A_k\cos\psi_k\,,\ \ b_k = A_k\sin\psi_k\,;$$

$$\psi_k = \operatorname{arctg}(b_k / a_k)\,. \tag{1.9}$$

Весьма удобной является комплексная форма записи ряда Фурье, к которой легко перейти, если в разложении (1.5) выразить тригонометрические функции через показательные, воспользовавшись известными формулами:

$$\cos k\Omega_1 t = \frac{1}{2}(e^{jk\Omega_1 t} + e^{-jk\Omega_1 t})\,;\ \sin k\Omega_1 t = \frac{1}{2j}(e^{jk\Omega_1 t} - e^{-jk\Omega_1 t})\,. \tag{1.10}$$

В результате получим

$$f(t) = \frac{a_0}{2} + \frac{1}{2}\sum_{k=1}^{\infty}(A_k e^{jk\Omega_1 t} + \overset{*}{A}_k e^{-jk\Omega_1 t})\,, \tag{1.11}$$

где $\dot{A}_k$ и $\overset{*}{A}_k$ – комплексные амплитуды, связанные с a_k и b_k соотношениями

$$\dot{A}_k = A_k e^{-j\psi_k} = a_k - jb_k\,, \tag{1.12}$$

$$\overset{*}{A}_k = A_k e^{j\psi_k} = a_k + jb_k\,. \tag{1.13}$$

Таким образом, комплексные амплитуды $\dot{A}_k$ и $\overset{*}{A}_k$ являются комплексно-сопряженными величинами. Действительно, каждое слагаемое первого ряда в

выражении (1.11) можно представить как вектор на комплексной плоскости (рис. 1.4), вращающийся с частотой $k\Omega_1$ (т.е. в положительном направлении отсчета углов – против направления движения часовой стрелки). Каждое слагаемое второго ряда – вектор, вращающийся в обратном направлении.

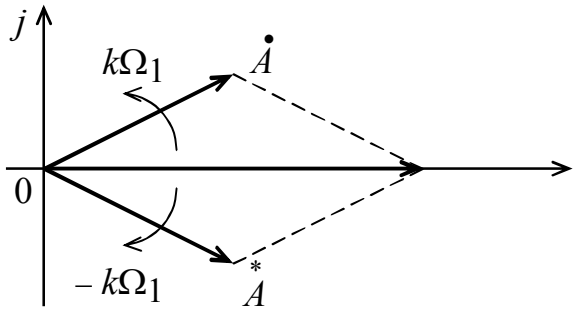


Рис. 1.4. Векторная диаграмма комплексно-сопряженных величин

Так как $\dot{A}_k$ и $\dot{A}_k^*$ – комплексно-сопряженные величины, то сумма векторов в любой момент времени дает вектор, направленный по вещественной оси, т.е. k-ю гармоническую составляющую вещественной функции времени $f(t)$. Отрицательная частота – $k\Omega_1$ только указывает направление вращения вектора.

Комплексная амплитуда $\dot{A}_k$ определяется по выражению

$$\dot{A}_k = \frac{1}{T}\int_0^T f(t)e^{-jk\Omega_1 t}dt = \frac{\Omega_1}{2\pi}\int_0^T f(t)e^{-jk\Omega_1 t}dt. \qquad (1.14)$$

При $k = 0$

$$\frac{A_0}{2} = \frac{1}{T}\int_0^T f(t)dt = \frac{\Omega_1}{2\pi}\int_0^T f(t)dt = \frac{a_0}{2}. \qquad (1.15)$$

Тогда выражение (1.11) можно переписать в виде

$$f(t) = \frac{1}{2}\sum_{k=-\infty}^{k=\infty}\dot{A}_k\, e^{jk\Omega_1 t}. \qquad (1.16)$$

При такой записи ряда Фурье периодический сигнал заменяется суммой простых гармонических колебаний как с положительными частотами ($k>0$), так и с отрицательными ($k<0$). Конечно, отрицательные частоты не имеют здесь физического смысла, а являются формальным следствием произведенного математического преобразования.

1.3. Спектры периодических сигналов и необходимая ширина полосы частот

1.3.1. Дискретный спектр.

Представить сигнал с заданным периодом T рядом Фурье – это значит найти амплитуды и начальные фазы всех его гармонических составляющих. Совокупность амплитуд называют спектром амплитуд, а совокупность начальных фаз – спектром фаз. Во многих частных случаях достаточно рассчитать только спектр амплитуд сигнала, который для краткости назовем просто спектром.

Определим спектр периодической последовательности прямоугольных импульсов (рис. 1.5) длительностью τ и с периодом T. Напряжение такой формы действует в каналах связи и часто рассматривается как основной периодический сигнал при исследовании передачи информации по линии связи.

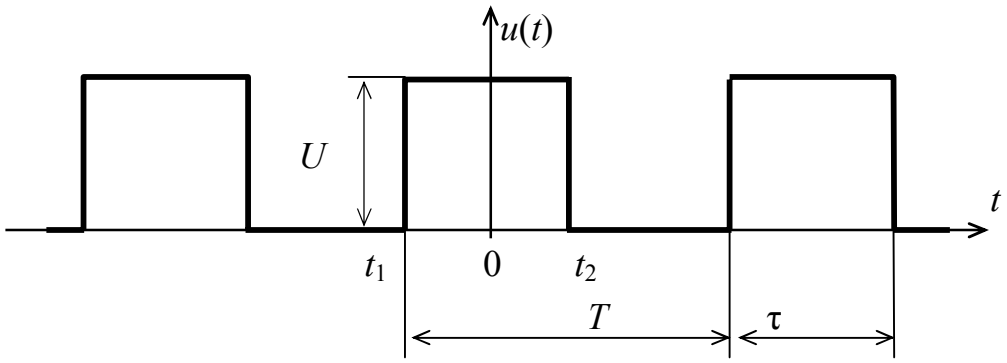


Рис. 1.5. Периодическая последовательность прямоугольных импульсов

Для такого сигнала по формулам (1.6) – (1.8)

$$\frac{a_0}{2}=\frac{A_0}{2}=\frac{1}{T}\int_{-\tau/2}^{\tau/2}Udt=U\frac{\tau}{T};\ a_k=\frac{2}{T}\int_{-\tau/2}^{\tau/2}U\cos k\Omega_1 t dt=\frac{2U}{k\pi}\sin k\frac{\tau}{T}\pi;$$

$b_k=0$, т.е. $\psi_k=0$ или π и $A_k=|a_k|$.

Следовательно, напряжение можно представить рядом Фурье

$$u(t)=U(\frac{\tau}{T}+\frac{2}{\pi}(\sin\frac{\tau}{T}\pi\cos\Omega_1 t+\frac{1}{2}\sin 2\frac{\tau}{T}\pi\cos 2\Omega_1 t+$$

$$+\frac{1}{3}\sin 3\frac{\tau}{T}\pi\cos 3\Omega_1 t+\ldots))=U\frac{\tau}{T}(1+\sum_{k=1}^{\infty}2\frac{\sin k\Omega_1\tau/2}{k\Omega_1\tau/2}\cos k\Omega_1 t). \quad (1.17)$$

Спектр амплитуд сигнала изображают в виде спектральных линий, длины которых пропорциональны амплитудам гармоник (рис. 1.6).

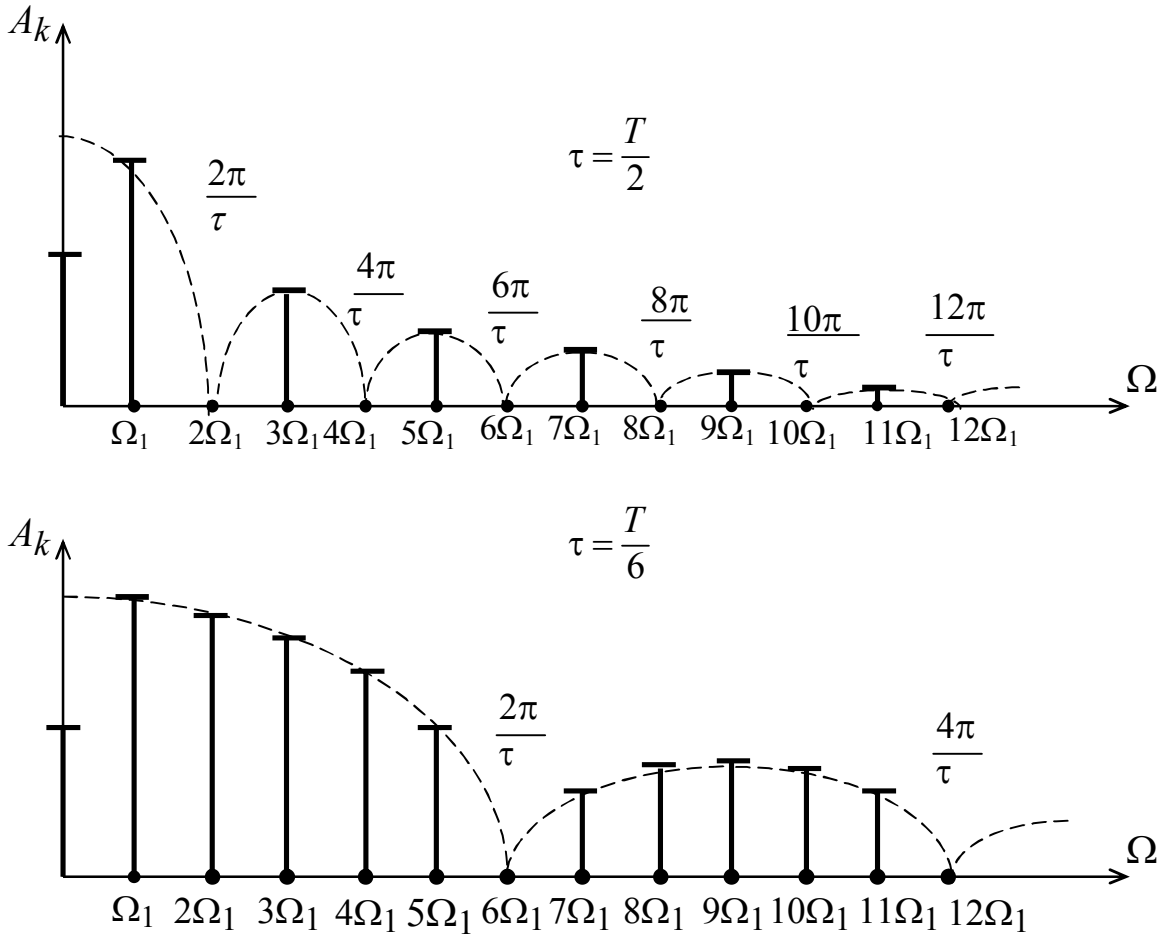


Рис. 1.6. Спектры периодически повторяющихся прямоугольных импульсов при Q=2 и Q=6

Такой спектр называют линейчатым или дискретным. Спектр фаз ψ_k также линейчатый, причем в рассматриваемом частном случае ψ_k может иметь только два значения: 0 или π.

Непрерывная кривая, соединяющая концы линий спектра и показанная на рис. 1.5 пунктиром, носит название огибающей спектра амплитуд, которая определяется уравнением

$$A(\Omega) = \frac{2U\tau}{T}\left|\frac{\sin(\Omega\tau/2)}{\Omega\tau/2}\right|, \qquad (1.18)$$

где $\Omega = k\Omega_1$ для k-й гармоники.

Выражение для фазы гармоники можно записать в виде

$$\psi_k = k\Omega_1(t_1 + \tau/2) + (k-1)\pi\,. \qquad (1.19)$$

На рис. 1.7 приведены спектры фаз и их огибающие при различно выбранных началах отсчета времени. Наиболее простым получается спектр фаз при $t_1 = -\tau/2$.

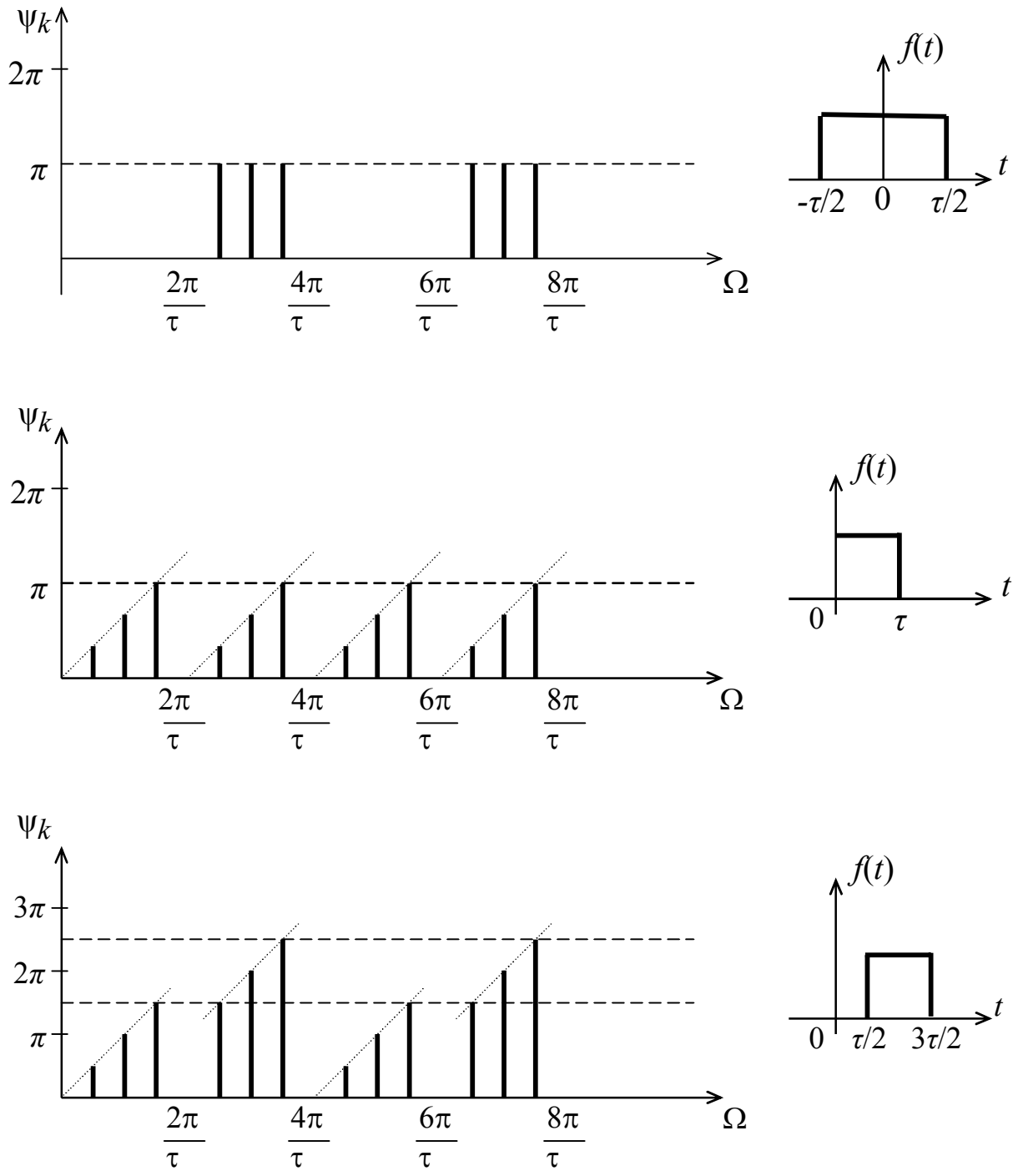


Рис. 1.7. Спектры фаз при различных началах отсчета времени

Кроме того, из (1.17) и рис. 1.6 следует, что периодическую последовательность прямоугольных импульсов можно рассматривать как результат наложения друг на друга бесконечного количества гармоник с частотами, кратными основной частоте $\Omega_1 = 2\pi/T$, а также постоянной составляющей. Амплитуды гармонических составляющих кратных скважности Q равны нулю (например, равны нулю амплитуды четных гармоник на рис. 1.6, где принято $\tau = T/2$, и шестая, двенадцатая и т.д., где принято $\tau = T/6$).

С изменениями длительности импульса τ при том же периоде следования импульсов T или с изменением периода T при постоянной длительности τ спектр существенно преобразуется. Если длительность импульса растет, то увеличивается удельный вес постоянной составляющей и гармоник с небольшими порядковыми номерами, а удельный вес высших гармоник падает. Если, наоборот, уменьшить длительность импульса τ, то удельный вес гармоник с небольшим порядковым номером уменьшается, а удельный вес высших гармоник растет.

При изменении не длительности импульсов τ, а периода их повторения T спектр амплитуд становится реже или гуще. Так, с увеличением периода T основная частота уменьшается ($\Omega_1 = 2\pi/T$) и спектр становится гуще.

1.3.2. Практическая ширина спектра.

Теоретически, как указывалось выше, для большинства периодических функций спектр неограничен, т.е. для передачи сигналов телемеханики без изменения формы необходимы бесконечно большая полоса пропускания канала связи и отсутствие амплитудных и фазовых искажений. Практически все каналы связи имеют ограниченную полосу пропускания, и форма сигналов при передаче по каналу изменяется даже при отсутствии в этой полосе амплитудных и фазовых искажений. Очевидно, важно передать ту часть спектра сигнала, которая содержит гармонические составляющие с относительно большими амплитудами. В связи с этим вводится понятие практической ширины спектра сигнала. Под практической шириной спектра сигнала понимается та область частот, в пределах которой лежат гармонические составляющие сигнала с амплитудами, превышающими наперед заданную величину.

Поскольку средняя мощность, выделяемая сигналом на активном сопротивлении, равном 1 Ом, складывается из мощностей, выделяемых на этом сопротивлении гармоническими составляющими,

$$P_{cp} = \frac{A_0^2}{4} + \sum_{k=1}^{\infty} \frac{A_k^2}{2}, \qquad (1.20)$$

практическая ширина спектра с энергетической точки зрения может быть определена как область частот, в пределах которой сосредоточена подавляющая часть мощности сигнала.

В качестве примера определим практическую ширину спектра периодической последовательности прямоугольных импульсов (рис. 1.8,а), если требуется учесть все гармонические составляющие сигнала, амплитуды которых более 0,2 от амплитуды первой гармоники. Число подлежащих учету гармоник k может быть получено из выражения

$$\frac{A_k}{A_1} = \frac{2U}{k\pi} \cdot \frac{\pi}{2U} = \frac{1}{k} = 0{,}2 \ ,$$

откуда k=5.

Таким образом, практическая ширина спектра в рассмотренном примере оказывается равной $5\Omega_1$, в ней размещаются всего три гармоники (первая, третья и пятая) и постоянная составляющая.

Средняя мощность P_{k5}, выделяемая в активном сопротивлении, равном 1 Ом, перечисленными составляющими, равна

$$P_{k5} = \frac{U^2}{4} + \frac{1}{2}\left(\frac{2U}{\pi}\right)^2 + \frac{1}{2}\left(\frac{2U}{3\pi}\right)^2 + \frac{1}{2}\left(\frac{2U}{5\pi}\right)^2 \cong 0{,}48 \ \ U^2 .$$

Средняя мощность, выделяемая в этом же сопротивлении всеми составляющими сигнала, будет

$$P_{k\Sigma} = P_{имп} \Big/ \mathrm{Q} = 0{,}5 \ \ U^2 .$$

Таким образом, $(P_{k5} / P_{k\Sigma}) \cdot 100 = 96\%$, т.е. составляющие, входящие в практический спектр, выделяют в активном сопротивлении 96 % всей мощности сигнала.

Очевидно, расширение практического спектра данного сигнала (свыше $5\Omega_1$) с энергетической точки зрения нецелесообразно.

Ограничение спектра сигнала оказывает также влияние на его форму. Для иллюстрации на рис. 1.8 показано изменение формы прямоугольных импульсов при сохранении в спектре только постоянной составляющей и первой гармоники (рис. 1.8, б), при ограничении спектра частотой $3\Omega_1$ (рис. 1.8,в) и при ограничении спектра частотой $5\Omega_1$ (рис. 1.8,г). Как следует из рисунка, чем круче должен быть фронт импульса, тем большее число высших гармонических составляющих должно входить в состав сигнала.

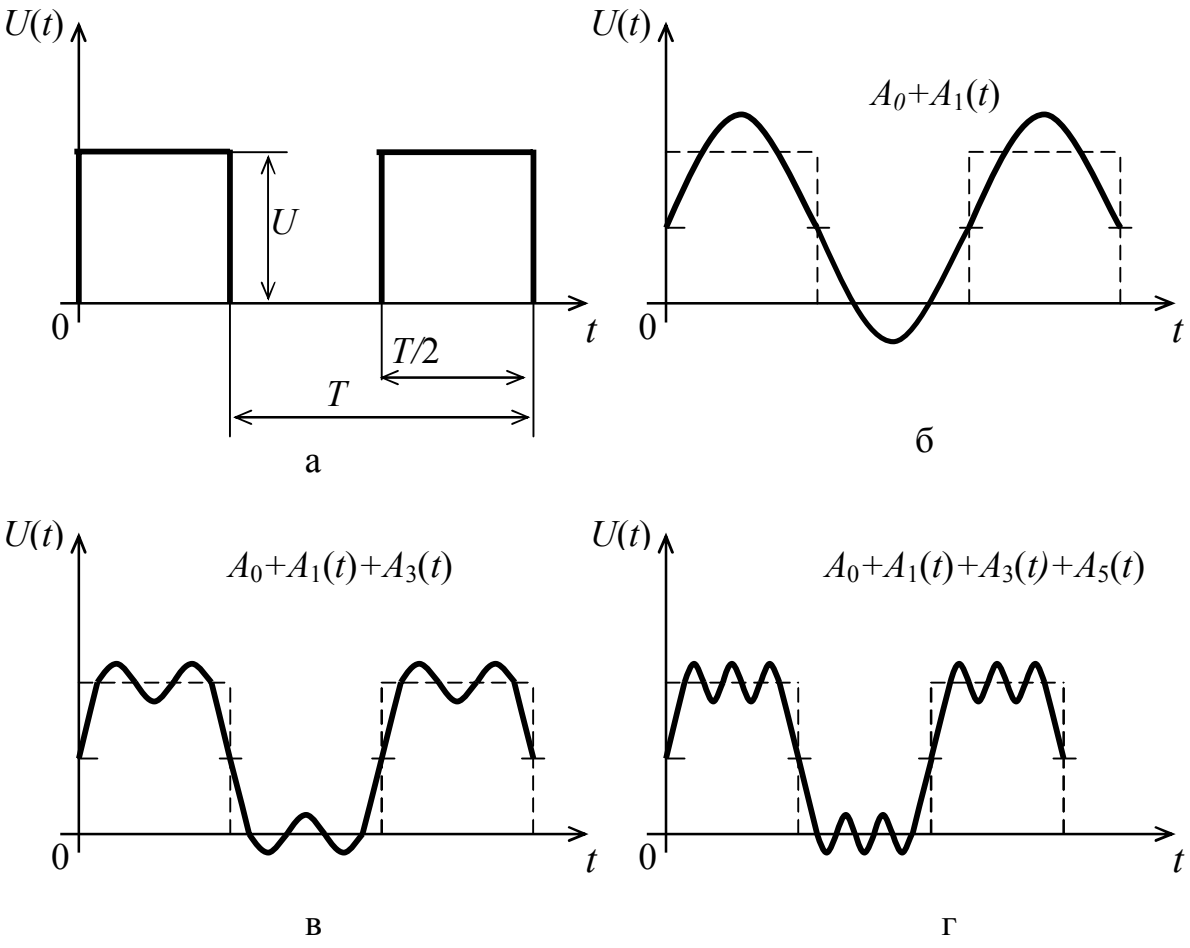


Рис. 1.8. Формы сигнала при ограничении спектра последовательности прямоугольных импульсов

Рассмотренная зависимость формы периодического сигнала от количества суммируемых гармоник показывает, что при выборе практической ширины спектра сигнала нельзя ограничиваться только энергетическими соображениями. Необходимо учитывать требования к сигналу на выходе системы, как с энергетической точки зрения, так и с точки зрения сохранения его формы. В общем случае практическая ширина спектра сигнала выбирается из условия

$$\Delta\omega = 2\pi\mu/\tau, \qquad (1.21)$$

где $\mu = 0{,}5\ldots 2$ – коэффициент формы импульса; при $\mu = 1$ обеспечивается передача около 90% всей энергии сигнала.

В кодоимпульсных системах телеизмерения, а также во многих системах телеуправления каждая кодовая комбинация состоит из определенной последовательности прямоугольных импульсов и пауз. Кодовая комбинация, соответствующая данной величине измеряемого параметра или команде, может периодически передаваться по каналу связи. Спектр такого сигнала зависит, конечно, от того какая именно кодовая комбинация передается. Но самым главным фактором, определяющим удельный вес высших гармоник спектра, остается

наибольшая частота следования импульсов. Поэтому и для кодоимпульсных систем при определении практически необходимой ширины полосы частот выбирают сигнал в виде периодической последовательности прямоугольных импульсов (рис. 1.5). Параметр τ выбирают равным длительности самого короткого импульса среди всех встречающихся в кодовых комбинациях, период следования T=2τ. В этом случае наибольшая частота следования импульсов $\Omega_{max} = 2\pi / T$ и частота основной гармоники спектра $\Omega_1 = \Omega_{max}$. Необходимая ширина полосы частот сигнала определяется дискретным спектром с ограниченным числом составляющих и в соответствии с выражением (1.21).

Характер спектра, определяющий требуемую полосу частот, зависит не только от вида сигнала, но и от условий, существующих в тракте передачи. Если переходные процессы, возникающие в системе при передаче одного импульса, заканчиваются до момента возникновения следующего импульса, то вместо периодической последовательности импульсов можно рассматривать передачу независимых одиночных импульсов.

1.4. Спектр одиночного прямоугольного импульса

Одиночный импульс можно рассматривать как непериодический сигнал, так как не существует конечного интервала времени T, отвечающего условию

$$f(t) = f(t + nT). \tag{1.22}$$

Наиболее просто и наглядно спектр непериодического сигнала можно получить из спектра периодического сигнала (1.16), принимая, что период T стремится к бесконечности, т.е. путем предельного перехода от ряда Фурье к интегралу Фурье

$$S(\Omega) = \int_{-\infty}^{\infty} f(t) e^{-j\Omega t} dt. \tag{1.23}$$

Величину $S(\Omega)$ называют спектральной функцией или просто спектральной плотностью.

Рассчитаем спектральную плотность одиночного прямоугольного импульса длительностью τ (рис. 1.9).

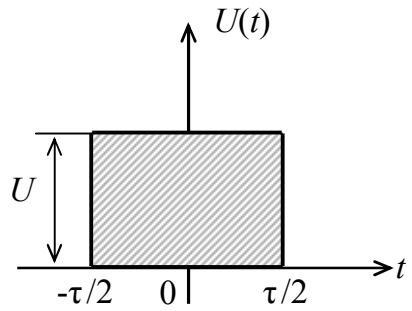
Согласно (1.23)

$$S(\Omega) = \int_{-\tau/2}^{\tau/2} U e^{-j\Omega t} dt = \frac{2U}{\Omega} \sin \Omega\tau/2. \tag{1.24}$$

Последнее выражение может быть представлено в несколько ином виде:

$$S(\Omega) = 2U \frac{\tau}{2} \cdot \frac{\sin \Omega\tau/2}{\Omega\tau/2} = U\tau \frac{\sin \Omega\tau/2}{\Omega\tau/2}. \tag{1.25}$$

Здесь текущая частота Ω может принимать любые значения от нулевой до бесконечно большой (сплошной спектр). График для $S(\Omega)$ приведен на рис.1.10.



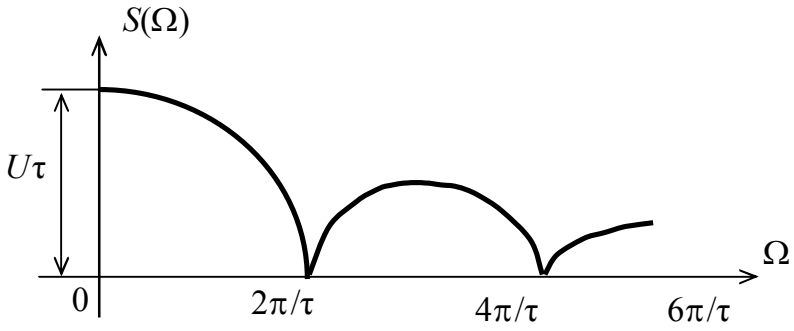


Рис. 1.9. Прямоугольный импульс Рис. 1.10. Спектр амплитуд прямоугольного импульса

При частотах $\Omega = 2k\pi/\tau$ (k = 1, 2, 3,…) спектральная плотность $S(\Omega) = 0$. Учитывая характер распределения $S(\Omega)$, можно отметить, что требуемая полоса частот вполне определяется спектром в пределах первого (k =1) нулевого значения спектральной плотности. При этом $\Omega = 2\pi/\tau = 2\pi F$, где $F=1/\tau$. Таким образом, для непериодического сигнала необходимая полоса частот может быть найдена из уравнения

$$F\tau = 1. \qquad (1.26)$$

Данный вывод вытекает и из того, что энергия непериодического сигнала пропорциональна интегралу от квадрата спектральной плотности

$$W = \frac{1}{\pi}\int_0^{\infty} |S(\Omega)|^2 \, d\Omega. \qquad (1.27)$$

Если спектр сигнала ограничивается частотой Ω_{max}, то энергия уменьшается до значения

$$W_\Omega = \frac{1}{\pi}\int_0^{\Omega_{max}} |S(\Omega)|^2 \, d\Omega. \qquad (1.28)$$

Зависимость энергии W_Ω от наибольшей частоты ограничения Ω_{max} спектра прямоугольного импульса показана на рис. 1.11.

Из рис. 1.10 и 1.11 следует, что наибольшее энергетическое значение имеют составляющие низкочастотной части спектра импульса. С ростом ширины сохраняемой части спектра от нуля до величины $\Omega_{max} = 2\pi/\tau$ энергия W_Ω быстро увеличивается и достигает 90 % всей энергии W. При дальнейшем увеличении спектра энергия W_Ω нарастает все медленнее. Таким образом, при ширине спектра $\Omega_{max} = 2\pi/\tau$ или $F=1/\tau$ обеспечивается передача значительной части энергии сигнала. Чем короче импульс, тем более широкий спектр должен быть сохранен.

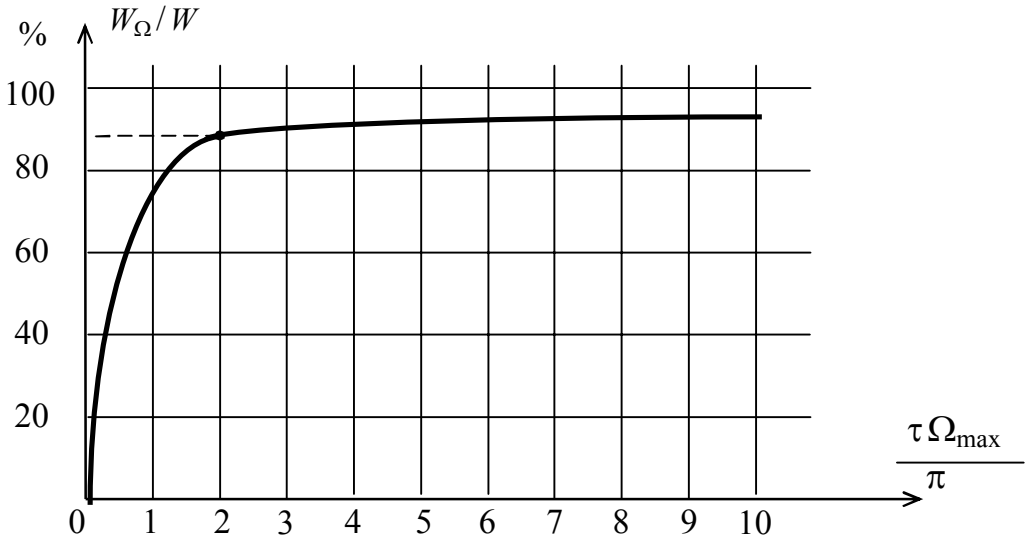


Рис. 1.11. Зависимость энергии импульса от ширины сохраняемой части спектра

Итак, мы рассмотрели как сообщения (первичные сигналы), с которыми приходится иметь дело в телемеханике, так и переносчики, с помощью которых они передаются. Прежде чем переходить к изучению методов образования сигналов, остановимся на некоторых вопросах преобразования непрерывных сообщений в дискретные. Такое преобразование имеет место в цифровых телеизмерительных системах, в системах связи при передаче речи, музыки, телевизионных изображениях и т.п.

1.5. Преобразование непрерывных сообщений в дискретные сигналы

1.5.1. Квантование по времени (дискретизация).

Непрерывные сообщения представляют собой непрерывные функции времени с бесконечным числом промежуточных точек. Для передачи таких сообщений без погрешности необходим канал связи с бесконечной пропускной способностью. На практике всегда передача сообщений осуществляется с ограниченными спектром частот и точностью, так как все каналы имеют ограниченную пропускную способность.

Если непрерывное сообщение имеет ограниченный спектр частот, оно всегда может быть передано своими значениями в отдельные моменты времени, т.е. может быть превращено в дискретное во времени сообщение, состоящее из последовательного во времени ряда значений.

Возможность такой замены была впервые установлена и сформулирована в 1933 г. В. А. Котельниковым в виде следующей теоремы: «Если функция $f(t)$ не содержит частот выше F_{max} Гц, то она полностью определяется своими мгновенными значениями в моменты времени, отстоящие друг от друга на $1/2F_{max}$», т. е.

$$\Delta t \le 1/2F_{max} . \qquad (1.29)$$

Функцию с ограниченным спектром можно записать в виде тригонометрического ряда

$$f(t) = \sum_{k=-\infty}^{\infty} f(k\Delta t)\frac{\sin 2\pi F_{\max}(t - k\Delta t)}{2\pi F_{\max}(t - k\Delta t)}, \qquad (1.30)$$

где k – порядковый номер отсчета функции.

При этом функция вполне определяется своими мгновенными значениями $f(k\Delta t)$, отсчитанными через равные интервалы времени Δt, называемые интервалами дискретизации (рис. 1.12).

Свойства ряда (1.30) основываются на свойстве функции $(\sin x)/x$, равной 1 при x=0 и равной 0 при x, кратных π (180, 360, 540° и т.д.).

Физический смысл преобразования состоит в том, что каждый член ряда (1.30) представляет собой отклик идеального фильтра нижних частот с граничной частотой среза $F_{\max}$ на очень короткий импульс, возникающий в момент времени $k\Delta t$ (рис. 1.12) и имеющий площадь, равную мгновенному значению функции $f(t)$.

Интересным свойством ряда (1.30) является то, что значения ряда в момент $k\Delta t$ определяются только k-м членом ряда, так как все другие члены в этот момент времени обращаются в нуль:

$$\frac{\sin 2\pi F_{\max}(t - k\Delta t)}{2\pi F_{\max}(t - k\Delta t)} = \begin{cases} 1 & \text{при } t = k\Delta t \\ 0 & \text{при } t = i\Delta t (i \neq k). \end{cases} \qquad (1.31)$$

Следовательно, несмотря на то, что выходные функции перекрываются, значением заданной функции в момент отсчета является только одно из ее значений.

Согласно теореме Котельникова для однозначного представления функции с ограниченным спектром на интервале времени T достаточно иметь N значений этой функции, т.е.

$$N = T/\Delta t = 2F_{\max}T. \qquad (1.32)$$

Аналогичные результаты можно получить для функций со спектром частот в промежутке от F_1 до F_2.

Таким образом, непрерывное сообщение сводится к сигналу в виде последовательности импульсов, амплитуда которых равна значению исходной функции, передаваемой в дискретные моменты времени $k\Delta t$, а интервалы между ними Δt=1/2$F_{\max}$.

При выполнении условий (1.29) непрерывная и дискретная во времени функции обратимы между собой (тождественны).

Для преобразования дискретной функции в непрерывную нужно включить идеальный фильтр частот с частотой среза равной F_{max}.

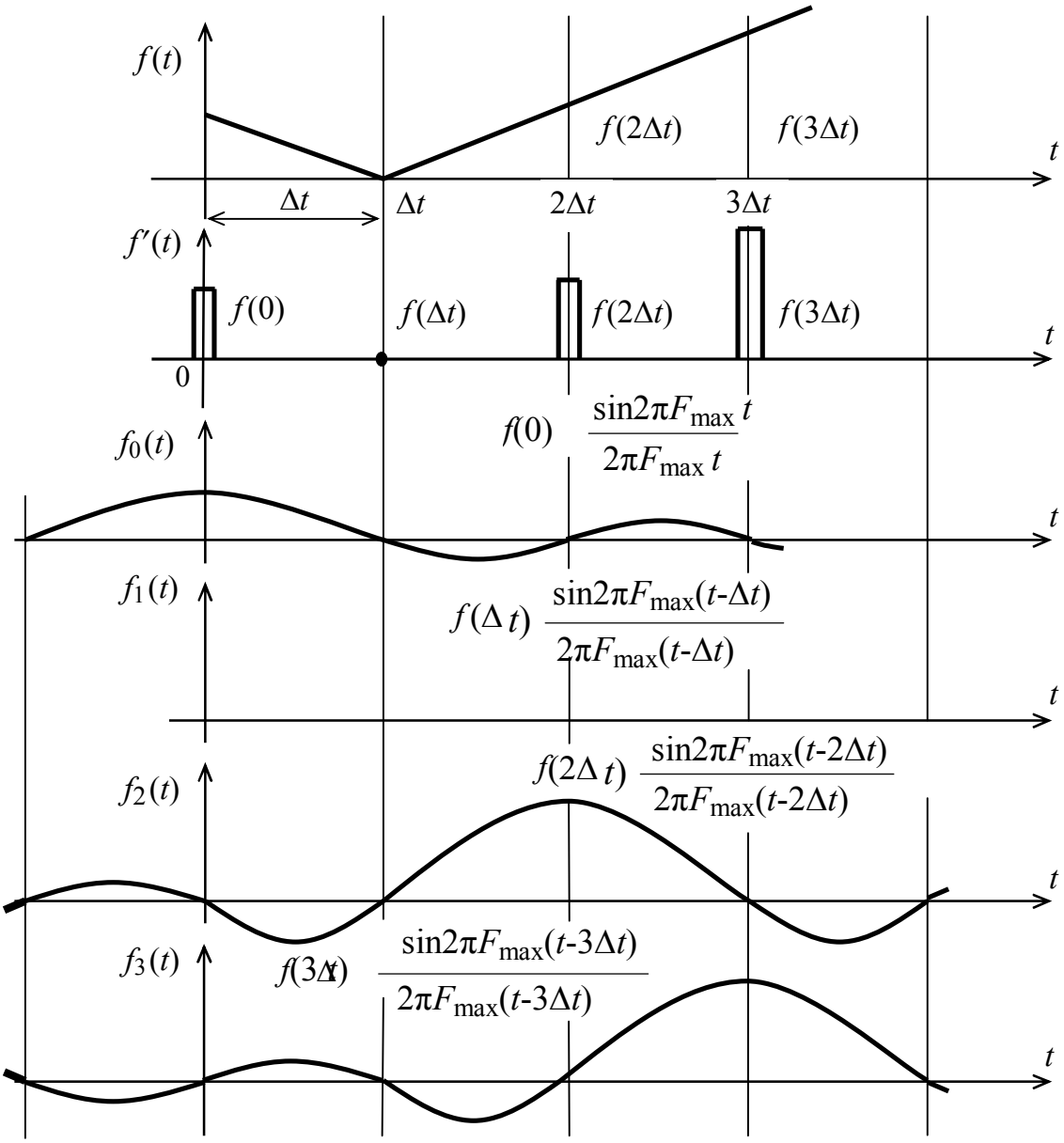


Рис. 1.12. Разложение функции $f(t)$ с ограниченным спектром частот по В.А.Котельникову

Рассмотренный процесс преобразования непрерывного сообщения в дискретный во времени сигнал называется дискретизацией во времени.

В заключение следует отметить, что при определении на практике интервала дискретизации теорему Котельникова можно применять с поправкой

$$\Delta t \approx 1/(\eta 2F_{max}), \qquad (1.33)$$

где η – коэффициент, зависящий от точности воспроизведения функции и способа интерполяции; при линейной интерполяции $\eta_л = 0{,}75/\sqrt{\delta_{отн}}$, при ступенчатой $\eta_{ст} = (3 \div 5)\eta_л$ (относительная погрешность воспроизведения).

1.5.2. Дискретизация двумерной функции.

Все большую часть передаваемых по линии связи сообщений, составляют сигналы, являющиеся функциями не только времени - *λ(t)* (речь, музыка и т.п.), но и ряда других переменных, например, *λ(x,y), λ(x,y,t)* (статические и динамические изображения, карты физических полей и т.п.). В связи с этим естественным является вопрос: можно ли так, как это делается для временных сигналов (или других функций одной переменной), производить дискретизацию многомерных сигналов (функций нескольких переменных)?

Ответ на этот вопрос дает теорема дискретизации для двумерных (или в общем случае - для многомерных) сигналов, которая утверждает: функция двух переменных λ(x,y), двумерное преобразование Фурье которой

$$FF\{\lambda(x,y)\} = \int_{-\infty}^{+\infty}\int \lambda(x,y) * \exp(-j2\pi f_x * x) * \exp(-j2\pi f_y * y)dxdy \qquad (1.34)$$

равно нулю при $f_x \geq f_x\,max$ и $f_y \geq f_y\,max$, однозначно определяется своими значениями в равноотстоящих точках плоскости переменных **x** и **y**, если интервал дискретизации удовлетворяет условию $\Delta x \leq 1/2f_x$ **max** , $\Delta y \leq 1/2f_y$ **.** Процедура дискретизации двумерной функции иллюстрируется примером, приведенным на рис. 1.13.

Доказательство двумерной теоремы дискретизации основано, так же как и для одномерного случая, на однозначном соответствии между сигналами и их спектрами: одинаковым изображениям (двумерным функциям) соответствуют одинаковые спектры, и наоборот, если спектры двух функций одинаковы, то и сами эти функции равны друг другу.

Преобразование Фурье (спектр) дискретизованной двумерной функции $FF\{\lambda(i\Delta x,j\Delta y)\}$ получается периодическим продолжением спектра исходной непрерывной функции $\lambda\,(x,y)$ в точки частотной плоскости $(k\Delta f_x, l\Delta f_y)$ (рис. 1.14), где f_x и f_y - так называемые "пространственные частоты", являющиеся аналогами обычной "временной" частоты и отражающие скорость изменения двумерной функции $\lambda\,(x,y)$ по соответствующим координатам (крупные фрагменты изображения - низкие частоты, мелкие детали - высокие частоты).

Аналитически это можно записать следующим образом:

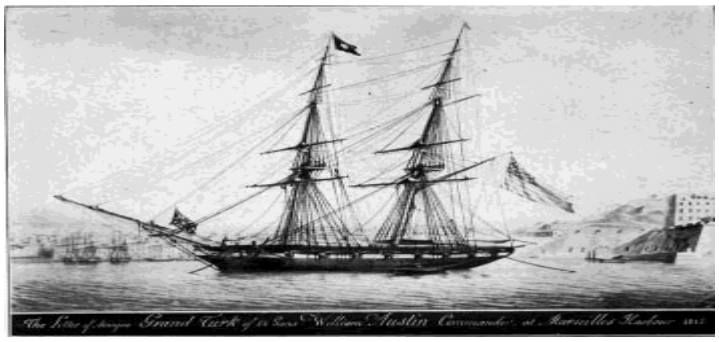
$$FF\{\lambda(i\Delta x, j\Delta y)\} = 1/\Delta x * 1/\Delta y \sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\lambda(f_x - k\Delta f_x, f_y - l\Delta f_y). \qquad (1.35)$$

Из рис.1.8 видно, что если соблюдается условие неперекрываемости периодических продолжений спектра $FF\{\lambda(i\Delta x,j\Delta y)\}$, а это справедливо при $\Delta x \leq 1/2f_x max$, $\Delta y \leq 1/2f_y\,max$, то с помощью идеального двумерного ФНЧ с ча-

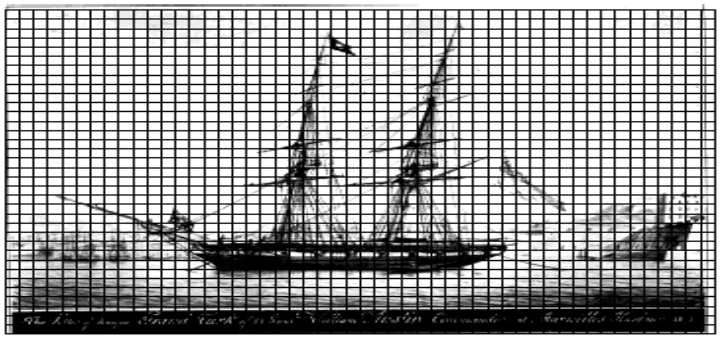
стотной характеристикой вида:

$$H\left(f_x, f_y\right) = \begin{cases} \Pi(f_x / \Delta f_x) * \Pi(f_y / \Delta f_y), \left|f_x\right| \le 1/2\,\Delta x, \left|f_y\right| \le 1/2\,\Delta y, \\ 0, \left|f_x\right| \ge 1/2\,\Delta x, \left|f_y\right| \ge 1/2\,\Delta y \end{cases} \quad (1.36)$$

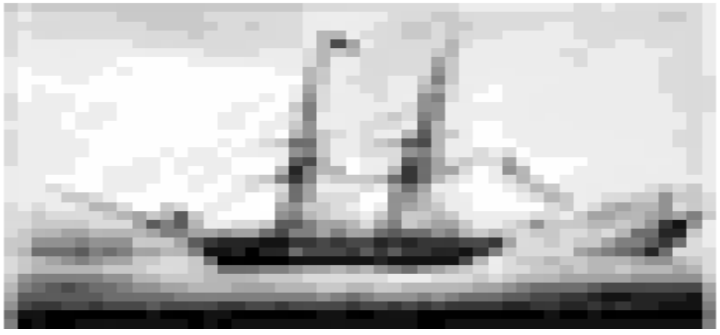
из спектра дискретизованной функции ***FF{λ(iΔx,jΔy)}*** можно абсолютно точно выделить спектр исходной непрерывной функции ***FF{λ(x,y)}*** и, следовательно, восстановить саму функцию.



а



б



в

Рис.1.13. Процедура дискретизации двухмерных изображение: а - исходное изображение; б - дискретизация по осям x и y; в – дискретизированное изображение.

Таким образом, видно, что не существует принципиальных отличий в дискретизации между одномерными и двумерными (многомерными) функциями. Результатом дискретизации в обоих случаях является совокупность отсчетов функции, различия могут быть лишь в величине шага дискретизации, числе отсчетов и порядке их следования.

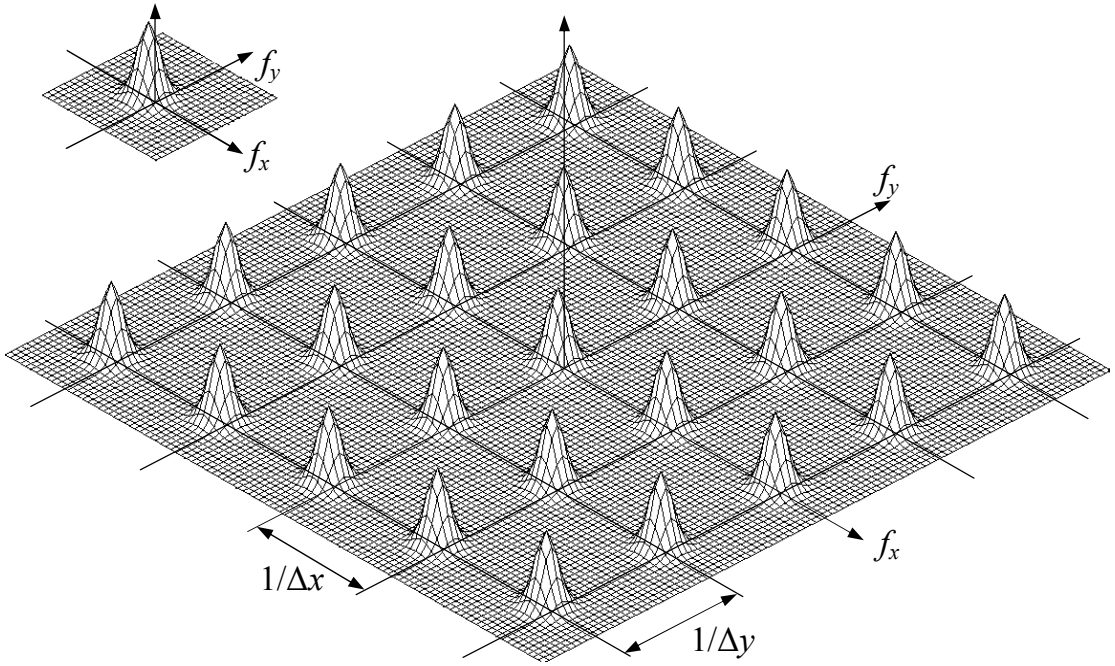


Рис. 1.14 Спектр дискретизированной двухмерной функции.

1.5.3. Квантование сообщений по уровню и по времени. Ошибки квантования.

Итак, показано, что передачу практически любых сообщений $\lambda(t)$ ($\{\lambda(x,y)\}$) можно свести к передаче их отсчетов, или чисел $\lambda_i = \lambda(i\,\Delta t)$, следующих друг за другом с интервалом дискретности $\Delta t \le 1/2F_m$ ($\Delta x \le 1/2f_x$, $\Delta y \le 1/2f_y$). Тем самым непрерывное (бесконечное) множество возможных значений сообщения $\lambda(t)$ заменяется конечным числом его дискретных значений $\{\lambda(i\,\Delta t)\}$. Однако сами эти числа имеют непрерывную шкалу уровней (значений), то есть принадлежат опять же континуальному множеству. Для абсолютно точного представления таких чисел, к примеру, в десятичной (или двоичной) форме, необходимо теоретически бесконечное число разрядов. Вместе с тем на практике нет необходимости в абсолютно точном представлении значений λ_i , как и любых чисел вообще.

Во-первых, сами источники сообщений обладают ограниченным динамическим диапазоном и вырабатывают исходные сообщения с определенным уровнем искажений и ошибок. Этот уровень может быть большим или меньшим, но абсолютной точности воспроизведения достичь невозможно.

Во-вторых, передача сообщений по каналам связи всегда производится в присутствии различного рода помех. Поэтому принятое (воспроизведенное) сообщение (оценка сообщения $\lambda^*(t)$ или Λ^*) всегда в определенной степени отли-

чается от переданного, то есть на практике невозможна абсолютно точная передача сообщений при наличии помех в канале связи.

Наконец, сообщения передаются для их восприятия и использования получателем. Получатели же информации - органы чувств человека, исполнительные механизмы и т.д. - также обладают конечной разрешающей способностью, то есть не замечают незначительной разницы между абсолютно точным и приближенным значениями воспроизводимого сообщения. Порог чувствительности к искажениям также может быть различным, но он всегда есть.

С учетом этих замечаний процедуру дискретизации сообщений можно продолжить, а именно подвергнуть отсчеты λ_i квантованию.

1.5.4. Квантование по времени и по уровню.

При преобразовании аналоговой величины в код квантование осуществляется с заданными шагами как по времени, так и по уровню.

На рис. 1.15 показано, как производится квантование по уровню и по времени функции $f(t)$. Сначала проводят линии, параллельные вертикальной оси $f(t)$ с шагом Δt, затем параллельные горизонтальной оси t с шагом q.

Квантование осуществляют заменой через шаг Δt значений функции $f(t)$ ближайшим дискретным уровнем. Этот уровень и является тем дискретным значением, которое заменяет значение функции в данный дискретный момент времени.

Если необходимо представить себе ступенчатую ломаную линию, которая в результате квантования заменяет непрерывную функцию, все полученные точки следует соединить так, как сделано на рис. 1.15.

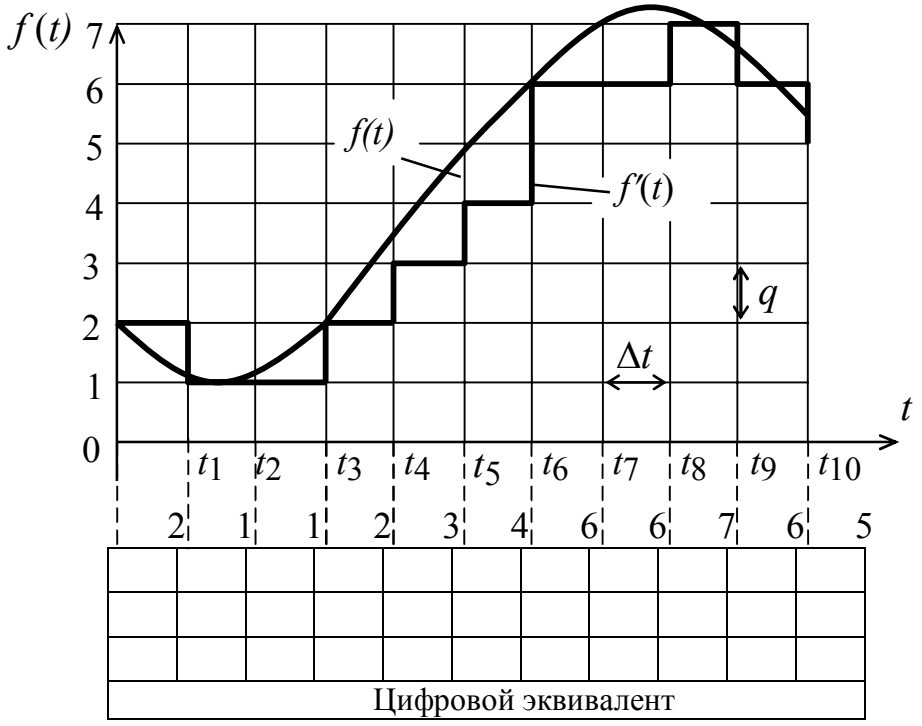


Рис. 1.15. Преобразование непрерывной величины в код

Различают равномерное и неравномерное квантование. В большинстве случаев применяется и далее подробно рассматривается равномерное квантование (рис. 1.16), при котором шаг квантования постоянный: $\boldsymbol{q = f_i - f_{i-1}}$ = const; однако иногда определенное преимущество дает неравномерное квантование, при котором шаг квантования $\boldsymbol{q_i}$ разный для различных $\boldsymbol{f_i}$ (рис. 1.17).

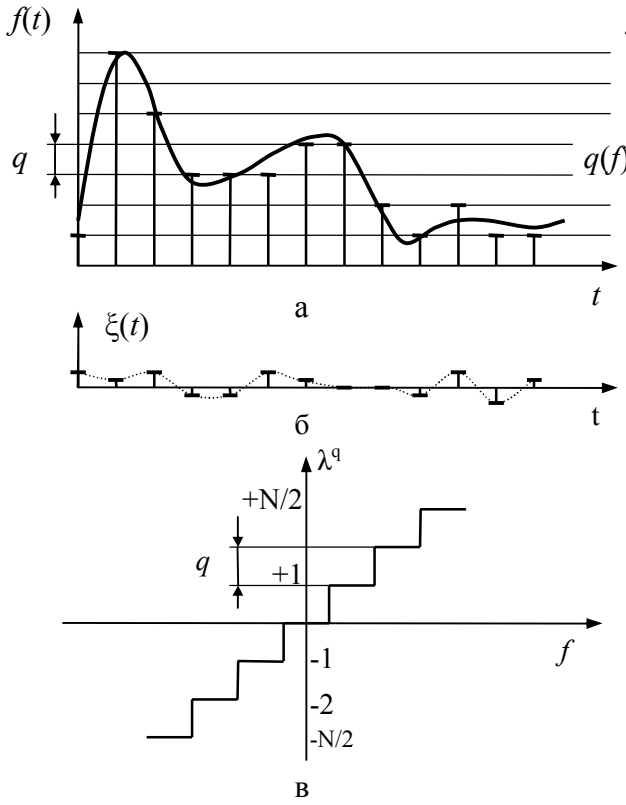


Рис. 1.16. Квантование по уровню равномерное: а - процесс квантования; б - погрешность квантования; в - характеристика квантования.

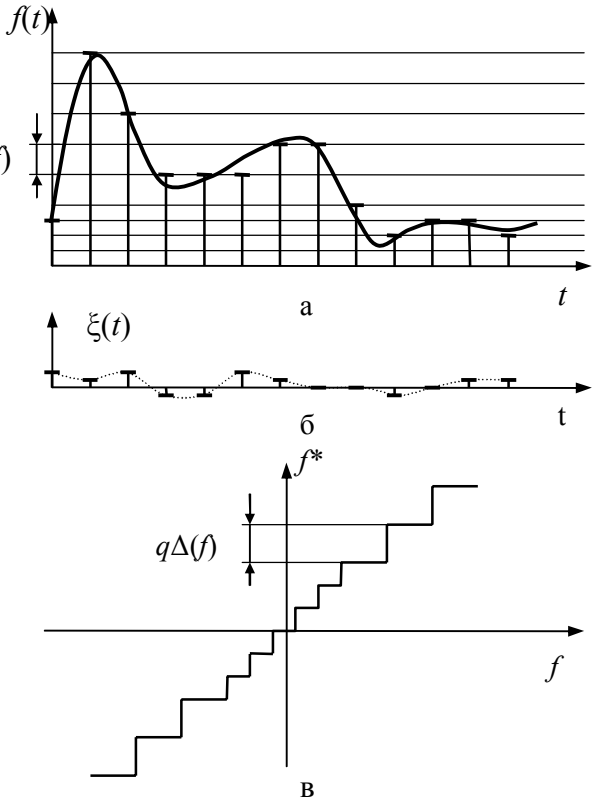


Рис. 1.17. Квантование по уровню неравномерное: а - процесс квантования; б - погрешность квантования; в - характеристика квантования.

Квантование приводит к искажению сообщений. Так как наименее точно функция передается в точке, находящейся между двумя уровнями квантования и отстоящей от них на половину интервала квантования $q/2$, то максимальная ошибка квантования по уровню

$$\Delta = \pm q/2\,, \tag{1.37}$$

а мощность шума квантования при равномерном квантовании

$$P_{ш} = q^2/12 \tag{1.38}$$

При достаточно большом числе уровней квантования N распределение погрешности квантования в пределах от $-q/2$ до $+q/2$ будет равномерным незави-

симо от закона распределения самой функции $f(t)$. Среднеквадратичное значение погрешности квантования по уровню

$$\delta_{ск} = q/(2\sqrt{3}), \tag{1.39}$$

т.е. в $\sqrt{3}$ раз меньше максимальной.

Что касается точности преобразования (квантования), то обычно она задается в виде приведенной относительной погрешности $\delta_{отн}$ в процентах. По определению, $\delta_{отн}\% = (\Delta \cdot 100)/(f(t)_{\max} - f(t)_{\min})$. Подставив значение Δ из (1.37), получим выражение для шага квантования при $f(t)_{\min} = 0$

$$q = 2f(t)_{\max}\delta_{отн}/100. \tag{1.40}$$

После того как непрерывное сообщение с помощью квантования будет преобразовано в дискретное сообщение, необходимо каждому его уровню присвоить цифровой эквивалент, как правило, в двоичном неизбыточном коде (см. рис. 1.15) и передать по каналу связи. При этом, если известен шаг квантования q, то число уровней квантования N и число разрядов кодовой комбинации K при $f(t)_{\min} = 0$ можно определить из выражения

$$N = f(t)_{\max}/q = 2^k - 1. \tag{1.41}$$

Пример 1. Предположим, что необходимо произвести квантование непрерывной функции, изменяющейся от нуля до 100 В, с точностью $\delta_{ск}$ = 1%. Определить величину шага квантования, число уровней квантования и число разрядов кодовой комбинации. Согласно (1.40), $q = 2$ В. Из (1.41) определим, что необходимо 50 уровней квантования, а число разрядов $K = E\log 51 = 6$. Такое число уровней устанавливается, если измерение в данной точке производят до ближайшего уровня (нижнего или верхнего). При схемной реализации отсчет часто производят до какого-нибудь одного уровня (только нижнего или только верхнего). В этом случае для обеспечения точности квантования в 1% от 100 В число уровней следует взять 100, так как $\Delta = q = f(t)_{\max}\delta_{отн}/100$, а следовательно k = 7.

Контрольные вопросы

1. Какой сигнал называется регулярным?
2. Запишите выражение для периодичного сигнала.
3. Какой сигнал называется не периодическим?
4. Приведите временную диаграмму для дискретного сигнала.
5. Запишите ряд Фурье для периодичного сигнала.

6. Что означает представление сигнала с заданным периодом рядом Фурье?

7. Приведите спектр периодически повторяющихся прямоугольных импульсов.

8. Приведите выражение для средней мощности периодических прямоугольных импульсов.

9. Что понимается под практической шириной спектра сигнала?

10. Приведите форму сигнала при ограничении спектра прямоугольных импульсов, имеющих скважность Q =3 и ограниченных третьей гармонической составляющей.

11. Приведите выражение для спектра непериодического сигнала.

12. Поясните график зависимости энергии импульсов от ширины сохраняемой части спектра.

13. Сформулируйте теорему В.А. Котельникова о преобразовании непрерывных сообщений в дискретные сигналы.

14. Сформулируйте теорему дискретизации для двумерных сигналов.

2. КОЛИЧЕСТВЕННАЯ ОЦЕНКА ИНФОРМАЦИИ

2.1. Количество информации при равновероятности состояний источника сообщений

Сообщения разнятся как по своей природе, так и по содержанию и по назначению. В связи с этим возникают трудности в оценке количества информации, которое содержится в сообщениях. Количество информации должно определяться через нечто общее, объективно присущее всему многообразию различной информации, оставаясь при этом созвучным нашим интуитивным представлениям, связанным с фактом получения информации. Этим общим, характеризующим фактом получения произвольной информации, является, во-первых, наличие опыта. Всякая информация добывается нами в результате опыта и только опыта. Во-вторых, до опыта должна существовать некоторая неопределенность в том или ином исходе опыта.

Таким образом, до опыта всегда имеется большая или меньшая неопределенность в интересующей нас ситуации. После опыта ситуация становится более определенной и на поставленный вопрос мы можем ответить либо однозначно, либо число возможных ответов уменьшится и, следовательно, уменьшится существовавшая ранее неопределенность. Количество уменьшенной неопределенности после опыта, очевидно, можно отождествить с количеством получаемой информации в результате опыта.

Теперь ясно, что для установления формулы для вычисления количества информации необходимо уметь вычислять неопределенность некоторой ситуации до и после опыта. Разность между этими количествами неопределенности и

дает нам искомое количество информации, полученное от такого опыта.

К количеству информации (неопределенности до опыта) можно предъявить три интуитивных требования.

1. Количество получаемой информации больше в том опыте, у которого большее число возможных исходов.

Обозначая количество информации буквой I, а число возможных исходов n, первый постулат запишем в виде:

$$I(n_1) \geq I(n_2), \text{ если } n_1 \geq n_2. \qquad (2.1)$$

2. Опыт с единственным исходом несет количество информации, равное нулю, т.е.

$$I(n = 1) = 0. \qquad (2.2)$$

3. Количество информации от двух независимых опытов равно сумме количества информации от каждого из них:

$$I(n_1 \cdot n_2) = I(n_1) + I(n_2). \qquad (2.3)$$

Очевидно, единственной функцией аргумента n, удовлетворяющей трем поставленным условиям, является логарифмическая. Итак, количество информации от опыта с N исходами при условии, что после опыта неопределенность отсутствует:

$$I = C \log_2 N. \qquad (2.4)$$

Выбор постоянной C и основания логарифмов здесь несущественен, так как определяет только масштаб на единицу неопределенности. Поэтому положим $C = 1$, $a = 2$. Тогда

$$I = \log N. \qquad (2.5)$$

Указанная мера была предложена *P.* Хартли в 1928г. для количественной оценки способности системы хранить или передавать информацию.

Такая мера удовлетворяет требованию аддитивности. Емкость устройства состоящего из n ячеек, имеющего $N = m^n$ состояний, равна емкости одной ячейки, умноженной на число ячеек:

$$C = \log m^n = n \log m. \qquad (2.6)$$

За единицу измерения емкости принята двоичная единица или bit, равная емкости одной ячейки с двумя возможными состояниями.

Следует отметить, что мера количества информации в виде (2.6) относится к весьма частному случаю, когда после опыта неопределенности в исходе нет и все исходы равновероятны.

Дальнейшее развитие теории информации шло в направлении учета стати-

стических характеристик.

Если от источника информации по каналу связи передавать сообщение о событии, априорная вероятность которого на передающей стороне равна P_1, то после приема сообщения апостериорная вероятность этого события для приемника информации равна P_2 и количество информации, полученное в результате приема сообщения, будет

$$I = \log(P_2 / P_1) = \log P_2 - \log P_1. \qquad (2.7)$$

Для канала связи без помех и искажений прием сообщения становится достоверным событием, т.е. вероятность $P_2 = 1$, тогда из (2.7) следует, что

$$I = -\log P_1. \qquad (2.8)$$

Из (2.8) следует, что чем меньше вероятность P_1, тем больше неопределенность исхода, т.е. тем большее количество информации содержится в принятом сообщении.

Значение P_1 находится в пределах $0 < P_1 < 1$, следовательно, $I = -\log P_1$ всегда положительная величина.

Количество информации $I = -\log P$, где P – вероятность события, было положено в основу и было исходной точкой создания теории информации.

2.2. Энтропия ансамбля

Ансамблем называется полная совокупность состояний с вероятностями их появлений, составляющими в сумме единицу:

$$X = \begin{pmatrix} X_1 & X_2 & \cdots X_j & \cdots X_k \\ P_1 & P_2 & \cdots P_j & \cdots P_k \end{pmatrix}, \qquad (2.9)$$

причем $\sum_{i=1}^{k} P_i = 1$.

Пусть имеет место N возможных исходов опыта, из них k разных, и i-й исход (i = 1, 2,..., k) повторяется n_i раз и вносит информацию, количество которой оценивается как I_i . Тогда средняя информация, доставляемая одним опытом, равна

$$I_{CP} = \frac{n_1 I_1 + n_2 I_2 + \cdots + n_k I_k}{N}. \qquad (2.10)$$

Но количество информации в каждом исходе согласно (2.8) будет

$$I_i = -\log P_1. \qquad (2.11)$$

Тогда

$$I_{CP} = \frac{n_1(-\log P_1) + n_2(-\log P_2) + \cdots + n_k(-\log P_k)}{N}. \quad (2.12)$$

Но отношение $\frac{n_i}{N}$ представляют собой частоты повторения исходов, а следовательно, могут быть заменены их вероятностями:

$$\frac{n_i}{N} = P_i \ . \quad (2.13)$$

Подставляя (2.13) в (2.12), получим

$$I_{CP} = P_1(-\log P_1) + P_2(-\log P_2) + \cdots + P_k(-\log P_k) = -\sum_{i=1}^{k} P_i \log P_i.$$

Полученную величину К. Шеннон назвал энтропией и обозначил буквой H, бит:

$$H = I_{cp} = -\sum_{i=1}^{k} P_i \log P_i \ . \quad (2.14)$$

Энтропия H представляет собой логарифмическую меру беспорядочности состояния источника сообщений и характеризует степень неопределенности состояния этого источника. Получение информации – это процесс раскрытия неопределенности.

В информационных системах неопределенность снижается за счёт принятой информации, поэтому численно энтропия H равна среднему количеству информации, несомой произвольным исходом x_i, т.е. является количественной мерой информации.

Если все k различных состояний источника равновероятны, то

$$P_i = \frac{1}{k},$$

энтропия максимальна и из (2.14) имеем

$$H\max = -\sum_{i=1}^{k} \frac{1}{k} \log \frac{1}{k} = \log k \cdot \quad (2.15)$$

Нетрудно заметить, что в частном случае при равновероятных сообщениях формулы (2.14) и (2.5) совпадают. Совпадение оценок количества информации по Шеннону и Хартли свидетельствуют о полном использовании информационной емкости системы. В случае неравных вероятностей количество информации по Шеннону меньше информационной емкости системы.

2.3. Энтропия объединения

Объединением называется совокупность двух и более взаимозависимых ансамблей дискретных случайных переменных.

Рассмотрим объединение, состоящее из двух ансамблей X и Y, например из двух дискретных измеряемых величин, связанных между собой вероятностными зависимостями. Объединение ансамблей характеризуется матрицей $P(X,Y)$ вероятностей $P(x_i, y_i)$ всех возможных комбинаций состояний $x_i (1 \le i \le n)$ ансамбля X и состояний $y_i (1 \le i \le m)$ ансамбля Y:

$$P(X,Y) = \begin{Vmatrix} P(x_1,y_1) & \cdots & P(x_i,y_1) & \cdots & P(x_n,y_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ P(x_1,y_j) & \cdots & P(x_i,y_j) & \cdots & P(x_n,y_j) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ P(x_1,y_m) & \cdots & P(x_i,y_m) & \cdots & P(x_n,y_m) \end{Vmatrix}. \qquad (2.16)$$

Суммируя столбцы и строки матрицы (2.16), получим информацию об ансамблях X и Y исходных источников:

$$X = \begin{vmatrix} x_1 & \cdots & x_i & \cdots & x_n \\ P(x_1) & \cdots & P(x_i) & \cdots & P(x_n) \end{vmatrix}, \quad Y = \begin{vmatrix} y_1 & \cdots & y_j & \cdots & y_m \\ P(y_1) & \cdots & P(y_j) & \cdots & P(y_m) \end{vmatrix}, \qquad (2.17)$$

где $P(x_i) = \sum_{j=1}^{m} P(x_i, y_j)$ и $P(y_j) = \sum_{i=1}^{n} P(x_i, y_j)$.

Вероятности $P(x_i, y_j)$ совместной реализации взаимозависимых состояний x_i и y_i можно выразить через условные вероятности $P(x_i/y_j)$ или $P(y_j/x_i)$ в соответствии с тем, какие состояния принять за причину, а какие – за следствие.

$$P(x_i, y_j) = P(x_i)P(y_j / x_i) = P(y_j)P(x_i / y_j), \qquad (2.18)$$

где $P(x_i/y_j)$ – вероятность реализации состояний x_i ансамбля X при условии, что реализовалось состояние y_j ансамбля Y; $P(y_j/ x_i)$ – вероятность реализации состояний y_j ансамбля Y при условии, что реализовалось состояние x_i ансамбля X.

Тогда выражение для энтропии объединения в соответствии с (2.14) принимает вид:

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j)\log P(x_i, y_j) =$$

$$= -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i)P(y_j / x_i)\log P(x_i)P(y_j / x_i) =$$

$$= -\sum_{i=1}^{n} P(x_i)\log P(x_i) - \sum_{i=1}^{n} P(x_i)\sum_{j=1}^{m} P(y_j / x_i)\log P(y_j / x_i), \qquad (2.19)$$

где $-\sum_{j=1}^{m} P(y_j / x_i)\log P(y_j / x_i)$ – случайная величина, характеризующая неопределенность, приходящуюся на одно состояние ансамбля Y при условии, что реализовалось конкретное состояние x_i ансамбля X. Назовем её частной условной энтропией ансамбля Y и обозначим $H(Y/x_i)$:

$$H(Y / x_i) = -\sum_{j=1}^{m} P(y_j / x_i)\log P(y_j / x_i). \qquad (2.20)$$

При усреднении по всем состояниям ансамбля X получаем среднюю неопределенность, приходящуюся на одно состояние ансамбля Y при известных состояниях ансамбля X:

$$H(Y / X) = -\sum_{i=1}^{n} P(x_i)H(Y / x_i) =$$

$$= -\sum_{i=1}^{n} P(x_i)\sum_{j=1}^{m} P(x_i)P(y_j / x_i)\log P(y_j / x_i). \qquad (2.21)$$

Величину $H(Y/X)$ называют полной условной или просто условной энтропией ансамбля Y по отношению к ансамблю X.

Подставляя (2.21) в (2.19), получаем

$$H(X,Y) = H(X) + H(Y / X). \qquad (2.22)$$

Выражая $P(x_i, y_j)$ через другую условную вероятность в соответствии с (1.18), найдем

$$H(X,Y) = H(Y) + H(X / Y), \qquad (2.23)$$

где

$$H(X / Y) = \sum_{j=1}^{m} P(y_j)H(X / y_j) \qquad (2.24)$$

и

$$H(X / y_j) = -\sum_{i=1}^{n} P(x_i / y_j)\log P(x_i / y_j). \qquad (2.25)$$

Таким образом, энтропия объединения двух статистически связанных ансамблей X и Y равна безусловной энтропии одного ансамбля плюс условная энтропия другого относительно первого.

В случае статистической независимости ансамблей X и Y имеют

$$P(x_i, y_j) = P(x_i)P(y_j).$$

Тогда

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i)P(y_j)\log P(x_i)P(y_j) =$$

$$= -\sum_{i=1}^{n} P(x_i)\log P(x_i)\sum_{j=1}^{m} P(y_j) - \sum_{j=1}^{m} P(y_j)\log P(y_j)\sum_{i=1}^{n} P(x_i).$$

Учитывая, что

$$\sum_{j=1}^{m} P(y_j) = 1 \text{ и } \sum_{i=1}^{n} P(x_i) = 1,$$

получим

$$H(X,Y) = H(X) + H(Y) = H(Y,X). \qquad (2.26)$$

2.4. Свойства энтропии

2.4.1. Энтропия всегда неотрицательна, так как значения вероятностей выражаются дробными величинами, а их логарифмы – отрицательными величинами (2.14).

2.4.2. Энтропия равна нулю в том крайнем случае, когда одно событие равно единице, а все остальные – нулю. Это положение соответствует случаю, когда состояние источника полностью определено.

2.4.3. Энтропия имеет наибольшее значение при условии, когда все вероятности равны между собой (2.15).

2.4.4. Энтропия источника X с двумя состояниями x_1 и x_2 изменяется от нуля до единицы, достигая максимума при равенстве их вероятностей

$$P(x_1) = P = P(x_2) = 1 - P = 0{,}5.$$

График зависимости $H(X)$ в функции P:

$$H(X) = -[P\log P + (1-P)\log(1-P)], \qquad (2.27)$$

приведен на рис. 2.1.

Отметим, что энтропия непрерывно зависит от вероятности отдельных состояний, что непосредственно вытекает из непрерывности функции $-P\log P$.

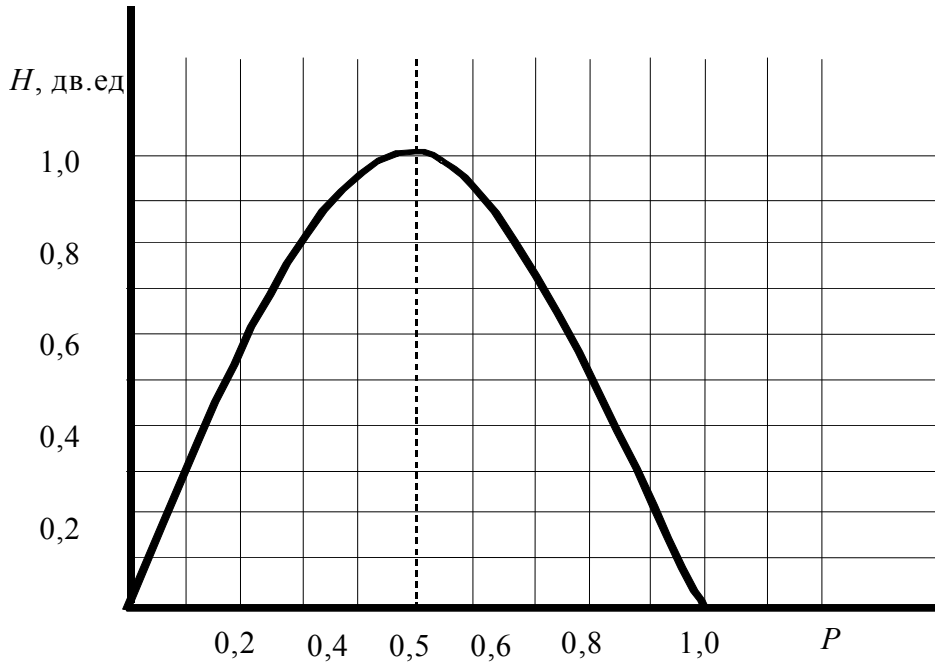


Рис. 2.1. Зависимость $H(X)$ в функции P

2.4.5. Энтропия объединения нескольких статистически независимых источников информации равна сумме энтропий исходных источников

$$H(X,Y,Z,...,W) = H(X) + H(Y) + H(Z) + ... + H(W). \qquad (2.28)$$

2.4.6. Энтропия объединения двух статистически связанных ансамблей X и Y равна

$$H(X,Y) = H(X) + H(Y,X).$$

2.4.7. Энтропия объединения любого числа зависимых ансамблей определяется из выражения

$$H(X,Y,Z,...,W) = H(X) + H(Y/X) + H(Z/X,Y) + ... + H(W/X,Y,Z...). \quad (2.29)$$

2.4.8. Энтропия не зависит от значений, принимаемых случайными величинами, а зависит только от вероятностей их появления (2.14).

2.4.9. Если события x_i и y_j статистически независимы при любых i и j, то

$$H(Y/X) = H(Y) \text{ и } H(X/Y) = H(X). \qquad (2.30)$$

Таким образом, сведения о результатах выбора состояний из одного ансамбля не снижает неопределенности выбора состояний из другого ансамбля. Если имеет место однозначная связь в реализациях состояний $x_i (1 \le i \le n)$ из ансамбля X и $y_j (1 \le j \le n)$ из ансамбля Y, то условная энтропия любого из ансамблей равна нулю:

$$H(Y/X) = 0,\ H(X/Y) = 0. \qquad (2.31)$$

Действительно, условные вероятности $P(x_i/y_j)$ и $P(y_j, x_i)$ в этом случае принимают значения, равные нулю или единице. Поэтому все слагаемые, входящие в выражения (2.20) и (2.25), для частных условных энтропий равны нулю. Тогда в соответствии с (2.21) и (2.24) условные энтропии равны нулю.

Равенства (2.31) отражают факт отсутствия дополнительной неопределенности при выборе событий из второго ансамбля.

Уяснению соотношений между рассмотренными энтропиями дискретных источников информации (ансамблей) соответствует их графическое отображение (рис. 2.2).

2.5. Количество информации от опыта в общем случае

Передача информации инициируется либо самим источником информации, либо осуществляется по запросу. На приемной стороне любой системы передачи информации до получения сигнала от интересующего нас источника неизвестно, какой из возможных сигналов будет передан, но считается известным распределение вероятностей $P(x_i)$ по всем сигналам. Неопределенность ситуации до приема сигнала характеризуется энтропией:

$$H(X) = -\sum_{i=1}^{m} P(x_i) \log P(x_i). \qquad (2.32)$$

Далее в приемное устройство поступает принятый сигнал. Поскольку предполагается, что принятый сигнал соответствует переданному (помехи отсутствуют), то неопределенность относительно источника информации снимается полностью.

Таким образом, в результате приема сигнала, с одной стороны, произошло уменьшение неопределенности с $H(X)$ до нуля, а с другой стороны, получено количество информации I, численно равное энтропии $H(X)$. Отсюда следует, что количество информации может быть определено как мера снятой неопределенности. Численное значение количества информации о некотором объекте равно разности энтропий объекта до и после приема сигнала. Значит, понятие энтропия является первичным, исходным, а понятие количество информации – вторичным, производным понятием. Энтропия есть мера неопределенности, а количество информации – мера изменения неопределенности.

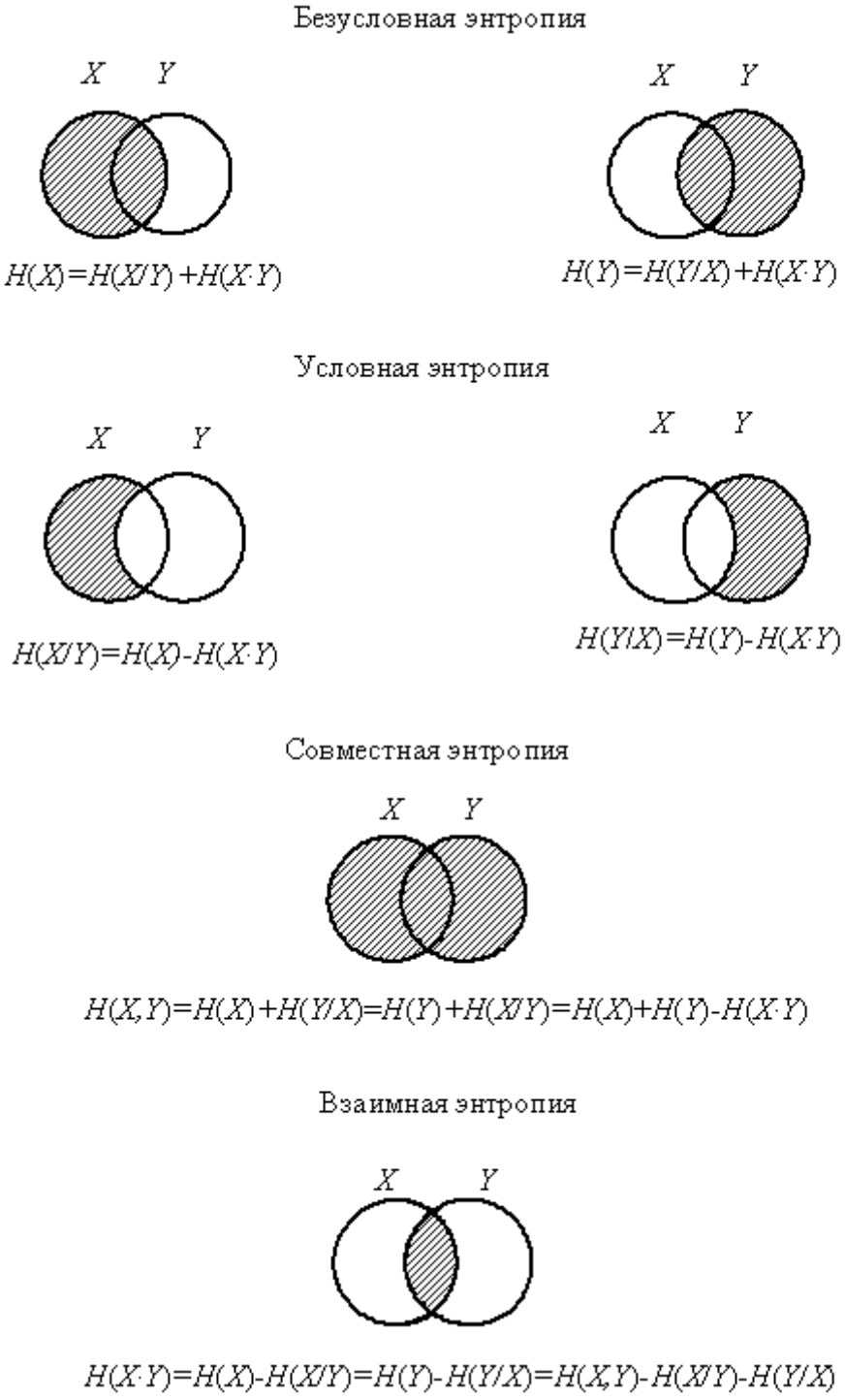


Рис. 2.2. Энтропия объединения

Если помехи существуют, то принятый сигнал в той или иной степени не тождественен переданному. Здесь исчезает численное совпадение *I* и *H*. Количество информации будет меньше, чем при отсутствии помех, так как прием сигнала не уменьшает энтропию до нуля.

Рассмотрим случай, когда между элементами сообщения и помехой статистические связи отсутствуют, искажения отдельных элементов сообщения яв-

ляются событиями независимыми и адресату известна совокупность условных вероятностей $P(x_i / y_j)(1 \le i \le m, 1 \le j \le m)$ того, что вместо элемента сообщения x_i будет принят элемент сообщения y_j.

Среднее количество неопределенности, которым мы обладали до опыта, равнялось $H(X)$. Представим теперь что мы приняли какой-то сигнал y_j и оцениваем, какова неопределенность (после опыта) соответствия его некоторому переданному x_i. Эта неопределенность равна

$$H(x_i / y_j) = -\log P(x_i / y_j). \qquad (2.33)$$

Как видим, неопределенность этого соответствия является случайной величиной, значения которой при каждом заданном y_j наступают с вероятностями $P(x_i/y_j)$. Поэтому среднее значение количества неопределенности соответствия данного y_j любому из x_i равно

$$H(X / y_j) = -\sum_{i=1}^{m} P(x_i / y_j) \log P(x_i / y_j). \qquad (2.34)$$

Величина $H(X/y_j)$ также случайна. Вероятности её значений равны $P(y_j)$. Тогда среднее значение $H(X/y_j)$ определит среднее количество неопределённости соответствия любого y_j любому из x_i. Обозначим это среднее $H(X/Y)$:

$$H(X / Y) = \sum_{i=1}^{m} H(X / y_j) P(y_j) = -\sum_{i=1}^{m} \sum_{j=1}^{m} P(x_i, y_j) \log P(x_i / y_j). \qquad (2.35)$$

Другими словами, $H(X/Y)$ есть средняя неопределённость в передаче того или иного x_i, если известно, что принят тот или иной y_j, или, кратко, средняя неопределённость ансамбля X после опыта.

Таким образом, мы установили, что неопределённость передачи некоторого сигнала X до опыта $H(X)$, а после опыта $H(X/Y)$. Поэтому количество информации, имеющееся в Y о X:

$$I(Y,X) = H(X) - H(X/Y). \qquad (2.36)$$

Эта мера количества информации получена нами на примере передачи сообщений по каналу связи. Совершенно аналогичные рассуждения могут быть применены к случайным объектам произвольного вида и приведут нас к той же мере.

Подставим в выражение (2.36) необходимые значения $H(X)$ и $H(X/Y)$ из (2.32) и (2.35) соответственно получим

$$I(Y,X) = -\sum_{i=1}^{m} P(x_i)\log(x_i) + \sum_{i=1}^{m}\sum_{j=1}^{m} P(x_i, y_j)\log P(x_i / y_j) =$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} P(x_i, y_j)\log P(x_i) + \sum_{i=1}^{m}\sum_{j=1}^{m} P(x_i, y_j)\log P(x_i / y_j) =$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} P(x_i, y_j)\log \frac{P(x_i, y_j)}{P(x_i)P(y_j)}. \qquad (2.37)$$

Если частный характер количества информации специально не оговаривается, мы всегда имеем дело с количеством информации, приходящимся в среднем на один элемент сообщения. Поэтому указание об усреднении опускаются.

2.6. Основные свойства количества информации

2.6.1. $I(X,Y) = I(Y,X)$, т.е. количество информации, содержащееся в случайном объекте Y о случайном объекте X, равно количеству информации, содержащемуся в случайном объекте X о случайном объекте Y. Свойство 2.6.1 сразу же следует из (2.37), если учесть, что $P(x_i, y_j) = P(y_j, x_i)$.

2.6.2. $I(X,Y) \geq 0,$ причём знак равенства имеет место, когда объекты X и Y независимы.

Положительность $I(X,Y)$ следует из свойства энтропии : если события x_i и y_j статистически зависимы, то всегда $H(Y/X) < H(Y)$ и $H(X/Y) < H(X)$.

2.6.3. $I(X,Y) = H(X)$, т.е. энтропия может быть истолкована как информация, содержащаяся в объектах относительно самих себя. Из этого также непосредственно вытекает, что энтропия есть максимальное количество информации, которое можно получить об объекте. Это возможно при взаимно однозначном соответствии между множествами передаваемых и принимаемых сообщений, что имеет место в отсутствии помехи, апостериорная энтропия равна нулю и количество информации численно совпадает с энтропией источника.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чём сущность требования аддитивности к мере неопределённости выбора?
2. Назовите основной недостаток меры неопределённости, предложенной Р. Хартли.
3. Укажите достоинства и недостатки способа, предложенного К. Шенноном.
4. Что необходимо учитывать при выборе способа измерения количества информации?
5. В каких единицах измеряется количество информации?
6. Дайте определение энтропии.
7. Назовите основные свойства энтропии дискретного ансамбля.

8. Почему вводится понятие условной энтропии? Запишите выражение для условной энтропии и поясните её смысл.

9. Приведите выражение для энтропии двух взаимосвязанных ансамблей.

10. Как связаны между собой понятия количества информации и энтропии?

11. В чём различаются понятия частного и среднего количества информации?

12. Когда энтропия источника с двумя состояниями достигает максимума?

13. От чего не зависит энтропия случайного процесса?

14. Запишите выражение для энтропии объединения нескольких независимых источников информации.

15. Перечислите свойства количества информации.

3. ИСТОЧНИКИ ДИСКРЕТНЫХ СООБЩЕНИЙ

3.1. Энтропия эргодического источника

Достаточно хорошей математической моделью дискретных источников, встречающихся на практике, являются так называемые эргодические источники. Назовём эргодическим источником r-го порядка такой источник, у которого вероятность появления некоторого символа x_j зависит от r предыдущих. Для такого источника может быть найдено конечное число характерных состояний S_1, S_2,..., таких, что условная вероятность появления очередного символа зависит лишь от того, в каком из этих состояний находится источник. Вырабатывая очередной символ, источник переходит из одного состояния в другое либо возвращается в исходное состояние.

Определим энтропию эргодического источника в предположении, что он работает длительное время и, всякий раз, когда мы ждём появления очередного символа, нам известно, какие символы были выбраны ранее, и, следовательно, известно, в каком характерном состоянии находится источник.

Обозначим через $P(S_i)$ вероятность того, что источник находится в состоянии S_i, причём

$$\sum_{i=1}^{n} P(S_i) = 1. \qquad (3.1)$$

Предположим, мы установили, что источник находится в состоянии S_b. У нас имеется неопределённость, из какого состояния S_k источник, выработав некоторый символ, перешёл в состояние S_b. Так как вероятность состояния S_b зависит только от предыдущего состояния S_k и не зависит от того, в каких состояниях находился источник ранее, неопределённость источника в состоянии S_k можно найти по формуле, аналогичной (2.14):

$$H(S_k) = -\sum_{b/k} P(S_b / S_k) \log P(S_b / S_k). \qquad (3.2)$$

Величина $H(S_k)$ случайно меняется в зависимости от состояния источника, поэтому только среднее значение $H(S_k)$ может характеризовать энтропию источника:

$$H(X) = \sum_k P(S_k)H(S_k) = -\sum_k \sum_{b/k} P(S_k)P(S_b / S_k)\log P(S_b / S_k) = \\ = \sum_k \sum_{b/k} P(S_b, S_k)\log P(S_b / S_k), \qquad (3.3)$$

где значок b/k у суммы означает, что производится суммирование по всем переходам из состояния S_k в S_b.

Таким образом, энтропия $H(X)$ есть среднее значение (математическое ожидание) энтропий всех характерных состояний источника.

В случае, когда символы источника независимы, имеется лишь одно состояние S_1, вероятность которого $P(S_1) = 1$. При появлении символа x_i источник вновь возвращается в состояние S_1 (рис. 3.1), и при этом $P(S_1/S_1) = P(x_i)$, следовательно,

$$H(X) = H(S_1) = -\sum_{i=1}^{n} P(x_i)\log P(x_i),$$

что совпадает с (2.14).

Если коррелятивные связи имеются между двумя соседними символами, то $P(S_k) = P(x_k)$ и $P(S_b/S_k) = P(x_b/x_k)$.

Из (3.3) тогда получим

$$H(X) = -\sum_{k=1}^{n} P(x_k)\sum_{b=1}^{n} P(x_b / x_k)\log P(x_b / x_k) = \\ = -\sum_{k=1}^{n}\sum_{b=1}^{n} P(x_k, x_b)\log P(x_b / x_k) \ \frac{\text{дв.ед.}}{\text{символ}}. \qquad (3.4)$$

Источник, генерирующий n разных символов – $x_1, x_2, \ldots, x_n$, в этом случае может иметь n характерных состояний. Пример такого источника для случая $n = 3$ приведён на рис. 3.2.

В случае когда коррелятивные связи имеются между тремя символами, характерные состояния определяются передачей двух символов, и их удобно нумеровать двумя индексами. Так, если генерируются $x_h\, x_j$, то источник переходит в состояние $S_{h\text{j}}$ и тогда:

$$P(S_{h\text{j}}) = P(x_h, x_j) \text{ и } P(S_{ji}/S_{hj}) = P(x_i/x_h, x_j)\ .$$

Из (3.4) получаем

$$H(X) = \sum_{h=1}^{n}\sum_{j=1}^{n} P(x_h, x_j)\sum_{i=1}^{n} P(x_i / x_h, x_j)\log P(x_i / x_h, x_j) = \\ = \sum_{h=1}^{n}\sum_{j=1}^{n}\sum_{i=1}^{n} P(x_h, x_j, x_i)\log P(x_i / x_h, x_j) \ \frac{\text{дв.ед.}}{\text{символ}}. \qquad (3.5)$$

Чисел характерных состояний для этого случая столько, сколько имеется различных пар (x_i,x_h). Таких пар, очевидно, n^2.

Аналогичные соотношения получаются и в случае, когда корреляционные связи распространяются на большее число символов.

3.2. Свойство энтропии эргодических источников

Теорема 1. Для любых $\varepsilon > 0$ и $\delta > 0$ можно найти такое M_0, при котором эргодические последовательности с длиной $M > M_0$ распадаются на два класса:

1) типичные, вероятности которых удовлетворяют следующему неравенству:

$$\left| H(X) - \frac{\log \dfrac{1}{P(C)}}{M} \right| < \delta, \tag{3.6}$$

где $H(X)$ – энтропия эргодического источника;

2) нетипичные, сумма вероятностей которых меньше ε.

Доказательство. Последовательность C длины M образуется в результате поочередного перехода источника из одного характерного состояния в другое.

Рассмотрим переход из состояния S_k в состояние S_b во множестве последовательностей C. Число M всегда может быть выбрано настолько большим, что сумма вероятностей всех нетипичных последовательностей меньше ε. Учитывая, что для типичной последовательности частота событий может быть как угодно близка к их вероятности, можно утверждать, что в каждой типичной последовательности источник пребывает в состоянии S_k приблизительно $M_p(S_k)$ раз, а число переходов из состояния S_k в состояние S_b приблизительно равно $M_p(S_k)\, P(S_b/S_k)$, а точнее

$$M(P(S_k)\, P(S_b/S_k) \pm \eta), \tag{3.7}$$

где η с увеличением M может быть сделано как угодно малым.

Вероятность того, что в рассматриваемой последовательности имеет место M переходов из состояния S_k в S_b, равна по теореме умножения вероятностей

$$P(S_b / S_k)^{M(P(S_k)P(S_b/S_k)\pm\eta)}. \tag{3.8}$$

Вероятность появления конкретной последовательности C определяется как вероятность всех возможных переходов, и, следовательно,

$$P(C) = \prod_k \prod_{b/k} P(S_b / S_k)^{M(P(S_k)P(S_b/S_k)\pm\eta)}. \tag{3.9}$$

Логарифмируя последнее равенство, получаем:

$$\frac{\log \dfrac{1}{P(C)}}{M} = -\sum_k \sum_{b/k} P(S_k) P(S_b / S_k) \log P(S_b / S_k) \pm \eta \sum_k \sum_{b/k} \log(S_b / S_k) \,. (3.10)$$

Если учесть, что первая сумма в правой части совпадает с (3.3), а вторая вследствие произвольной малости η всегда может быть меньше δ, то получим неопределённость (3.6).

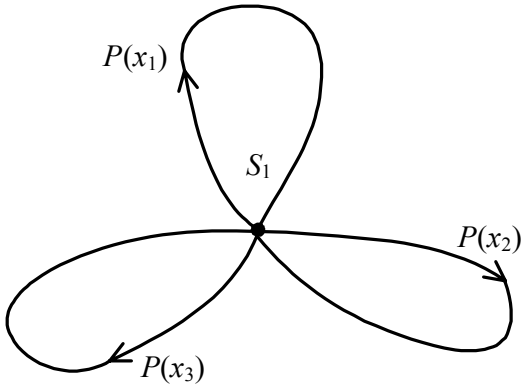


Рис. 2.1. Диаграмма перехода, когда источник имеет одно характерное состояние

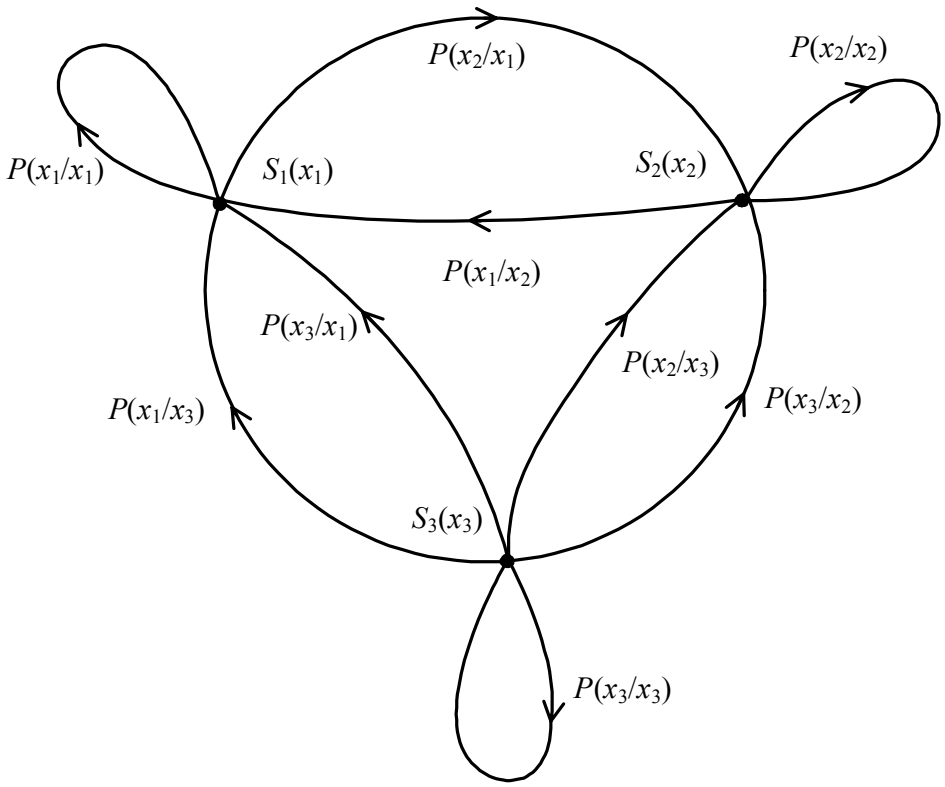


Рис. 2.2. Диаграмма перехода, когда источник имеет три характерных состояния

Следствие 1. Типичные последовательности приблизительно равновероятны. Для доказательства из (3.6) найдём $P(C)$:

$$2^{-M(H+\delta)} < P(C) < 2^{-M(H-\delta)}.$$

При $M \to \infty,\ \delta \to 0$. Поэтому при достаточно большом M можно положить

$$P(C) \approx 2^{-MH(X)}. \qquad (3.11)$$

Из (2.11) видно, что все последовательности C равновероятны и число их

$$N_{\mathrm{T}} = 2^{MH(X)}. \qquad (3.12)$$

В случае, если все n символов источника независимы и равновероятны, то

$$N_{\mathrm{T}} = 2^{M\log n} = n^{M}. \qquad (3.13)$$

Легко видеть, что (3.13) определяет число всех возможных последовательностей длины M, содержащих n различных символов.

Следствие 2. Чтобы экспериментально определить энтропию эргодического источника, у которого вероятностные связи распространяются на очень большое число символов, нам необходимо располагать последовательностью ещё большей длины ($M >> r$); при этом вычисленная энтропия будет как угодно близка к своему пределу $\log 1/P(C) = H(X)$.

3.3. Избыточность источника сообщений

Как известно, энтропия характеризует среднее количество информации, несомое одним символом источника. Она максимальна, когда символы вырабатываются источником с равной вероятностью. Если же некоторые символы появляются чаще других, энтропия уменьшается, а при появлении дополнительных вероятностных связей между символами становится ещё меньше. Чем меньше энтропия источника отличается от максимальной, тем рациональнее он работает, тем большее количество информации несут его символы.

Для сравнения источников по их информативности введём параметр, называемый избыточностью, равный

$$R = \frac{H_{\max}(X) - H(X)}{H_{\max}(X)}. \qquad (3.14)$$

Источник, избыточность которого $R = 0$, называется оптимальным. Если $R = 1$, то $H(X) = 0$, и, следовательно, информация, вырабатываемая источником, равна нулю. В общем случае $0 \le R \le 1$. Чем меньше избыточность R, тем рациональнее работает источник.

Следует, однако, иметь в виду, что не всегда нужно стремиться к тому, чтобы $R = 0$. Некоторая избыточность бывает полезной для обеспечения надежности передачи сообщений. Простейшим видом введения избыточности для борьбы с шумами является многократная передача одного и того же символа.

3.4. Поток информации источника сообщений

При работе источника сообщений на его выходе отдельные символы появляются через некоторые интервалы времени; в этом смысле мы можем говорить о длительности отдельных символов, и, следовательно, может быть поставлен вопрос о количестве информации, вырабатываемой источником в единицу времени.

Длительность выдачи знаков источником в каждом состоянии в общем случае может быть различной. Тогда средняя длительность выдачи источником одного знака:

$$\overline{\tau} = \sum_k P(S_k) \sum_i P(x_i) \tau_{x_i}, \tag{3.15}$$

где $P(S_k)$ – вероятность того, что источник сообщений находится в состоянии S_k; $P(x_i)$ – вероятность появления символа x_i в состоянии S_k; τ_{xi} – длительность выдачи знака x_i источником в состоянии S_k.

Энтропия источника, приходящаяся на единицу времени, может быть названа скоростью создания сообщений, или потоком информации, т.е.

$$\overline{H}(X) = \frac{H(X)}{\overline{\tau}} \ \frac{\text{дв.ед.}}{\text{с}}. \tag{3.16}$$

Если длительность выдачи знака не зависит от состояния источника, для всех знаков одинакова и равна τ, то $\overline{\tau} = \tau$. Выражение для $\overline{H}(X)$ принимает вид:

$$\overline{H}(X) = \frac{H(X)}{\tau} \ \frac{\text{дв.ед.}}{\text{с}}. \tag{3.17}$$

В этом случае поток информации максимальный, если энтропия источника на символ максимальна. Для увеличения потока информации необходимо по возможности уменьшить среднюю длительность символов $\overline{\tau}$. С этой целью, например, необходимо, чтобы длительность тех символов, вероятность появления которых больше, была меньше, чем для символов, вероятность появления которых относительно велика. Таким образом, для получения большого потока информации на выходе источника необходимо не только обеспечить по воз-

можности большую энтропию на символ, но и правильно выбрать длительность разных символов.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение эргодическому источнику.
2. Запишите выражение для энтропии эргодического источника, когда корелятивные связи имеются между двумя и тремя символами.
3. Приведите диаграмму переходов, когда источник имеет четыре характерных состояния.
4. Перечислите свойства энтропии эргодических источников.
5. Что характеризует избыточность источника сообщений?
6. Что понимается под потоком информации источника сообщений?

4. ИСТОЧНИКИ НЕПРЕРЫВНЫХ СООБЩЕНИЙ

4.1. Дифференциальная энтропия

Источники информации, множество возможных состояний которых составляют континуум, называют непрерывными.

Во многих случаях они преобразуются в дискретные посредством использования устройств дискретизации и квантования. Вместе с тем существует немало и таких систем, в которых информация передаётся и преобразуется непосредственно в форме непрерывных сигналов. Примерами могут служить системы телеизмерений с частотным разделением сигналов.

Основные информационные характеристики источников непрерывных сообщений следующие: энтропия, условная энтропия, эпсилон – энтропия, эпсилон – производительность, избыточность, объём информации.

Формулу для энтропии источника непрерывных сообщений получают путем предельного перехода из формулы (2.14) для энтропии дискретного источника. С этой целью разобьём диапазон изменения непрерывной случайной величины X, характеризующейся плотностью распределения вероятностей $W(X)$, на конечное число m малых интервалов шириной Δx (рис. 4.1).

При реализации любого значения x, принадлежащего интервалу $[x_i, x_i+\Delta x]$, будем считать, что реализовалось значение x_i дискретной случайной величины X. Поскольку Δx мало, то вероятность $P(x_i \le x \le x_i + \Delta x)$ реализации значения x из интервала $[x_i, x_i+\Delta x]$ равна

$$P(x_i < x < x_i + \Delta x) = \int_{x_i}^{x_i+\Delta x} W(x)dx \approx W(x)\Delta x.$$

Тогда энтропия дискретной случайной величины $\overline{X}$ может быть записана в виде

$$H(\overline{X}) = -\sum_{i=1}^{m} W(x)\Delta x \log W(x)\Delta x = -\sum_{i=1}^{m} W(x)\Delta x \log W(x) - \sum_{i=1}^{m} W(x)\Delta x \log \Delta x =$$

$$= -\sum_{i=1}^{m} W(x)\Delta x \log W(x) - \log \Delta x, \qquad (4.1)$$

так как $\sum_{i=1}^{m} W(x)\,\Delta x = 1$.

По мере уменьшения Δx $P(x_i < x < x + \Delta x)$ все больше приближается к вероятности $P(x_i)$, равной нулю, а свойства дискретной величины $\overline{X}$ – к свойствам непрерывной случайной величины X.

В результате предельного перехода при $\Delta x \to 0$ получено

$$H(X) = -\lim_{\Delta x \to 0} H(\overline{X}) = -\int_{-\infty}^{\infty} W(x) \log W(x) dx - \lim_{\Delta x \to 0} \log \Delta x. \qquad (4.2)$$

Первый член выражения (4.2) зависит только от закона распределения непрерывной случайной величины X и имеет такую же структуру, как энтропия дискретного источника. Второй член $\lim_{\Delta x \to 0} \log \Delta x$ стремится к бесконечности, это полностью соответствует интуитивному представлению о том, что неопределенность выбора из бесконечно большого числа возможных состояний (значений) бесконечно велика.

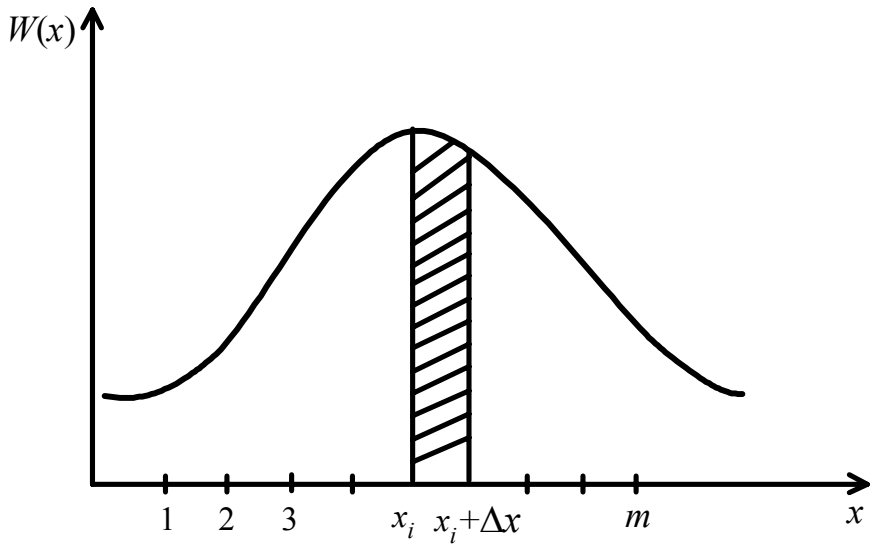


Рис. 3.1. Зависимость плотности распределения вероятностей случайной величины

Чтобы избавить теорию от бесконечности, имеется единственная возможность – ввести относительную меру неопределенности исследуемой непрерывной случайной величины X по отношению к заданной X_0 . В качестве заданной величины X_0 возьмем непрерывную случайную величину, равномерно распределенную на интервале с шириной $\varepsilon = \beta - \alpha$. Тогда её плотность вероятности $W(X_0) = 1/\varepsilon$, а энтропия

$$H(X_0) = -\int_{\alpha}^{\beta} \frac{1}{\varepsilon} \log \frac{1}{\varepsilon} dx - \lim_{\Delta x \to 0} \log \Delta x = \log \varepsilon - \lim_{\Delta x \to 0} \log \Delta x .$$

Положив для простоты записи ε = 1, составим разность

$$H_{\Delta}(X) = H(X) - H(X_0) = -\int_{-\infty}^{\infty} W(X) \log W(X) dx, \qquad (4.3)$$

которая показывает, насколько неопределенность непрерывной случайной величины X с законом распределения $W(X)$ больше $[H_{\Delta}(X) > 0]$ или меньше $[H_{\Delta}(X) < 0]$ неопределенности случайной величины, распределенной равномерно на интервале ε = 1. Поэтому величину

$$H_{\Delta}(X) = \int_{-\infty}^{\infty} W(X) \log W(X) dx \qquad (4.4)$$

называют относительной дифференциальной энтропией или просто дифференциальной энтропией непрерывного источника информации (непрерывного распределения случайной величины X). В отличие от энтропии источников дискретных сообщений $H_{\Delta}(X)$ может принимать положительные, отрицательные и нулевые значения. Величиной $H_{\Delta}(X)$ можно характеризовать информационные свойства источников непрерывных сообщений.

Аналогично, используя операции квантования и предельного перехода, найдем выражение для условной энтропии непрерывного источника сообщений.

$$H(X/Y) = -\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W(x,y) \log \frac{W(x,y)}{W(y)} dx dy - \lim_{\Delta x \to 0} \log \Delta x . \qquad (4.5)$$

Обозначим первый член через $H_{\Delta}(X/Y)$:

$$H_{\Delta}(X/Y) = -\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W(x,y) \log \frac{W(x,y)}{W(y)} dx dy . \qquad (4.6)$$

Эта величина конечна и называется относительной дифференциальной условной энтропией, или просто дифференциальной условной энтропией непрерывного источника. Она характеризует неопределенность выбора непрерывной случайной величины X при условии, что известны результаты реализации значений другой статистически связанной с ней непрерывной случайной величины Y, и по сравнению со средней неопределенностью выбора случайной величины X_0, изменяющейся в диапазоне, равном единице, и имеющей равномерное распределение вероятностей.

4.2. Свойства дифференциальной энтропии

1. Величина $H_{\text{Д}}(X)$ изменяется при изменении масштаба измерения X.

Изменим масштаб случайной величины X в k раз, оставив неизменным масштаб равномерно распределенной в единичном интервале случайной величины X_0, принятой за эталон. Если $x_i = kx$, то $W(x_i) = \frac{W(x)}{k}$. Тогда

$$H'_{\Delta}(X) = -\int_{-\infty}^{\infty} W(x_i)\log W(x_i)dx_i = -\int_{-\infty}^{\infty} \frac{W(x)}{k}\log\left(\frac{W(x)}{k}\right)kdx =$$

$$= -\int_{-\infty}^{\infty} W(x)\log W(x)dx + \log k\int_{-\infty}^{\infty} W(x)dx = H_{\Delta}(X) + \log k. \qquad (4.7)$$

Если одновременно изменить масштаб X_0, соотносительная неопределенность также изменится, так как значение эталона будет уже иным.

Из относительности дифференциальной энтропии следует, что энтропия может принимать положительные, отрицательные и нулевые значения.

2. Дифференциальная энтропия не зависит от конкретных значений случайной величины X и, в частности, от изменения всех её значений на постоянное. Действительно, масштаб X при этом не меняется и справедливо равенство

$$H_{\Delta}(X + \theta) = -\int_{-\infty}^{\infty} W(x+\theta)\log W(x+\theta)d(x+\theta) =$$

$$= -\int_{-\infty}^{\infty} W(x)\log W(x)dx = H_{\Delta}(X). \qquad (4.8)$$

3. Если единственным ограничением для случайной величины X является область её возможных значений $[\alpha, \beta]$, то максимальной дифференциальной энтропией обладает равномерное распределение вероятностей в этой области, т.е. при $W(X) = (\beta - \alpha)^{-1}$:

$$H_{\Delta}(X) = \log(\beta - \alpha). \qquad (4.9)$$

4. Если ограничения на область значений непрерывной случайной величины X отсутствуют, но известно, что дисперсия её ограничена, то максимальной дифференциальной энтропией, равной

$$H_{\Delta}(X)_{\max} = \log\sigma_x\sqrt{2\pi e}, \qquad (4.10)$$

обладает нормальное (Гауссовское) распределение величин X:

$$W(X) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-x^2/2\sigma_x^2}. \qquad (4.11)$$

5. Соотношения для дифференциальной энтропии объединения статистически зависимых непрерывных источников аналогичны соответствующим формулам для дискретных источников:

$$H_{\Delta}(X,Y) = H_{\Delta}(X) + H_{\Delta}(Y/X) = H_{\Delta}(Y) + H_{\Delta}(X/Y), \qquad (4.12)$$

где $H_{\Delta}(X,Y) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W(x,y)\log W(x,y)\,dxdy$.

6. Если статистические связи между X и Y отсутствуют, то

$$H_{\Delta}(X,Y) = H_{\Delta}(X) + H_{\Delta}(Y). \qquad (4.13)$$

4.3. Эпсилон - энтропия источника сообщений

Реальная чувствительность приемных устройств, органов чувств человека и разрешающая способность различных информационно-измерительных систем ограничены. Поэтому воспроизводить непрерывные сообщения абсолютно точно не требуется. Наличие помех и искажений сигналов в реальных каналах делает точное воспроизведение сообщений невозможным. Поэтому введём понятие эпсилон–энтропии. Эпсилон–энтропия – это то среднее количество информации в одном независимом отсчете непрерывного случайного процесса $X(t)$, которое необходимо для воспроизведения этого сигнала с заданной среднеквадратичной погрешностью ε_o.

Рассмотрим подробнее сущность этого понятия. Предположим, что передавался сигнал $X(t)$, а был принят сигнал $Y(t)$. Пусть в канале действует аддитивная помеха $E(t)$, тогда $Y(t) = X(t) + E(t)$. Расстояние между сигналами $X(t)$ и $Y(t)$ определяется величиной

$$\varepsilon^2 = \frac{1}{T}\int_0^T (Y(t) - X(t))^2 dt = \frac{1}{T}\int_0^T E^2(t)dt\,, \qquad (4.14)$$

где T – длительность сигналов.

Если $\varepsilon^2 \le \varepsilon_0^2$, то сигналы называют ε_0-близкими.

В соответствии с определением эпсилон-энтропии можно записать, что

$$H(X)_\varepsilon = \min I(Y,X) = H_\Delta(X) - \max H_\Delta(X/Y)\,. \qquad (4.15)$$

Так как $X(t) = Y(t) - E(t)$, то условная энтропия $H_\Delta(X/Y)$ при принятом $Y(t)$ полностью определяется "шумом" воспроизведения $E(t)$. Поэтому

$$\max H_\Delta(X/Y) = \max H_\Delta(E). \qquad (4.16)$$

Учитываем, что мощность помехи ограничена величиной ε_0^2, тогда максимальная энтропия помехи, отнесенная к одному отсчету, определяется по формуле (4.10)

$$\max H_\Delta(E) = \log e_0\sqrt{2\pi e} = \log y_E\sqrt{2\pi e}\,, \qquad (4.17)$$

где σ_E – среднеквадратическое значение помехи.

С учетом (4.17)

$$H(X)_\varepsilon = H_\Delta(X) - \log e_0\sqrt{2\pi e}\,. \qquad (4.18)$$

Эпсилон-энтропия имеет максимальное значение, когда процесс X(t) также является гауссовским:

$$\max H(X)_e = \log y_X\sqrt{2\pi e} - \log y_E\sqrt{2\pi e} = 0{,}5\log\left(\frac{y_X^2}{y_E^2}\right) = 0{,}5\log\frac{P_X}{P_E}\,. \qquad (4.19)$$

Отношение сигнал/шум $\sigma_X^2/\sigma_E^2 = P_X/P_E$ характеризует то количество полученной информации, при котором принятый сигнал $Y(t)$ и переданный сигнал $X(t)$ "похожи" в среднеквадратичном смысле с точностью до $\varepsilon_0^2 = \sigma_E^2$. В формуле (4.19) значение эпсилон-энтропии определено для одного независимого отсчета.

4.4. Эпсилон-производительность источника

Если источник выдает независимые отсчеты сигнала $X(t)$ в дискретные моменты времени со скоростью $v_\tau = 1/\Delta t$, где интервал дискретизации $\Delta t = 1/2\Delta F_m$ (ΔF_m – полоса частот сигнала $X(t)$), то эпсилон-производительность источника (эпсилон-энтропия, приходящаяся на единицу времени)

$$H'(X)_\varepsilon = V_\tau H(X)_\varepsilon = V_\tau (H_\Delta(X) - \log\sqrt{2\pi e \varepsilon_0{}^2}) \quad \frac{\text{бит}}{\text{с}}. \quad (4.20)$$

Если время непрерывное, то

$$H'(X)_\varepsilon = 2\Delta F_{\text{m}} (H_\Delta(X) - \log\sqrt{2\pi e \varepsilon_0{}^2}) \quad \frac{\text{бит}}{\text{с}}. \quad (4.21)$$

Максимальное значение эпсилон-производительность источника имеет, когда сигнал $X(t)$ является гауссовским (4.19):

$$\max H'(X)_\varepsilon - \frac{V_\tau}{2} \log\left(\frac{\sigma_X^2}{\varepsilon_0^2}\right) \quad \frac{\text{бит}}{\text{с}}, \quad (4.22)$$

$$\max H'(X)_\varepsilon = \Delta F_{\text{m}} \log\left(\frac{\sigma_X^2}{\varepsilon_0^2}\right) \quad \frac{\text{бит}}{\text{с}}. \quad (4.23)$$

За время T существования сигнала максимальный объем V информации, выданной источником, составит

$$\max V = \max H'(X)_\varepsilon \cdot T = \Delta F_m T \log\left(\frac{\sigma_X^2}{\varepsilon_0^2}\right) \text{ бит}. \quad (4.24)$$

Объем сигнала – это максимальное количество информации, которое сигнал может переносить.

4.5. Избыточность источника непрерывных сигналов

Избыточность определяют так же, как и для источника дискретных сигналов:

$$R_X = 1 - \frac{H(X)_\varepsilon}{\max H(X)_\varepsilon} = 1 - \frac{H_\Delta(X) - \log\sqrt{2\pi e \varepsilon_0^2}}{0{,}5 \log\left(\frac{\sigma_X^2}{\varepsilon_0^2}\right)}. \quad (4.25)$$

Избыточность источника равна нулю только в случае, когда распределение сигнала является гауссовским.

При определении эпсилон-характеристик источников непрерывных сигналов критерием близости служило среднеквадратичное отклонение одного сигнала от другого. Если выбрать другую меру близости сигналов – другую метрику пространства сигналов, можно получить другие эпсилон-характеристики ис-

точников. Наибольшее распространение получил среднеквадратический критерий близости сигналов.

4.6. Количество информации

Количество информации, содержащееся в одной непрерывной случайной величине относительно другой, определим как разность безусловной и условной дифференциальных энтропий:

$$I(X,Y) = H_{\Delta}(X) - H_{\Delta}(X / Y) = -\int_{-\infty}^{+\infty} w(x)\log(w(x))dx +$$

$$+\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} w(x,y)\log(w(x / y))dxdy = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} w(x,y)\log\frac{w(x,y)}{w(x)w(y)}dxdy. \quad (4.26)$$

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какова особенность определения энтропии непрерывного источника информации?
2. Дайте определение дифференциальной энтропии.
3. Перечислите свойства дифференциальной энтропии.
4. Какие распределения обладают максимальной дифференциальной энтропией при ограничении на область изменения случайной величины и при ограничении на дисперсию случайной величины?
5. Что такое эпсилон-энтропия источника непрерывных сообщений?
6. Дайте определение эпсилон-производительности источника.
7. Запишите выражение для определения источника непрерывных сообщений.

5. ИНФОРМАЦИОННЫЕ ХАРАКТЕРИСТИКИ НЕПРЕРЫВНЫХ КАНАЛОВ

5.1 Скорость передачи информации и пропускная способность

Непрерывным каналом называется канал, предназначенный для передачи непрерывных сообщений. Канал считается заданным, если известны статистические данные о сообщениях на его входе и выходе и ограничения, накладываемые на входные сообщения физическими характеристиками канала.

При рассмотрении информационных характеристик канала: (скорости передачи, пропускной способности, коэффициента использования) применяют

модель реального канала, называемую гауссовым каналом, предполагая, что по каналу передаются сигналы с постоянной средней мощностью, статистические связи между сигналами и помехой отсутствуют (аддитивная помеха), ширина спектра сигнала и помехи ограничены полосой пропускания канала, а в канале действует флуктуационная помеха ограниченной мощности с равномерным частотным спектром и нормальным распределением амплитуд ("белый шум").

Если $X(t)$ рассматривать как переданный сигнал, $Y(t)$ – как принятый, а $E(t)$ – как аддитивную помеху в непрерывном канале, то скорость передачи информации по непрерывному каналу (среднее количество информации, которое можно передать по каналу в единицу времени) равна

$$R_t = V_\tau(H_\Delta(X) - H_\Delta(X/Y)) = V_\tau(H_\Delta(Y) - H_\Delta(Y/X)), \quad (5.1)$$

где

$$H_\Delta(X) = -\int_{-\infty}^{+\infty} w(x)\log w(x)dx; \quad (5.2)$$

$$H_\Delta(Y) = -\int_{-\infty}^{+\infty} w(y)\log w(y)dy; \quad (5.3)$$

$$H_\Delta(X/Y) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(x,y)\log w(x/y)dxdy; \quad (5.4)$$

$$H_\Delta(Y/X) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(y,x)\log w(y/x)dxdy. \quad (5.5)$$

Скорость передачи в предположении, что передаваемые сообщения имеют структуру "белого шума", составит

$$R_t = 2\Delta F_m(\log\sqrt{2\pi e(\sigma_X^2 + \sigma_E^2)} - \log\sqrt{2\pi e\sigma_E^2}) =$$

$$= \Delta F_m \log\left(\frac{\sigma_X^2}{\sigma_E^2} + 1\right) = \Delta F_m \log\left(\frac{P_x}{P_E} + 1\right), \quad (5.6)$$

где P_x/P_E – отношение средних мощностей сигнала и помехи на выходе приёмника; $ДF_m$ – полоса частот передаваемого сообщения.

Пропускная способность (максимальное значение скорости передачи информации по каналу) непрерывного сигнала:

$$C = \max R_t = V_\tau \max\left(H_\Delta(X) - H_\Delta(X/Y)\right) = V_\tau \max\left(H_\Delta(Y) - H_\Delta(Y/X)\right). \quad (5.7)$$

Для гауссова непрерывного канала с дискретным временем

$$C = V_\tau \max (H_\Delta(Y) - H_\Delta(E)). \qquad (5.8)$$

Учитывая, что

$$\max H_\Delta(Y) = \log\sqrt{2\pi e(\sigma_X^2 + \sigma_E^2)}, \qquad (5.9)$$

$$\max H_\Delta(E) = \log\sqrt{2\pi e\sigma_E^2}, \qquad (5.10)$$

тогда

$$C = 0{,}5\, V_\tau \log\left(1 + \frac{\sigma_X^2}{\sigma_E^2}\right). \qquad (5.11)$$

Если в канале нет искажений и помех, то σ_E^2 можно рассматривать как мощность шумов квантования при дискретной передаче непрерывных сигналов. В канале с помехами мощность шумов квантования складывается с мощностью помех, следовательно, в этом случае σ_E^2 необходимо рассматривать как суммарную мощность помехи и шума квантования. Мощность шума квантования при равномерном квантовании:

$$\sigma_{\text{ш}}^2 = \frac{\Delta U^2}{12}, \qquad (5.12)$$

где ΔU – шаг квантования.

Для непрерывного канала с непрерывным временем $V_\tau = 2\Delta F_k$ и формула (5.11) переходит в известную формулу Шеннона для пропускной способности гауссова непрерывного канала с флуктуационной помехой:

$$C = \Delta F_k \log(1 + \frac{\sigma_X^2}{\sigma_E^2}) = \Delta F_k \log(1 + \frac{P_X}{P_E}), \qquad (5.13)$$

где ΔF_k – полоса пропускания канала; $\frac{P_X}{P_E}$ – отношение средних мощностей сигнала и помехи на входе приёмника.

Из (5.13) следует, что одну и ту же пропускную способность можно получить при различных соотношениях ΔF_k и $\frac{\sigma_X^2}{\sigma_E^2}$. Кроме того, выражение (5.13) указывает теоретический предел скорости передачи информации по каналу связи при ограниченной средней мощности передаваемых сигналов и при наличии аддитивной помехи в виде "белого шума" с ограниченным спектром.

Так как энергетический спектр помехи типа "белого шума" равномерен в пределах от 0 до ΔF_k, мощность P_E можно выразить через удельную мощность $P_{\text{ОШ}}$ на единицу частоты. Тогда выражение (5.13) примет вид

$$C = \Delta F_{\text{k}} \log\left(1 + \frac{P_{\text{X}}}{P_{\text{ош}} \cdot \Delta F}\right). \qquad (5.14)$$

При расширении полосы пропускания канала ΔF пропускная способность увеличивается, но стремиться к конечному пределу:

$$C_{\text{m}} = \underbrace{\lim}_{\Delta F_k \to \infty} C = \underbrace{\lim}_{\Delta F_k \to \infty} \cdot \Delta F_k \log\left(1 + \frac{P_X}{P_{ош} \cdot \Delta F}\right) = 1,443 \frac{P_X}{P_{\text{ош}}}. \qquad (5.15)$$

Это ограничение, вносимое помехой с уровнем мощности $P_{\text{ОШ}}$, которое не может быть превышено без увеличения мощности сигнала.

Если плотность распределения $w(x)$ непрерывных сообщений, вырабатываемых источником информации, отличается от гауссовской, то скорость передачи информации будет меньше.

Необходимо отметить существенную разницу R_t и C. Пропускная способность C характеризует канал, его предельные возможности независимо от системы источник-потребитель, а скорость передачи R_t характеризует некоторую конкретную систему передачи информации.

Кроме того, как следует из (5.13), если сигнал смешан с шумом, то амплитуда сигнала может быть измерена лишь с точностью до эффективного значения шума. Другими словами, неопределенность оценки точного значения амплитуды сигнала равна квадратному корню из среднего квадрата шумового напряжения. Как следует из указанных выше предположений, изменение входного сигнала меньше, чем $\delta = \sqrt{P_{\text{E}}}$, приемник не различает. Следовательно, число уровней, которое может быть различимо без ошибок, определится из выражения

$$M = \frac{\sqrt{P_X + P_E}}{\sqrt{P_E}} = \sqrt{1 + \frac{P_X}{P_E}}. \qquad (5.16)$$

Итак, наибольшее количество информации, переносимое каждым импульсом, имеющим M различных уровней, равно

$$I = \log \sqrt{1 + \frac{P_X}{P_E}} = \frac{1}{2} \log\left(1 + \frac{P_X}{P_E}\right). \qquad (5.17)$$

5.2. Согласование источников с каналами

Предельные возможности согласования источника непрерывных сообщений с непрерывным каналом определяются следующей теоремой кодирования Шеннона: если эпсилон-производительность $H'(X)_\varepsilon$ источника непрерывных сообщений меньше пропускной способности канала, то существует способ оптимального кодирования и декодирования, при котором с вероятностью, сколь угодно близкой к единице, переданное и принятое сообщения не будут отличаться в среднеквадратическом смысле более чем на ε_0^2.

При $H'(X)_\varepsilon > C$ такого способа нет. Под оптимальным кодированием непрерывных сообщений в непрерывные сигналы понимают преобразование без предварительной дискретизации по времени и квантования по уровню. Речь идёт о выборе способа аналоговой модуляции, оптимальное кодирование соответствует идеальной модуляции.

Для гауссова канала условие существования оптимального кодирования принимает вид

$$\Delta F_m \log\left(\frac{\sigma_X^2}{\varepsilon_0^2}\right) < \Delta F_k \log\left(1+\frac{\sigma_X^2}{\sigma_E^2}\right), \qquad (5.18)$$

где отношение $\frac{\sigma_X^2}{\varepsilon_0^2}$ рассматривается на выходе детектора, а величина $\frac{\sigma_X^2}{\sigma_E^2}$ – как отношение сигнал/шум на входе приёмника.

Для того чтобы оценить, как используется пропускная способность непрерывных каналов, вводят коэффициент эффективности, который определяется выражением

$$\eta = \frac{\Delta F_m \log\left(1+\left(\frac{\sigma_X^2}{\sigma_E^2}\right)_{\text{вых}}\right)}{\Delta F_k \log\left(1+\left(\frac{\sigma_X^2}{\sigma_E^2}\right)_{\text{вх}}\right)}, \qquad (5.19)$$

где $\left(\frac{\sigma_X^2}{\sigma_E^2}\right)_{\text{вых}}$ и $\left(\frac{\sigma_X^2}{\sigma_E^2}\right)_{\text{вх}}$ обозначено отношение сигнал/шум на выходе и входе приёмника соответственно.

По существу η характеризует эффективность способа модуляции. Для идеальной модуляции $\eta = 1$ и

$$\left(\frac{\sigma_X^2}{\sigma_E^2}\right)_{\text{вых}} = \left(1+\left(\frac{\sigma_X^2}{\sigma_E^2}\right)_{\text{вх}}\right)^{\frac{\Delta F_k}{\Delta F_m}} - 1. \qquad (5.20)$$

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение скорости передачи информации.
2. Запишите выражение для скорости передачи и поясните его.
3. Дайте определение пропускной способности непрерывного канала.
4. Что понимается под дисперсией помехи в канале связи, когда нет искажений и помех и, когда они имеют место?
5. Как определяется мощность шума квантования?
6. Сформулируйте теорему Шеннона для непрерывного канала связи.
7. Запишите условие существования оптимального кодирования.

6. ИНФОРМАЦИОННЫЕ ХАРАКТЕРИСТИКИ ДИСКРЕТНЫХ КАНАЛОВ СВЯЗИ

6.1. Информационная модель канала и основные характеристики

Дискретным каналом называется совокупность средств, предназначенных для передачи дискретных сигналов.

Для анализа информационных возможностей удобно пользоваться информационной моделью канала связи, представленной на рис. 5.1.

Источник информации создаёт сообщения, состоящие из последовательности знаков алфавита источника $Z = (Z_1, Z_2,\ldots, Z_n)$, которые в кодирующем устройстве преобразуются в последовательность символов. Объём алфавита символов $X=(X_1, X_2,\ldots, X_m)$, как правило, меньше объёма алфавита знаков, но они могут и совпадать. В результате модуляции каждой последовательности символов ставится в соответствие сложный сигнал. Множество сложных сигналов конечно. Они различаются числом, составом и взаимным расположением элементарных сигналов. В результате действия помех сигнал на приёмной стороне может отличаться от переданного. Помехи имеют случайный характер и подчиняются статистическим законам. Удобно условно считать, что помехи создаются некоторым воображаемым источником помех и поступают в линию связи в виде мешающего сигнала E. Приёмная сторона содержит демодулятор, где сигналы преобразуются в последовательность символов $Y = (y_1, y_2,\ldots, y_m)$, декодирующее устройство, выполняющее ответные функции кодированию, и приемник информации, перерабатывающий принятые сообщения $V = (v_1,v_2,\ldots,v_n)$.

С математической точки зрения дискретный канал можно определить алфавитом единичных элементов на его входе $x_i(i=1,2,...,m)$ и выходе $y_j(j=1,2,...,m)$, а также вероятностями перехода единичного элемента одного вида (передаваемого) в элемент того же вида или другого вида в пункте приёма

$P(y_j/x_i)$. Значения вероятностей $P(y_j/x_i)$ зависят от характера ошибок в дискретном канале, т.е. от интенсивности ошибок и их статистического распределения во времени. Если при передаче *i*-го единичного элемента принят элемент такого же вида ($i = j$), то считается, что ошибки нет.

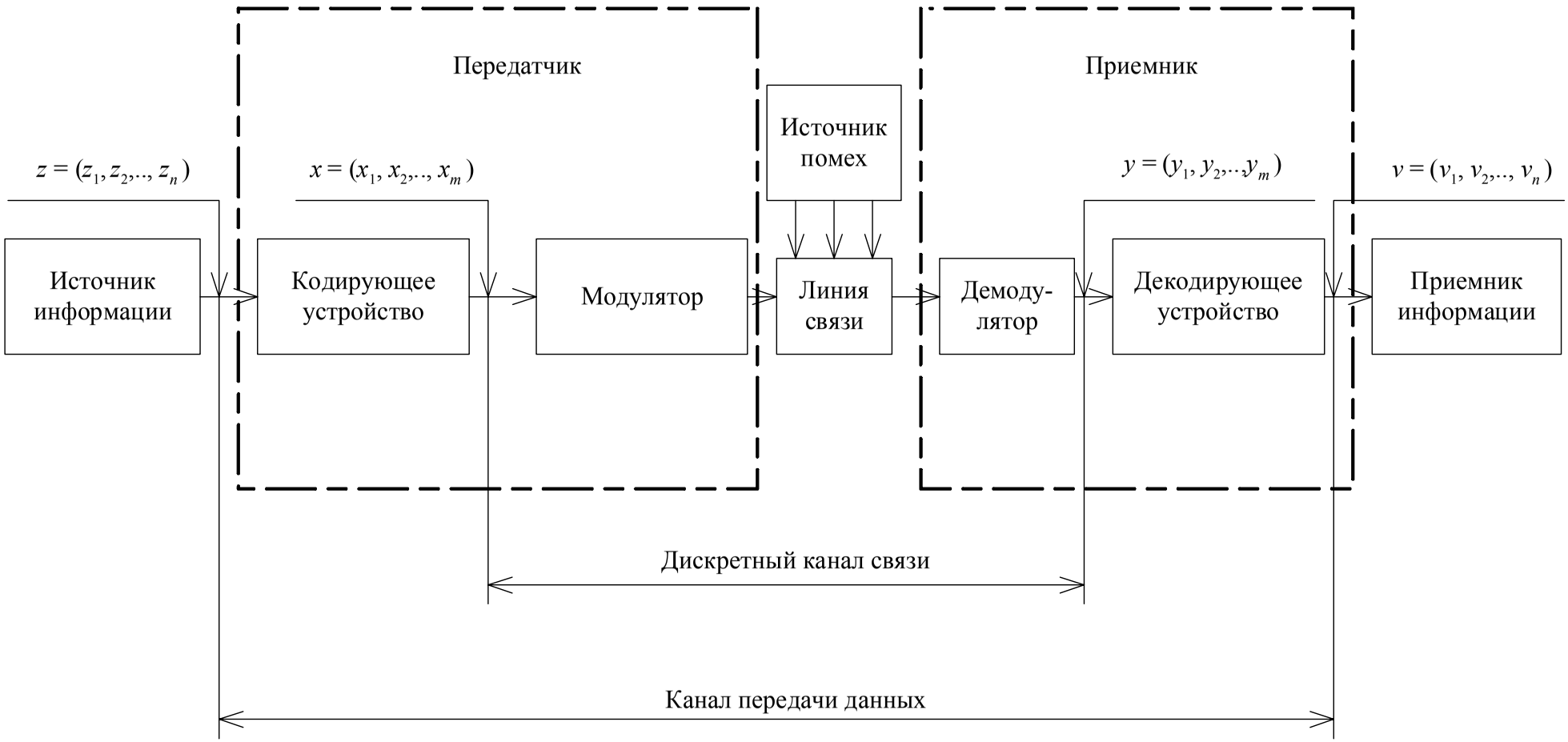


Рис. 6.1. Информационная модель канала связи

Если при передаче *i*-го элемента принят элемент нового вида, который не предусмотрен алфавитом, y_j (например, $i = m+1$), то его можно использовать для стирания принятого знака. При $i \neq j$ и $i \neq m+1$ считают, что произошла ошибка.

На рис. 6.2 и 6.3 приведены модели бинарного стирающего канала при отсутствии и при наличии трансформации символов соответственно.

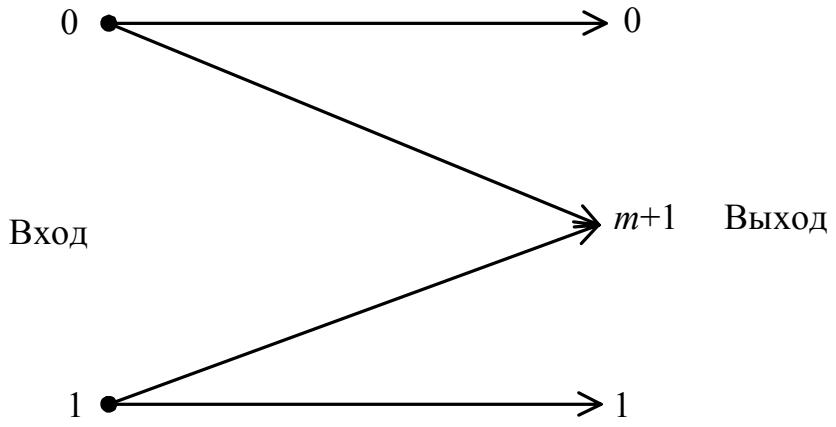


Рис. 6.2. Модель бинарного стирающего канала при отсутствии трансформации символов

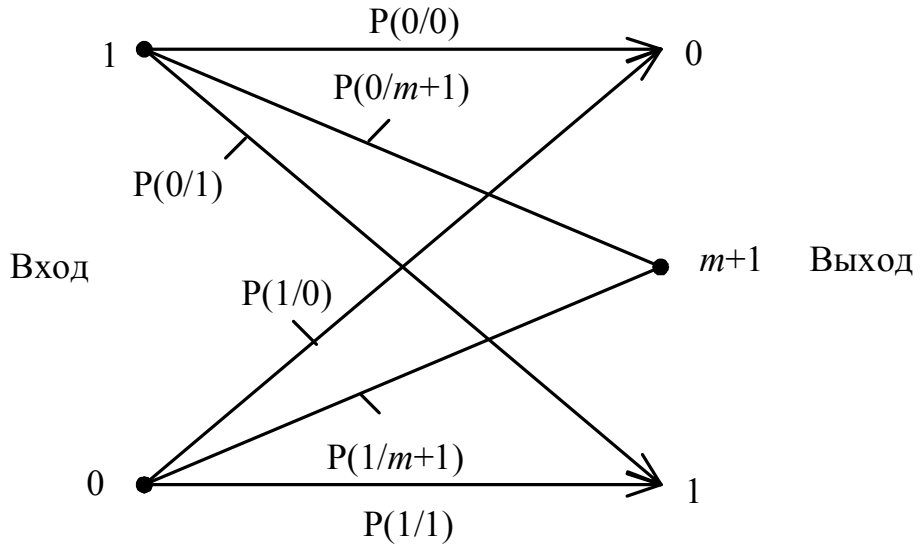
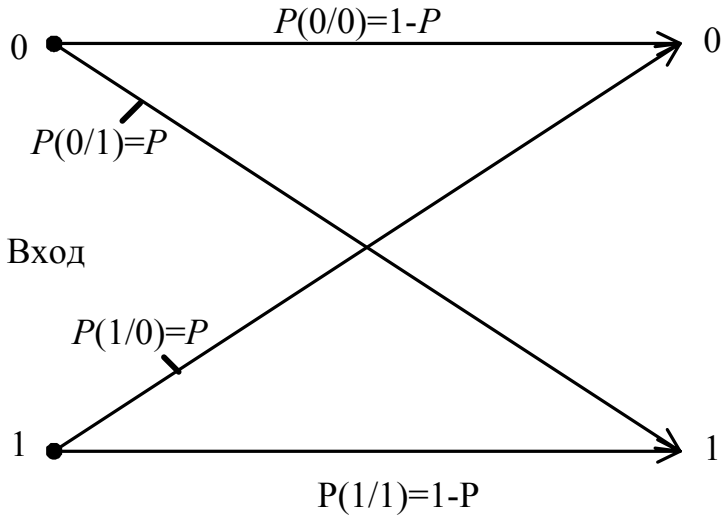


Рис. 6.3. Модель бинарного стирающего канала при наличии трансформации символов

Дискретные каналы классифицируются в зависимости от свойств вероятности перехода $P(y_j/x_i)$. Каналы, в которых $P(y_j/x_i)$, не зависят от времени для любых i и j, называются стационарными. Если $P(y_j/x_i)$ зависят от времени, то каналы называются нестационарными. Каналы, в которых $P(y_j/x_i)$ не зависят от значений ранее принятых элементов, называются каналами без памяти. При зависимости $P(y_j/x_i)$ от значений ранее принятых элементов возникает канал с памятью. Каналы, в которых вероятность перехода $P(y_j/x_i)$ = const не зависит от i и j, называются симметричными. В противном случае канал становится несимметричным. Заметим, что большинство каналов, образуемых по кабельным и радиорелейным линиям связи, симметричны и обладают памятью. Каналы космической связи симметричны и памятью не обладают.

На рис. 6.4. приведена модель бинарного канала без памяти.



при: $P(0/1) = P(1/0) = q$ – канал симметричный,

$P(0/1) \neq P(1/0)$ – канал несимметричный

Рис. 6.4. Модель бинарного канала без памяти

Для организации эффективной передачи информации по каналу требуется решение следующих задач: определение максимально возможной скорости передачи информации по каналу; разработка кодов, позволяющих увеличить скорость передачи информации; согласование канала с источником с целью передачи информации с минимальными потерями. Решение этих задач зависит от свойств источников, уровня и характера помех.

Если уровень помех мал и искажениями сигнала можно пренебречь, канал связи называется каналом без помех.

Для характеристики дискретного канала связи используют два понятия скорости передачи: технической и информационной.

Под технической скоростью передачи V_τ, называемой также скоростью манипуляции, подразумевают число элементарных символов (сигналов), передаваемых по каналу в единицу времени. Она зависит от свойств линии связи и быстродействия аппаратуры канала и определяется из выражения

$$V_\tau = \frac{1}{\bar{\tau}} \text{ Бод}, \tag{6.1}$$

где $\bar{\tau}$ – среднее значение длительности символа.

При одинаковой продолжительности τ всех символов, передаваемых в канал $\tau = \bar{\tau}$.

Единицей измерения технической скорости служит бод-скорость, при которой за одну секунду передаётся один символ. Информационная скорость

определяется средним количеством информации, которая передаётся по каналу в единицу времени. Она зависит как от характеристик данного канала связи (объём алфавита используемых символов, техническая скорость их передачи, статистические свойства помех в линии), так и от вероятностей поступающих на вход символов и их статистической взаимосвязи.

При известной скорости манипуляции V_τ скорость передачи информации по каналу R_t определяется из выражения

$$R_t = V_\tau I(Y,X), \tag{6.2}$$

где $I(Y,X)$ – среднее количество информации, переносимое одним символом.

Для теории и практики важно выяснить, до какого предела и каким путем можно повысить скорость передачи информации по конкретному каналу связи, т.е. определить пропускную способность канала.

Пропускная способность канала C равна той максимальной скорости передачи информации по данному каналу, которой можно достигнуть при самых совершенных способах передачи и приёма:

$$C = \max R_t = V_\tau \max I(Y,X). \tag{6.3}$$

Пропускная способность канала и скорость передачи по каналу измеряются числом двоичных единиц информации в секунду (дв.ед.\с).

Рассмотрим вопросы передачи сообщений для дискретного канала без помех и дискретного канала с помехами, для чего предварительно рассмотрим связь между энтропией источника и энтропией сообщения.

6.2. Энтропия источника и энтропия сообщения

Пусть источник информации выдаёт дискретные сообщения Z. С помощью кодирующего устройства каждое сообщение превращается в код. Множество символов кода обозначим через X. Если исследуется канал связи, то можно не обращаться к источнику информации, а рассматривать лишь источник символов (кодирующее устройство). Тогда возникает необходимость связать свойства

источника и отправителя. Эта связь возможна через энтропию.

Под энтропией сообщения будем понимать количество информации, содержащееся в любом усреднённом сообщении. Тогда

$$H(Z) = -\sum_{j=1}^{n} P(z_j) \log_2 P(z_j) \quad \frac{\text{двоичных единиц}}{\text{сообщение}}. \tag{6.4}$$

– усреднённая энтропия сообщения. Соответственно энтропия источника, или количество информации, содержащееся в одном символе сообщения:

$$H(X) = -\sum_{j=1}^{m} P(x_j) \log_2 P(x_j) \quad \frac{\text{двоичных единиц}}{\text{символ}}. \tag{6.5}$$

Пример. Пусть передаётся четыре равновероятных сообщения двоичным не избыточным кодом. Сообщения отображаются кодом 00, 01,10,11. Найдём энтропию сообщения:

$$H(Z) = -\sum_{4} \frac{1}{4} \log_2 (\frac{1}{4}) = 2 \quad \frac{\text{дв.ед.}}{\text{сообщ.}}$$

и энтропию источника

$$H(X) = -\sum_{2} \frac{1}{2} \log_2 (\frac{1}{2}) = 1 \quad \frac{\text{дв.ед.}}{\text{сообщ.}}.$$

Из примера видно, что каждый символ несёт одну двоичную единицу информации.

Разделим $H(Z)$ на $H(X)$ и получим число элементов в коде, т.е. $H(Z)/H(X) = n$. Если данное условие соблюдается, то код называется оптимальным, в противном случае в коде возникает избыточность, и он становится неоптимальным для канала без шума. Для получения оптимального кода необходимо, чтобы символы в нём встречались с равной вероятностью.

6.3. Дискретный канал без помех

В любом реальном канале всегда присутствуют помехи. Однако если их уровень настолько мал, что вероятность искажения практически равна нулю, можно считать, что все сигналы передаются не искажёнными. В этом случае среднее количество информации, переносимое одним символом, определяется по формуле

$$I(Y,X) = I(X,Y) = H(X), \tag{6.6}$$

а максимальное значение

$$\max I(Y, X) = H_m(X), \tag{6.7}$$

где $H_m(X)$ – максимальная энтропия источника сигналов, получающаяся при равномерном распределении вероятностей символов алфавита источника

$$P(X_1) = P(X_2) = \ldots = P(X_m) = 1/m. \tag{6.8}$$

Но максимальная энтропия выражается в единицах информации на символ сигнала, как

$$H_m(X) = \log_a m. \tag{6.9}$$

Следовательно, пропускная способность дискретного канала без помех в единицах информации за единицу времени равна

$$C = V_\tau \log_a m. \tag{6.10}$$

Шенноном сформулирована основная теорема о кодировании, которая утверждает, что если источник информации имеет энтропию $H(Z)$ единиц информации на символ сообщения, а канал связи обладает пропускной способностью C единиц информации в единицу времени, то:

1) сообщения, вырабатываемые источником, всегда можно закодировать так, чтобы скорость Vz их передачи была сколь угодно близкой к

$$Vzm = \frac{C}{H(Z)}, \tag{6.11}$$

где Vzm измеряется в символах сообщения за единицу времени;

2) не существует метода кодирования, позволяющего сделать эту скорость больше чем Vzm.

Согласно сформулированной теореме существует метод кодирования, позволяющий при $H'(Z) < C$ и $H'(Z) = C$ передавать всю информацию, вырабатываемую источником, а при $H'(Z) > C$ такого метода не существует, где $H'(Z) = V_Z H(Z)$ – поток информации.

6.4. Дискретный канал с помехами

Дискретный канал с помехами характеризуется условными вероятностями $P(y_j/x_i)$ того, что будет принят сигнал y_j, если передан x_i, т.е. матрицей

$$\left\| \begin{array}{llll} P(y_1/x_1) & P(y_2/x_1) & \ldots & P(y_m/x_1) \\ P(y_1/x_2) & P(y_2/x_2) & \ldots & P(y_m/x_2) \\ P(y_2/x_2) & P(y_2/x_m) & \ldots & P(y_m/x_m) \end{array} \right\|, \tag{6.12}$$

при отсутствии помех все $P(y_j/x_i)$ при $j \# i$ равны 0 и при $j = i$ равны 1.

Среднее количество информации на символ, получаемое при приёме одного элементарного сигнала равно:

$$I(Y,X) = H(Y) - H(Y/X). \tag{6.13}$$

В случае независимости отдельных символов сигнала энтропия на выходе линии

$$H(Y) = -\sum_{i=1}^{m} P(y_i) \log P(y_i), \tag{6.14}$$

предполагается, что число букв алфавита $Y = (y_1, y_2, \ldots, y_m)$ равно числу букв алфавита $X = (x_1, x_2, \ldots, x_m)$ и равно, следовательно, m.

Средняя условная энтропия:

$$H(\frac{Y}{X}) = -\sum_{i=1}^{m} P(x_i) \sum_{j=1}^{m} P(\frac{y_j}{x_i}) \log_a P(\frac{y_j}{x_i}) \ . \tag{6.15}$$

Пропускная способность канала высчитывается по формуле

$$C = V_\tau \max I(Y,X) \ , \tag{6.16}$$

где максимум определяется по всем возможным распределениям вероятностей, характеризующим источник сигналов

Пусть требуется определить пропускную способность канала связи, по которому передаются двоичные сигналы со скоростью V_τ, если вероятность превращения в результате действия помех каждого из этих сигналов в противоположный равна P (вероятность правильного приёма, следовательно, I-P), передаваемые символы предполагаются независимыми.

В этом случае алфавит X и алфавит Y состоит из двух символов:

$$X = (x_1, x_2), \ Y = (y_1, y_2).$$

Диаграмма (рис. 6.4) показывает возможные варианты передачи и соответствующие им вероятности.

Канал такого типа носит название ***симметричного***.

Средняя условная энтропия:

$$\begin{aligned} H(Y/X) &= -\sum_{i=1}^{2} P(x_i) \sum_{j=1}^{2} P(\frac{y_j}{x_i}) \log P(\frac{y_j}{x_i}) = \\ &= -P(x_1)[(1-P)\log(1-P) + P\log P] - \\ &\quad -P(x_2)[(1-P)\log(1-P) + P\log P] = \\ &= -[(1-P)\log(1-P) + P\log P][P(x_1) + P(x_2)], \end{aligned} \tag{6.17}$$

но $P(x_1) + P(x_2) = 1$.

Поэтому

$$H(Y/X) = -[(1 - P)\log(1 - P) + P\log P]. \quad (6.18)$$

Отсюда видно, что $H(Y/X)$ не зависит от характеристик источника, т.е. от $P(x_1)$ и $P(x_2)$, и определяется только помехами в канале передачи.

Максимальное количество информации на один символ получается при таком распределении вероятностей $P(x_i)$, при котором оказывается максимальным член $H(Y)$. Но $H(Y)$ не может превосходить величины

$$H_m(Y) = \log m = \log 2 = 1 \ \frac{\text{бит}}{\text{символ}},$$

что достигается при $P(x_1) = P(x_2)$ 1/2. Поэтому имеем

$$\max\,(I(Y,X)) = 1 + P\log P + (1\text{-}P)\,\log(1\text{-}P)$$

и, следовательно, пропускная способность:

$$C = V_\tau \max(I(Y,X)) = V_\tau\,[1 + P\log P + (1\text{-}P)\,\log(1\text{-}P)]. \quad (6.19)$$

Отсюда следует: в частности, что при $P = 0$, т.е. при отсутствии шумов в канале связи, имеем

$$C\max = V_\tau.$$

При $P = 1$ также имеем детерминированный случай, когда сигналы x_1 превращаются в сигналы x_2 и наоборот с вероятностью, равной единице. При этом пропускная способность канала также максимальная:

$$C\max = V_\tau.$$

Минимальное значение пропускная способность имеет при P1/2.

В этом случае независимо от полученных сигналов ничего нельзя сказать о том, какой сигнал был послан: имеет место такая ситуация, как если бы в линию связи вообще не посылались сигналы. Тогда, естественно, пропускная способность

$$C\min = 0.$$

6.5. Согласование характеристик сигнала и канала

Сигнал может быть охарактеризован различными параметрами. Таких па-

раметров, вообще говоря, очень много, но для задач, которые приходится решать на практике, существенно лишь небольшое их число.

Рассмотрим три основных параметра сигнала, существенных для передачи по каналу. Первый важный параметр – это время передачи сигнала $T_{x.}$ Второй характеристикой, которую приходится учитывать, является мощность P_x сигнала, передаваемого по каналу с определённым уровнем помех P_E. Чем больше значение P_x по сравнению с P_E, тем меньше вероятность ошибочного приёма. Таким образом, представляет интерес отношение P_x/P_E .Удобно пользоваться логарифмом этого отношения, называемым превышением сигнала над помехой:

$$Hx = \log \frac{P_x}{P_E}.$$

Третьим важным параметром является спектр частот F_x. Эти три параметра позволяют представить любой сигнал в трёхмерном пространстве с координатами H, T, F в виде параллелепипеда с объёмом $T_x\ F_x\ H_x$. Данное произведение носит название объём сигнала и обозначается через V_x :

$$V_x = T_x\ F_x\ H_x\ . \tag{6.20}$$

Соответственно канал связи может быть охарактеризован временем использования канала $T_к$ (т.е. временем, в течение которого канал представлен для работы), полосой пропускания $F_к$ и динамическим диапазоном $H_к$, равным разности максимально допустимого уровня сигнала в канале и уровня помех (в логарифмическом масштабе):

$$H_k = \log P_{x\max} - \log P_E = \log(\frac{P_{x\max}}{P_E}).$$

Таким образом, канал также можно охарактеризовать объёмом (ёмкостью):

$$V_k = T_k\ F_k\ H_k. \tag{6.21}$$

Для того чтобы сигнал мог быть передан по каналу, необходимо выполнение условий

$$T_x < T_k\,;\ F_x < F_k\,;\ H_x < H_k\,, \tag{6.22}$$

т.е. сигнал полностью уместится в объёме V_k. При этом, конечно, $V_x<V_k$, однако, только этого условия недостаточно. Тем не менее, если $V_x<V_k$, но условие (5.21) не выполняется, сигнал может быть определённым образом преобразован, так что передача окажется возможной.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какой канал называется каналом без помех?

2. Дайте определение скорости передачи и пропускной способности дискретного канала связи.

3. От чего зависит условная энтропия?

4. Чем определяется предельная скорость передачи по каналу элементарных сигналов?

5. Что понимается под энтропией сообщения и энтропией источника?

6. Какой код называется оптимальным для канала без шума?

7. Запишите выражения для пропускной способности дискретного канала без помех и с помехами, сравните их.

8. Приведите информационную модель канала связи.

9. Запишите выражения для пропускной способности симметричного бинарного канала поясните его.

10. Сформулируйте необходимые и достаточные условия неискаженной передачи сигнала по каналу связи.

7. КОДИРОВАНИЕ ИНФОРМАЦИИ ПРИ ПЕРЕДАЧЕ ПО ДИСКРЕТНОМУ КАНАЛУ БЕЗ ПОМЕХ

7.1. Эффективное кодирование

Сообщения, передаваемые с использованием систем связи (речь, музыка, телевизионные изображения и т.д.) в большинстве своем предназначены для непосредственного восприятия органами чувств человека и обычно плохо приспособлены для их эффективной передачи по каналам связи. Поэтому они в процессе передачи, как правило, подвергаются кодированию.

Что такое кодирование и зачем оно используется?

Под кодированием в общем случае понимают преобразование алфавита сообщения $A\{x_i\}$, ($\mathbf{i = 1,2…K}$) в алфавит некоторым образом выбранных кодовых символов $\Re\{A\{x_j\}$, ($\mathbf{j = 1,2…N}$).

Кодирование сообщений может преследовать различные цели. Например, это кодирование с целью засекречивания передаваемой информации. При этом элементарным сообщениям x_i из алфавита $A\{x_i\}$ ставятся в соответствие последовательности, к примеру, цифр или букв из специальных кодовых таблиц, известных лишь отправителю и получателю информации.

Другим примером кодирования может служить преобразование дискретных сообщений x_i из одних систем счисления в другие (из десятичной в двоич-

ную, восьмеричную и т. п., из непозиционной в позиционную, преобразование буквенного алфавита в цифровой и т. д.).

Кодирование в канале, или помехоустойчивое кодирование информации, может быть использовано для уменьшения количества ошибок, возникающих при передаче по каналу с помехами.

Наконец, кодирование сообщений может производиться с целью сокращения объема информации и повышения скорости ее передачи или сокращения полосы частот, требуемых для передачи. Такое кодирование называют экономным, безызбыточным, или эффективным кодированием, а также сжатием данных. В данном разделе будет идти речь именно о такого рода кодировании. Процедуре кодирования обычно предшествуют (и включаются в нее) дискретизация и квантование непрерывного сообщения $x(t)$, то есть его преобразование в последовательность элементарных дискретных сообщений $\{ x_{iq} \}$.

Прежде чем перейти к вопросу экономного кодирования, кратко поясним суть самой процедуры кодирования.

Любое дискретное сообщение x_i из алфавита источника $A\{ x_i \}$ объемом в K символов можно закодировать последовательностью соответствующим образом выбранных кодовых символов x_j из алфавита $R\{ x_j \}$.

Например, любое число (а x_i можно считать числом) можно записать в заданной позиционной системе счисления следующим образом:

$$x_i = a_{n-1} \cdot X^{n-1} + a_{n-2} \cdot X^{n-2} + \ldots + a_0 \cdot X^0,$$

где X - основание системы счисления; $a_0 \ldots a_{n-1}$ - коэффициенты при имеющие значение от 0 до $X - 1$.

Пусть, к примеру, значение $x_i = 59$, тогда его код по основанию $X = 8$, будет иметь вид

$$x_i = 59 = 7 \cdot 8^1 + 3 \cdot 8^0 = 73_8 .$$

Код того же числа, но по основанию $X = 4$ будет выглядеть следующим образом:

$$x_i = 59 = 3 \cdot 4^2 + 2 \cdot 4^1 + 3 \cdot 4^0 = 323_4 .$$

Наконец, если основание кода $X = 2$, то

$$x_i = 59 = 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 111011_2 .$$

Таким образом, числа 73, 323 и 111011 можно считать, соответственно, восьмеричным, четверичным и двоичным кодами числа $M = 59$.

В принципе основание кода может быть любым, однако наибольшее распространение получили двоичные коды, или коды с основанием 2, для которых размер алфавита кодовых символов $R\{ x_j \}$ равен двум, $x_j \subset 0,1$. Двоичные коды, то есть коды, содержащие только нули и единицы, очень просто формируются

и передаются по каналам связи и, главное, являются внутренним языком цифровых *ЭВМ*, то есть без всяких преобразований могут обрабатываться цифровыми средствами. Поэтому, когда речь идет о кодировании и кодах, чаще всего имеют в виду именно двоичные коды. В дальнейшем будем рассматривать в основном двоичное кодирование.

Самым простым способом представления или задания кодов являются кодовые таблицы, ставящие в соответствие сообщениям x_i соответствующие им коды (табл. 7.1).

Другим наглядным и удобным способом описания кодов является их представление в виде кодового дерева (рис. 2.1). Для того, чтобы построить кодовое дерево для данного кода, начиная с некоторой точки - корня кодового дерева - проводятся ветви - 0 или 1. На вершинах кодового дерева находятся буквы алфавита источника, причем каждой букве соответствуют своя вершина и свой путь от корня к вершине. К примеру, букве А соответствует код 000, букве В – 010, букве Е – 101 и т.д.

Код, полученный с использованием кодового дерева, изображенного на рис. 7.1, является равномерным трехразрядным кодом.

Равномерные коды очень широко используются в силу своей простоты и удобства процедур кодирования-декодирования: каждой букве – одинаковое число бит; принял заданное число бит – ищи в кодовой таблице соответствующую букву.

Таблица 7.1

Соответствие кодовых комбинаций сообщения

Буква x_i	Число x_i	Код с основанием 10	Код с основанием 4	Код с основанием 2
А	0	0	00	000
Б	1	1	01	001
В	2	2	02	010
Г	3	3	03	011
Д	4	4	10	100
Е	5	5	11	101
Ж	6	6	12	110
З	7	7	13	111

Наряду с равномерными кодами могут применяться и неравномерные коды, когда каждая буква из алфавита источника кодируется различным числом символов, к примеру, А - 10, Б – 110, В – 1110 и т.д.

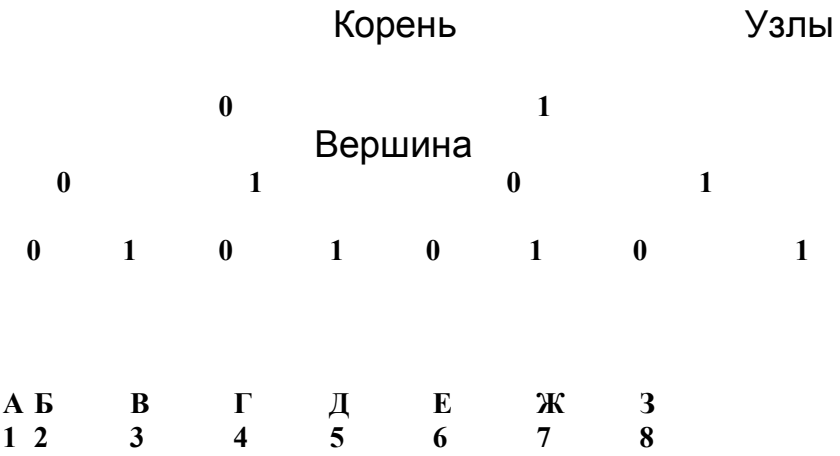


Рис. 7.1. Графическое представление кодового дерева

Кодовое дерево для неравномерного кодирования может выглядеть, например, так, как показано на рис. 7.2.

При использовании этого кода буква А будет кодироваться, как 1, Б - как 0, В – как 11 и т.д. Однако можно заметить, что, закодировав, к примеру, текст АББА = 1001 , мы не сможем его однозначно декодировать, поскольку такой же код дают фразы: ЖА = 1001, АЕА = 1001 и ГД = 1001. Такие коды, не обеспечивающие однозначного декодирования, называются приводимыми, или непрефиксными, кодами и не могут на практике применяться без специальных разделяющих символов. Примером применения такого типа кодов может служить азбука Морзе, в которой кроме точек и тире есть специальные символы разделения букв и слов. Но это уже не двоичный код.

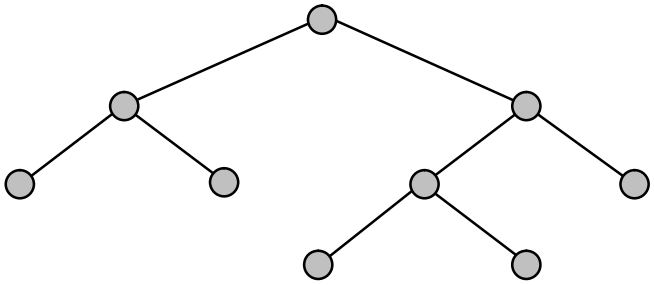


Рис. 7.2. Кодовое дерево для неравномерного кодирования

Однако можно построить неравномерные неприводимые коды, допускающие однозначное декодирование. Для этого необходимо, чтобы всем буквам алфавита соответствовали лишь вершины кодового дерева, например, такого, как показано на рис. 7.3. Здесь ни одна кодовая комбинация не является началом другой, более длинной, поэтому неоднозначности декодирования не будет. Такие неравномерные коды называются префиксными.

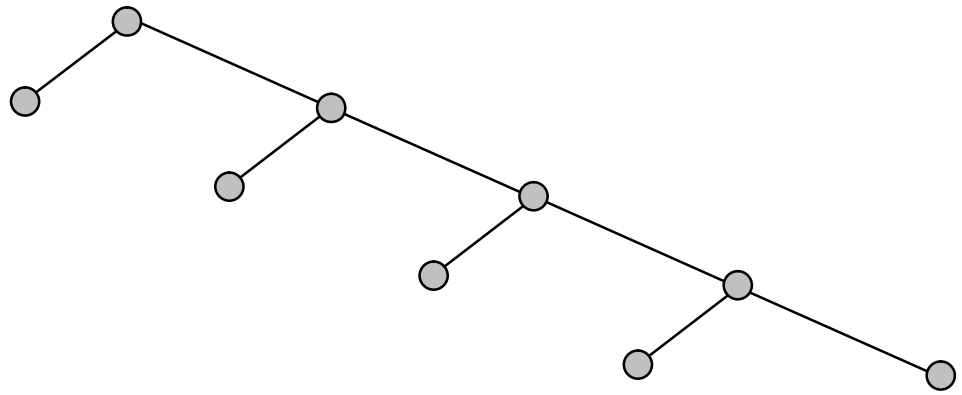


Рис. 7.3. Кодовое дерево для префиксного кода

Прием и декодирование неравномерных кодов - процедура гораздо более сложная, нежели для равномерных. При этом усложняется аппаратура декодирования и синхронизации, поскольку поступление элементов сообщения (букв) становится нерегулярным. Так, к примеру, приняв первый 0, декодер должен посмотреть в кодовую таблицу и выяснить, какой букве соответствует принятая последовательность. Поскольку такой буквы нет, он должен ждать прихода следующего символа. Если следующим символом будет 1, тогда декодирование первой буквы завершится – это будет Б, если же вторым принятым символом снова будет 0, придется ждать третьего символа и т.д.

Зачем же используются неравномерные коды, если они столь неудобны?

Рассмотрим пример кодирования сообщений x_i из алфавита объемом $K = 8$ с помощью произвольного n-разрядного двоичного кода.

Пусть источник сообщения выдает некоторый текст с алфавитом от **А** до **З** и одинаковой вероятностью букв $P(x_i) = 1/8$.

Кодирующее устройство кодирует эти буквы равномерным трехразрядным кодом (см. табл. 7.1).

Определим основные информационные характеристики источника с таким алфавитом:

- энтропия источника $H(X) = -\sum_{i=1}^{K} P_i \log P_i$, $H(X) = \log K = 3 \frac{бит}{символ}$;
- избыточность источника $R_u = 1 - \frac{H(X)}{\log K} = 0$;
- среднее число символов в коде $L = \sum_{i=1}^{K} \mu_i \cdot P_i = \sum_{i=1}^{8} 3 \cdot \frac{1}{8} = 3$;
- избыточность кода $R_k = 1 - \frac{H(X)}{L} = 0$.

Таким образом, при кодировании сообщений с равновероятными буквами избыточность выбранного (равномерного) кода оказалась равной нулю.

Пусть теперь вероятности появления в тексте различных букв будут разными (табл. 7.2).

Таблица 7.2

Вероятности появления букв

А	Б	В	Г	Д	Е	Ж	З
Ра =0.6	***Рб*** =0.2	***Рв*** =0.1	***Рг***= 0.04	***Рд***=0 .025	***Ре***= 0.015	***Рж*** =0.01	***Рз*** =0.01

Энтропия источника $H(X)=-\sum_{i=1}^{K} P_i \log P_i$ в этом случае, естественно, будет меньшей и составит $H(X) = 1.781 \ \frac{бит}{символ}$.

Среднее число символов на одно сообщение при использовании того же равномерного трехразрядного кода

$$L = \sum_{i=1}^{K} \mu_i P_i = \mu \sum_{i=1}^{K} P_i = \mu = 3.$$

Избыточность кода в этом случае будет

$$R_k = 1 - \frac{H(X)}{L} = 1 - \frac{1.781}{3} \approx 0.41,$$

или довольно значительной величиной (в среднем 4 символа из 10 не несут никакой информации).

В связи с тем, что при кодировании неравновероятных сообщений равномерные коды обладают большой избыточностью, было предложено использовать неравномерные коды, длительность кодовых комбинаций которых была бы согласована с вероятностью выпадения различных букв.

Такое кодирование называется статистическим.

Неравномерный код при статистическом кодировании выбирают так, чтобы более вероятные буквы передавались с помощью более коротких комбинаций кода, менее вероятные - с помощью более длинных. В результате уменьшается средняя длина кодовой группы в сравнении со случаем равномерного кодирования.

Операция кодирования тем более эффективна (экономична), чем меньшей длины кодовые слова сопоставляются сообщениям. Поэтому за характеристику эффективности кода примем среднюю длину кодового слова:

$$L = \sum_{i=1}^{n} \mu_i P(x_i), \qquad (7.1)$$

где μ_i – длина кодового слова, сопоставляемая x_i сообщению.

При установлении оптимальных границ для L исходят из следующих соображений. Во-первых, количество информации, несомое кодовым словом, не должно быть меньше количества информации, содержащегося в соответствующем сообщении, иначе при кодировании будут происходить потери в передаваемой информации. Во-вторых, кодирование будет тем более эффективным, чем большее количество информации будет содержать в себе каждый кодовый символ. Это количество информации максимально, когда все кодовые символы равновероятны, и равно $\log m$. При этом i-е кодовое слово будет нести количество информации, равное $\mu_i \log m$.

Шенноном сформулирована следующая теорема. При кодировании сообщений x_i в алфавите, насчитывающем m символов, при условии отсутствия шумов, средняя длина кодового слова не может быть меньше, чем

$$L \geq {}^{З\ (Ч)}\!\!\diagup\!\!_{\log m}, \qquad (7.2)$$

где $H(X)$ – энтропия сообщения.

Если вероятности сообщений не являются целочисленными степенями числа m, точное достижение указанной границы невозможно, но при кодировании достаточно длинными группами к этой границе можно сколь угодно приблизиться.

Данная теорема не дает явных рецептов для нахождения кодовых слов со средней длиной (7.2), а поэтому она является теоремой существования. Важность этой теоремы состоит в том, что она определяет предельно возможную эффективность кода, позволяет оценить, насколько тот или иной конкретный код близок к самому экономному, и, наконец, придает прямой физический смысл одному из основных понятий теории информации – энтропии множества сообщений как минимальному числу двоичных символов (m = 2), приходящихся в среднем на одно сообщение.

Приведем два известных способа построения кодов, которые позволяют приблизиться к равновероятности и независимости кодовых символов.

7.1.1. Код Шеннона-Фано.

Для построения этого кода все сообщения X_i выписываются в порядке убывания их вероятностей (табл. 7.3).

Таблица 7.3

Построение кода Шеннона-Фано

x_i	$P(x_i)$	Разбиение сообщений на подгруппы					Код	μ_i	L_{xi}
x_1	0,35	1	1				1 1	2	0,70
x_2	0,15	1	0				1 0	2	0,30
x_3	0,13	0	1	1			0 1 1	3	0,39

x_4	0,09	0	1	0			0	1	0			3	0,27
x_5	0,09	0	0	1	1		0	0	1	1		4	0,36
x_6	0,08	0	0	1	0	1	0	0	1	0		4	0,32
x_7	0,05	0	0	0	0	1	0	0	0	1		4	0,20
x_8	0,04	0	0	0	0	1	0	0	0	0	1	5	0,20
x_9	0,02	0	0	0	0	0	0	0	0	0	0	5	0,10

Записанные таким образом сообщения затем разбиваются на две по возможности равновероятностные подгруппы. Всем сообщениям первой подгруппы присваивают цифру 1 в качестве первого кодового символа, а сообщениям второй подгруппы – цифру 0. Аналогичное деление на подгруппы продолжается до тех пор, пока в каждую подгруппу не попадает по одному сообщению.

Найденный код весьма близок к оптимальному. В самом деле, энтропия сообщений:

$$H(X) = -\sum_{i=1}^{9} P(x_i)\log P(x_i) = -(0{,}35\log 0{,}35 + 0{,}15\log 0{,}15 + 0{,}13\log 0{,}13 + $$
$$+ 0{,}09\log 0{,}09 + 0{,}09\log 0{,}09 + 0{,}08\log 0{,}08 + 0{,}05\log 0{,}05 +$$
$$+ 0{,}04\log 0{,}04 + 0{,}02\log 0{,}02) \cong 2{,}75 \ \frac{\text{бит}}{\text{собщение}}. \qquad (7.3)$$

Средняя длина кодового слова:

$$L = \sum_{i=1}^{9} L_{x_i} = 0{,}70 + 0{,}30 + 0{,}39 + 0{,}27 + 0{,}36 + 0{,}32 + 0{,}20 + 0{,}20 + 0{,}10 = 2{,}84. \quad (7.4)$$

Среднее число нулей:

$$L(0) = 0{,}15 + 0{,}13 + 0{,}18 + 0{,}18 + 0{,}24 + 0{,}15 + 0{,}16 + 0{,}10 = 1{,}29. \qquad (7.5)$$

Среднее число единиц:

$$L(1) = 0{,}70 + 0{,}15 + 0{,}26 + 0{,}09 + 0{,}18 + 0{,}08 + 0{,}05 + 0{,}04 = 1{,}55. \quad (7.6)$$

Вероятность появления нулей:

$$P(0) = \frac{L(0)}{L} = \frac{1{,}29}{2{,}84} = 0{,}455. \qquad (7.7)$$

Вероятность появления единиц

$$P(1) = \frac{L(1)}{L} = \frac{1{,}55}{2{,}84} = 0{,}545. \qquad (7.8)$$

Таким образом, получили код близкий к оптимальному.

7.1.2. Код Хаффмана.

Для получения кода Хаффмана все сообщения выписывают в порядке убывания вероятностей. Две наименьшие вероятности объединяют скобкой (табл. 7.4) и одной из них присваивают символ 1, а другой – 0.

Затем эти вероятности складывают, результат записывают в промежутке

между ближайшими вероятностями. Процесс объединения двух сообщений с наименьшими вероятностями продолжают до тех пор, пока суммарная вероятность двух оставшихся сообщений не станет равной единице. Код для каждого сообщения строится при записи двоичного числа справа налево путем обхода по линиям вверх направо, начиная с вероятности сообщения, для которого строится код.

Средняя длина кодового слова (табл. 7.4) L = 2,82, что несколько меньше, чем в коде Шеннона-Фано (L = 2,84). Кроме того, методика Шеннона-Фано не всегда приводит к однозначному построению кода. Ведь при разбиении на подгруппы можно сделать большей по вероятности как верхнюю, так и нижнюю подгруппы. От этого недостатка свободна методика Хаффмана. Она гарантирует однозначное построение кода с наименьшим для данного распределения вероятностей средним числом символов на букву. Однако, как следует из приведенных выше цифр, некоторая избыточность в кодовых комбинациях осталась. Из теоремы Шеннона следует, что эту избыточность также можно устранить, если перейти к кодированию достаточно большими блоками.

Рассмотрим процедуру эффективного кодирования двух сообщений X_1 и X_2 с вероятностями $P(X_1)$ = 0,9 и $P(X_2)$ = 0,1 по методу Хаффмана: отдельных сообщений; сообщений, составленных по два в группе; сообщений, составленных по три в группе. Сравним коды по эффективности L , по скорости передачи R_t и по избыточности R, если длительности кодовых символов одинаковы и равны $\tau = 10^{-6}\ C$.

Таблица 7.4

Построение кода Хаффмана

X_i	$P(X_i)$	Объединение сообщений							Код
X_1	0,35	0,35	0,35	0,35	0,35	0,35	0,37	0,63	1
X_2	0,15	0,15	0,15	0,17	0,20	0,28	0,35	0,37	101
X_3	0,13	0,13	0,13	0,15	0,17	0,20	0,28		100
X_4	0,09	0,09	0,11	0,13	0,15	0,17			010
X_5	0,09	0,09	0,09	0,11	0,13				001
X_6	0,08	0,08	0,09	0,09					000

Энтропия источника в соответствии с (2.14)

$$H(X) = -0{,}9 \log 0{,}9 - 0{,}1 \log 0{,}1 = 0{,}469 \ \frac{\text{бит}}{\text{сообщение}}. \qquad (7.8)$$

При кодировании отдельных сообщений методом Хаффмана сообщению X_1 сопоставляется кодовый символ 1, а сообщению X_2 – 0. Средняя длина кодового символа при этом:

$$L = 0{,}9 \cdot 1 + 0{,}1 \cdot 1 = 1. \qquad (7.9)$$

Скорость передачи:

$$Rt = \frac{H(X)}{\bar{\tau}} = \frac{0{,}469}{10^{-6}} = 469000 \ \frac{\text{бит}}{\text{с}}, \qquad (7.10)$$

что составляет 46,9% от пропускной способности $C = \frac{1}{\tau}$. Избыточность кода равна избыточности источника сообщений:

$$R_k = R = \frac{H\max(X) - H(X)}{H\max(X)} = \frac{1 - 0{,}469}{1} = 0{,}531.$$

Для повышения эффективности кода перейдем к кодированию групп сообщений (табл. 7.5)

Средняя длина кодового слова, приходящего на одно сообщение:

$$L = 1/2 \cdot (1 \cdot 0{,}81 + 2 \cdot 0{,}09 + 3 \cdot 0{,}09 + 3 \cdot 0{,}01) = 0{,}645. \qquad (7.11)$$

Скорость передачи при этом

$$Rt = \frac{H(X)}{\bar{\tau}} = \frac{0{,}469}{0{,}645 \cdot 10^{-6}} = 727000 \frac{\text{бит}}{\text{с}}, \qquad (7.12)$$

что составляет 72,7% от максимально возможной скорости передачи (10^6 бит/с).

Чтобы найти избыточность кода, вычислим вероятность появления кодового символа 0 и 1:

$$P(0) = \frac{(0{,}09 \cdot 2 + 0{,}09 + 2 \cdot 0{,}01)}{(2 \cdot 0{,}645)} = 0{,}23.$$

$$P(1) = 1 - P(0) = 0{,}77.$$

Энтропия кода:

$$H_k = -0{,}23 \log 0.23 - 0{,}77 \log 0{,}77 = 0{,}778 \ \frac{\text{бит}}{\text{символ}}.$$

Избыточность:

$$R_k = 1 - H_k = 0{,}222.$$

Таблица 7.5

Кодирование сообщений, составленных по два в группе

X_iX_j	$P(X_j, X_i)$			Код	μ_i
X_1X_1	0,81	0,81	0,81 (1)	1	1
X_1X_2	0,09	0,10 (1)	0,19 (0)	00	2
X_2X_1	0,09 (1)	0,09 (0)		011	3
X_2X_2	0,01 (0)			010	3

А сейчас перейдем к кодированию групп сообщений, содержащих три сообщения в группе (табл. 7.6)

Таблица 7.6

Кодирование сообщений, составленных по три в группе

$X_jX_iX_k$	$P(X_iX_jX_k)$	Объединение сообщений						Код
$X_1X_1X_1$	0,729	0,729	0,729	0,729	0,729	0,729	0,729	1
$X_1X_1X_2$	0,081	0,081	0,081	0,081	0,109	0,162	0,271	011
$X_1X_2X_1$	0,081	0,081	0,081	0,081	0,081	0,109		010
$X_2X_1X_1$	0,081	0,081	0,081	0,081	0,081			001
$X_1X_2X_2$	0,009	0,010	0,018	0,028				00011
$X_2X_1X_2$	0,009	0,009	0,010					00010

$X_2X_2X_1$	0,009 — 0,009 —	00001
$X_2X_2X_2$	0,001 —	00000

Средняя длина кодового слова, приходящаяся на одно сообщение, в этом случае будет:

$$L = 1/3 \cdot (0{,}729 + 0{,}081 \cdot 9 + 0{,}009 \cdot 15 + 0{,}001 \cdot 5) = 0{,}533.$$

При этом скорость передачи:

$$R_t = \frac{0{,}469}{0{,}533 \cdot 10^{-6}} = 880000 \ \frac{\text{бит}}{\text{с}}.$$

что составляет 88% от пропускной способности.

Вероятность появления символов 0 и 1:

$$P(0) = \frac{(0{,}081 \cdot 5 + 0{,}009 \cdot 11 + 0{,}001 \cdot 5)}{0{,}533 \cdot 3} = 0{,}32;$$

$$P(1) = 1 - P(0) = 0{,}68.$$

Найдем энтропию и избыточность кода в этом случае:

$$H_k = -0{,}32\log 0{,}32 - 0{,}68\log 0{,}68 = 0{,}904;$$

$$R_k = 1 - H_k = 0{,}096.$$

Сведем полученные результаты в табл. 7.7.

Таблица 7.7

Результаты, полученные при кодировании

Вычисляемые величины	Число сообщений в группе			Предельные значения вычисляемой величины
	1	2	3	
L	1	0,645	0,533	$H(X)/\log 2 = 0{,}469$
$R_{\text{t, бит/с}}$	469000	727000	880000	$C = 1/\tau = 10^{-6}$
$P(0)$	0,9	0,23	0,32	$P(0) = 0{,}5$
$P(1)$	0,1	0,77	0,68	$P(1) = 0{,}5$
R_{k}, %	53,1	23.2	9,6	$R_{\text{k}} = 0$

Если кодировать группы по 4 и более сообщений, мы еще более приблизимся к предельным значениям вычисляемых величин.

Следует подчеркнуть, что увеличение эффективности кодирования при укрупнении блоков не связано с учетом все более далеких систематических

связей, так как нами рассматривались алфавиты с некоррелированными знаками. Повышение эффективности определяется лишь тем, что набор вероятностей, получающихся при укрупнении блоков**,** можно делить на более близкие по суммарным вероятностям подгруппы.

7.2. Префиксные коды

Рассмотрев методики построения эффективных кодов, нетрудно убедиться в том, что эффект достигается благодаря присвоению более коротких кодовых комбинаций более вероятным сообщениям и более длинных менее вероятным сообщениям. Таким образом, эффект связан с различием в числе символов кодовых комбинаций. А это приводит к трудностям при декодировании. Конечно, для различения кодовых комбинаций можно ставить специальный разделительный символ, но при этом значительно снижается эффект, которого мы добивались, так как средняя длина кодовой комбинации по существу увеличивается на символ.

Более целесообразно обеспечить однозначное декодирование без введения дополнительных символов. Для этого эффективный код необходимо строить так, чтобы ни одна комбинация кода не совпадала с началом более длинной комбинации. Коды, удовлетворяющие этому условию, называют префиксными кодами. Последовательность 100000110110110100 комбинаций префиксного кода, например,

X_1	X_2	X_3	X_4
00	01	101	100

декодируется однозначно:

100	00	01	101	101	101	00
X_4	X_1	X_2	X_3	X_3	X_3	X_1

Последовательность 000101010101 комбинаций непрефиксного кода, например,

X_1	X_2	X_3	X_4
00	01	101	010

(комбинация 01 является началом комбинации 010), может быть декодирована по-разному:

00	01	01	01	010	101
X_1	X_2	X_2	X_2	X_4	X_3

00	010	101	010	101
X_1	X_4	X_3	X_4	X_3

00	01	010	101	01	01
X_1	X_2	X_4	X_3	X_2	X_2

Нетрудно убедиться, что коды, получаемые в результате применения методики Шеннона-Фано или Хаффмана, являются префиксными.

7.3. Недостатки системы эффективного кодирования

Причиной одного из недостатков является различие в длине кодовых комбинаций. Если моменты снятия информации с источника неуправляемы, кодирующее устройство через равные промежутки времени выдает комбинации различной длины. Так как линия связи используется эффективно только в том случае, когда символы поступают в нее с постоянной скоростью, то на выходе кодирующего устройства должно быть предусмотрено буферное устройство. Оно запасает символы по мере поступления и выдает их в линию связи с постоянной скоростью. Аналогичное устройство необходимо и на приемной стороне.

Второй недостаток связан с возникновением задержки в передаче информации. Наибольший эффект достигается при кодировании длинными блоками, а это приводит к необходимости накапливать знаки, прежде чем поставить им в соответствие определенную последовательность символов. При декодировании задержка возникает снова. Общее время задержки может быть велико, особенно при появлении блока, вероятность которого мала. Это следует учитывать при выборе длины кодируемого блока.

Еще один недостаток заключается в специфическом влиянии помех на достоверность приема. Одиночная ошибка может перевести передаваемую кодовую комбинацию в другую, не равную ей по длительности. Это повлечет за собой неправильное декодирование ряда последующих комбинаций, которые называют треком ошибки.

Специальными методами построения эффективного кода трек ошибки стараются свести к минимуму.

Следует отметить относительную сложность технической реализации систем эффективного кодирования.

Методы эффективного кодирования Шеннона-Фано и Хаффмана, рассмотренные выше, позволяют производить кодирование, если известна статистика входных сообщений, т.е. известна вероятность их появления $p(x_i)$.

7.4. Эффективное кодирование при неизвестной статистике сообщений

Коды, эффективные одновременно для некоторого класса источников, называют универсальными кодами. Сформулируем постановку задачи универсального кодирования источников. Предположим, что алфавит состоит из двух

X_1 и X_2, появляющихся независимо, с вероятностями p и $g = 1\text{-}p$. Однако величина p заранее неизвестна. Требуется построить код, для которого среднее число символов «0» и «1» на одну букву алфавита приближалось бы к $H(X)$ при любом p, $0 <= p <= 1$. Этот код строится так. Множество всех блоков длины n в алфавите X разбиваем на группы, которые имеют одинаковые вероятности при любом p. Таких групп будет n+1. В нулевой группе отсутствует буква X_2, она состоит из единственного блока $X_1X_1X_1\ldots X_1$, вероятность появления которого p^n. Первая группа состоит из блоков длиной n, содержащих одну букву X_2. Эта группа состоит из $C_n^1 = n$ блоков, вероятность каждого из которых равна $P^{n-1} \cdot g$. Группы с номером k состоят из всех блоков длиной n, содержащих k букв X_2. Эта группа содержит n блоков, вероятность каждого из которых $p^{n-k} \cdot g^k$.

Универсальный код для k-й группы состоит из двух частей: префикса и суффикса. Префикс содержит $\log(n+1)$ двоичных знаков. Префикс указывает, к какой группе сообщений принадлежит кодируемый блок, суффикс содержит $\log C_n^k$ двоичных символов и указывает номер блока в группе.

Построенный таким образом код будет однозначно дешифрируем. На приемном конце первоначально по $\log(n+1)$ элементам кода определяют, к какой группе принадлежит переданное сообщение, а затем по следующим $\log C_n^k$ элементам определяют, какое именно сообщение передавалось.

Код в табл. 7.8 построен описанным выше способом. Здесь выделены штриховой линией префиксы.

Из приведенного выше описания метода кодирования видно, что наиболее трудоемкой частью кодирования является нахождение суффикса. Опишем алгоритм нахождения суффикса. Пусть в блоке X длиной n буква X_1 встречается на местах $i_1, i_2, \ldots, i_r$, тогда суффиксом для X назовем число $N(x)$, вычисляемое по правилу

$$N(x) = C_{i1-1}^1 + C_{i2-1}^2 + \ldots + C_{ir-1}^r, \qquad (7.13)$$

очевидно, что блоки с разными наборами (i_1, i_2,…,i_r) получают разные номера. При этом максимально значение номера $C_n^r - 1$. Таким образом, двоичная запись номера (суффикса) должна иметь длину $\log C_n^r$.

Для нахождения $N(x)$ воспользуемся таблицей биноминальных коэффициентов (треугольником Паскаля):

8 7 21 35 35 21 7 1 0
7 6 15 20 15 6 1 0
6 5 10 10 5 1 0
5 4 6 4 1 0
4 3 3 1 0
3 2 1 0

2 1 0
1 0

Элементы этой таблицы вычисляются по мере надобности либо размещаются в памяти кодирующего устройства.

Приведем фрагмент этой таблицы, в которой на пересечении *i*-й строки и *j*-го столбца стоит C_{i-1}.

Пример 7.1. Пусть $n = 8$, $X = X_2 X_1 X_1 X_2 X_1 X_1 X_2 X_1$, тогда $r = 5$; $i_1 = 2$; $i_2 = 3$; $i_3 = 5$; $i_4 = 6$; $i_5 = 8$. Номер блока $N(x) = C_1^1 + C_2^2 + C_4^3 + C_5^4 + C_7^5$. Слагаемые в $N(x)$ находим, используя таблицу дополнительных коэффициентов. Таким образом, $N(x) = 1 + 1 + 4 + 5 + 21 = 32$ или в двоичной записи $N(x) = 100000$. Декодирование производится с помощью этой же таблицы.

Пример 7.2. Пусть нам известно, что длина передаваемого блока равна 8, и что в блоке пять букв X_i (количество букв в блоке находим по префиксу). Находим максимальное число в пятом столбце, не превосходящее 32, это $21 = C_{8-1}^5$, следовательно, $i_5 = 8$, находим разность $32 - 21 = 11$. Находим далее максимальное число четвертого столбца, не превосходящее 11. Это $5 = C_{6-1}^4$, т.е. $i_4 = 6$. Аналогично находим $i_3 = 5$, $i_2 = 3$, $i_1 = 2$. Следовательно, декодированное сообщение имеет вид $X = X_2 X_1 X_1 X_2 X_1 X_1 X_2 X_1$, т.е. совпадает с переданным.

Рассмотренные кодирование и декодирование достаточно просто осуществляются с помощью специализированных вычислительных устройств.

Таблица 7.8

Построение префиксного кода

Кодируемые слова	**Номер группы**	**Вероятность слова**	**Код**
$X_1X_1X_1X_1$	1	P^4	000
$X_1X_1X_1X_2$	2	P^3g	001 00
$X_1X_1X_2X_1$			001 01
$X_1X_2X_1X_1$			001 10
$X_2X_1X_1X_1$			001 11
Кодируемые слова	**Номер группы**	**Вероятность слова**	**Код**
$X_1X_2X_1X_2$	3	P^2g^2	010 001
$X_2X_1X_1X_2$			010 010

$X_1X_2X_2X_1$ $X_2X_1X_2X_1$ $X_2X_2X_1X_1$			010 011 010 100 010 101
$X_2X_2X_2X_1$ $X_2X_2X_1X_2$ $X_2X_1X_2X_2$ $X_1X_2X_2X_2$	4	Pg^2	011 00 011 01 011 10 011 11
$X_2X_2X_2X_2$	5	g^4	100

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие кодовые слова называются неперекрываемыми?
2. Запишите выражение для средней длины кодового слова?
3. Сформулируйте теорему существования.
4. Поясните принцип кодирования сообщений в коде Шеннона-Фано.
5. Поясните принцип кодирования сообщений в коде Хаффмана.
6. Сравните код Шеннона-Фано и код Хаффмана.
7. В чем преимущество кодирования групп сообщений?
8. Какие коды называются префиксными?
9. Перечислите недостатки систем эффективного кодирования.
10. Поясните принцип эффективного кодирования при неизвестной статистике сообщений.

8. СЖАТИЕ СООБЩЕНИЙ

8.1. Типы систем сжатия

Передача и хранение информации требуют достаточно больших затрат. И чем с большим количеством информации нам приходится иметь дело, тем дороже это стоит. К сожалению, большая часть данных, которые нужно передавать по каналам связи и сохранять, имеет не самое компактное представление. Скорее, эти данные хранятся в форме, обеспечивающей их наиболее простое использование, например: обычные книжные тексты, ASCII коды текстовых редакторов, двоичные коды данных ЭВМ, отдельные отсчеты сигналов в си-

стемах сбора данных и т.д. Однако такое наиболее простое в использовании представление данных требует вдвое - втрое, а иногда и в сотни раз больше места для их сохранения и полосу частот для их передачи, чем на самом деле нужно. Поэтому сжатие данных – это одно из наиболее актуальных направлений современной телемеханики. Таким образом, цель сжатия данных - обеспечить компактное представление данных, вырабатываемых источником, для их более экономного сохранения и передачи по каналам связи.

Ниже приведена условная структура системы сжатия данных:

Данные источника→Кодер→Сжатые данные→Декодер→Восстановленные данные

В этой схеме вырабатываемые источником данные определим как *данные источника*, а их компактное представление - как *сжатые данные*. Система сжатия данных состоит из *кодера* и *декодера источника.* Кодер преобразует данные источника в сжатые данные, а декодер предназначен для восстановления данных источника из сжатых данных. *Восстановленные* данные, вырабатываемые декодером, могут либо абсолютно точно совпадать с исходными *данными источника*, либо незначительно отличаться от них.

Существуют два типа систем сжатия данных:

- системы сжатия без потерь информации (неразрушающее сжатие);
- системы сжатия с потерями информации (разрушающее сжатие)

В системах сжатия без потерь декодер восстанавливает данные источника абсолютно точно, таким образом, структура системы сжатия выглядит следующим образом*:*

Вектор данных $\mathbf{X} \rightarrow$ Кодер $\rightarrow$ $\mathbf{B(X)} \rightarrow$ Декодер $\rightarrow$ $\mathbf{X}$

Вектор данных источника $\boldsymbol{X}$, подлежащих сжатию, представляет собой последовательность $\boldsymbol{X} = (\boldsymbol{x_1, x_2,\ldots x_n})$ конечной длины. Отсчеты $\boldsymbol{x_i}$ - составляющие вектора $\boldsymbol{X}$ **-** выбраны из конечного *алфавита данных* $\boldsymbol{A}$**.** При этом размер вектора данных $\boldsymbol{n}$ ограничен, но он может быть сколь угодно большим. Таким образом, источник на своем выходе формирует в качестве данных $\boldsymbol{X}$ последовательность длиной $\boldsymbol{n}$ из алфавита $\boldsymbol{A}$ **.**

Выход кодера - сжатые данные, соответствующие входному вектору $\boldsymbol{X}$**,** - представим в виде двоичной последовательности $\boldsymbol{B(X) = (\ b_1,b_2,\ldots b_k\ ),}$ размер которой $\boldsymbol{k}$ зависит от $\boldsymbol{X}$. Назовем $\boldsymbol{B(X)}$ кодовым словом, присвоенным вектору $\boldsymbol{X}$ кодером (или кодовым словом, в которое вектор $\boldsymbol{X}$ преобразован кодером). Поскольку система сжатия - неразрушающая, одинаковым векторам $\boldsymbol{X_l = X_m}$ должны соответствовать одинаковые кодовые слова $\boldsymbol{B(X_l) = = B(X_m).}$

При решении задачи сжатия естественным является вопрос, насколько эффективна та или иная система сжатия. Поскольку, как мы уже отмечали, в основном используется только двоичное кодирование, то такой мерой может служить *коэффициент сжатия* $\boldsymbol{r}$*,* определяемый как отношение

$$\boldsymbol{r} = \frac{\textbf{размер данных источника в битах}}{\textbf{.}} \qquad (8.1)$$

размер сжатых данных в битах

Таким образом, коэффициент сжатия ***r = 2*** означает, что объем сжатых данных составляет половину от объема данных источника. Чем больше коэффициент сжатия ***r,*** тем лучше работает система сжатия данных.

Наряду с коэффициентом сжатия ***r*** эффективность системы сжатия может быть охарактеризована *скоростью сжатия* ***R,*** определяемой как отношение

$$R = k/n \qquad (8.2)$$

и измеряемой в "количестве кодовых бит, приходящихся на отсчет данных источника". Система, имеющая *больший* коэффициент сжатия, обеспечивает *меньшую* скорость сжатия.

В системе сжатия с потерями (разрушением) кодирование производится таким образом, что декодер не в состоянии восстановить данные источника в первоначальном виде. Структурная схема системы сжатия с разрушением выглядит следующим образом*:*

$$\mathbf{X} \rightarrow \text{Квантователь} \rightarrow \mathbf{X^q} \rightarrow \text{Неразрушающий кодер} \rightarrow \mathbf{B\,(X^q)} \rightarrow \text{Декодер} \rightarrow \mathbf{X^*}$$

Как и в предыдущей схеме, $X = (x_1, x_2, \ldots x_n)$ - вектор данных, подлежащих сжатию. Восстановленный вектор обозначим как $X^* = (x_1, x_2, \ldots x_n)$. Отметим наличие в этой схеме сжатия элемента, который отсутствовал при неразрушающем сжатии, - *квантователя.*

Квантователь применительно к вектору входных данных X формирует вектор X^q, достаточно близкий к X в смысле среднеквадратического расстояния. Работа квантователя основана на понижении размера алфавита (простейший квантователь производит округление данных до ближайшего целого числа).

Далее кодер подвергает неразрушающему сжатию вектор квантованных данных X^q таким образом, что обеспечивается однозначное соответствие между X^q и $B(X^q)$ (для $X_l^q = X_m^q$ выполняется условие $B(X_l^q) = B(X_m^q)$). Однако система в целом остается разрушающей, поскольку двум различным векторам X может соответствовать один и тот же вектор X^*.

Разрушающий кодер характеризуется двумя параметрами - скоростью сжатия R и величиной искажений D, определяемых как

$$R = k / n,$$

$$D = (1/n)\sum (x_i - x_i^*)^2 . \qquad (8.3)$$

Параметр R характеризует скорость сжатия в битах на один отсчет источника, величина D является мерой среднеквадратического различия между X^* и X.

Если имеются система разрушающего сжатия со скоростью и искажениями R_1 и D_1 соответственно и вторая система со скоростью R_2 и искажениями D_2, то первая из них лучше, если $R_1 < R_2$ и $D_1 < D_2$. Однако, к сожалению, невозможно построить систему разрушающего сжатия, обеспечивающую одновременно

снижение скорости ***R*** и уменьшение искажений ***D,*** поскольку эти два параметра связаны обратной зависимостью. Поэтому целью оптимизации системы сжатия с потерями может быть либо минимизация скорости при заданной величине искажений, либо получение наименьших искажений при заданной скорости сжатия.

Выбор системы неразрушающего или разрушающего сжатия зависит от типа данных, подлежащих сжатию. При сжатии текстовых данных, компьютерных программ, документов, чертежей и т.п. совершенно очевидно, что нужно применять неразрушающие методы, поскольку необходимо абсолютно точное восстановление исходной информации после ее сжатия. При сжатии речи, музыкальных данных и изображений, наоборот, чаще используется разрушающее сжатие, поскольку при практически незаметных искажениях оно обеспечивает на порядок, а иногда и на два меньшую скорость ***R***. В общем случае разрушающее сжатие обеспечивает, как правило, существенно более высокие коэффициенты сжатия, нежели неразрушающее.

Ниже приведены ряд примеров, иллюстрирующих необходимость процедуры сжатия.

Пример 8.1. Предположим, что источник генерирует цифровое изображение (кадр) размером 512*512 элементов, содержащее 256 цветов. Каждый цвет представляет собой число из множества {0,1,2… 255}. Математически это изображение представляет собой матрицу 512x512, каждый элемент которой принадлежит множеству {0,1,2… 255}. (Элементы изображения называют пикселами).

В свою очередь, каждый пиксел из множества {0,1,2… 255} может быть представлен в двоичной форме с использованием 8 бит. Таким образом, размер данных источника в битах составит 8x512x512= 2^{21}, или 2,1 Мегабита.

На жесткий диск объемом в 1 Гигабайт поместится примерно 5000 кадров изображения, если они не подвергаются сжатию (видеоролик длительностью примерно в пять минут). Если же это изображение подвергнуть сжатию с коэффициентом r = 10, то на этом же диске мы сможем сохранить уже почти часовой видеофильм!

Предположим далее, что мы хотим передать исходное изображение по телефонной линии, пропускная способность которой составляет 14000 бит/с. На это придется затратить 21000000 бит/14000 бит/с, или примерно 3 минуты. При сжатии же данных с коэффициентом r = 40 на это уйдет всего 5 секунд!

Пример 8.2. В качестве данных источника, подлежащих сжатию, выберем фрагмент изображения размером 4x4 элемента и содержащее 4 цвета: R = ="красный", O = "оранжевый", Y = "синий", G = "зеленый":

R	R	O	Y
R	O	O	Y
O	O	Y	G
Y	Y	Y	G

Просканируем это изображение по строкам и каждому из цветов присвоим соответствующую интенсивность, например, R = 3, O = 2, Y = 1 и G = 0, в результате чего получим вектор данных X = (3,3,2,1,3,2,2,1,2,2,1,0,1,1,1,0).

Для сжатия данных возьмем кодер, использующий следующую таблицу перекодирования данных источника в кодовые слова (вопрос о выборе таблицы оставим на будущее):

Кодер	
Отсчет	Кодовое слово
3	001
2	01
1	1
0	000

Используя таблицу кодирования, заменим каждый элемент вектора ***X*** соответствующей кодовой последовательностью из таблицы (так называемое *кодирование без памяти)*. Сжатые данные (кодовое слово ***B(X))*** будут выглядеть следующим образом:

B(X) = (0,0,1,0,0,1,0,1,1,0,0,1,0,1,0,1,1,0,1,0,1,1,0,0,0,1,1,1,0,0,0).

Коэффициент сжатия при этом составит ***r*** = 32/31, или 1,03. Соответственно скорость сжатия ***R*** = 31/16 бит на отсчет.

8.2. Основные алгоритмы сжатия без потерь информации

Сжатие осуществляется либо на прикладном уровне с помощью программы сжатия, либо с помощью устройств защиты от ошибок непосредственно в составе модемов.

Основными методами сжатия являются: вероятностные, статические, арифметические, словарей и кодирование повторов.

К методам сжатия также относятся методы разностного кодирования, поскольку разности амплитуд представляется меньшим числом разрядов. Разностное кодирование реализовано в методах дельта-модуляции и её разновидностях.

Кодирование повторов (Run Length Encoding, RLE) применяется в основном для сжатия растровых изображений (графических файлов). Один из вариантов метода RLE предусматривает замену последовательности повторяющихся символов на строку, содержащую этот символ, и число, соответствующее количеству его повторений. Применение метода кодирования повторов для сжатия текстовых файлов оказывается неэффективным. Поэтому в современ-

ных системах передачи кодированной цифробуквенной информации алгоритм RLE используется мало.

Вероятностные методы сжатия используют кодовые слова переменной длинны. В основе вероятностных методов сжатия (алгоритмов Шеннона-Фано и Хаффмена) лежит идея построения «дерева», на «ветвях» которого положение символа определяется частостью его появления. Каждому символу присваивается код, длина которого обратно пропорциональна частости появления этого символа. Существуют две разновидности вероятностных методов, различающихся способом определения вероятности появления каждого символа:

– статические методы, использующие фиксированную таблицу частости появления символов, рассчитываемую перед началом процесса сжатия,

– динамические или адаптивные методы, в которых частость появления символов все время меняется и по мере считывания нового блока данных происходит перерасчет начальных значений частостей.

Статические методы имеют значительное быстродействие и не требуют большой оперативной памяти. Они нашли широкое применение в многочисленных программах–архиваторах, например ARC, PKZIP и др., но для сжатия передаваемых модемами данных используются редко – предпочтение отдается арифметическому кодированию и методу словарей, обеспечивающим большую степень сжатия.

Арифметические методы. При арифметическом кодировании строка символов заменяется действительным числом больше нуля и меньше единицы Арифметическое кодирование позволяет обеспечить высокую степень сжатия, особенно в случаях, когда сжимаются данные, где частость появления различных символов сильно варьируется. Однако сама процедура арифметического кодирования требует мощных вычислительных ресурсов, так как активно использует нецелочисленную арифметику, и до недавнего времени этот метод мало применялся при сжатии передаваемых данных. Лишь появление мощных процессоров, особенно с RISC–архитектурой, позволило создать эффективные устройства арифметического сжатия данных.

Метод словарей. Алгоритм для метода словарей описан в работах Зива и Лемпеля, которые впервые опубликовали его в 1977 г. В последующем алгоритм был назван Lempel-Ziv, или сокращенно LZ. На сегодня LZ–алгоритм и его модификации получили наиболее широкое распространение по сравнению с другими методами сжатия. В его основе лежит идея замены наиболее часто встречающихся последовательностей символов (строк) в передаваемом потоке ссылками на «образцы», хранящиеся в специально создаваемой таблице (словаре).

8.2.1 Вероятностные методы сжатия

Согласно методу Шеннона-Фано для каждого символа формируется битовый код, причем символы с различными частостями появления имеют коды различной длины [16]. Чем меньше частость появления символов в файле, тем больше размер его битового кода. Соответственно, чаще появляющийся символ имеет меньший размер кода.

Код строится следующим образом. Все символы, встречающиеся в файле, выписывают в таблицу в порядке убывания частости их появления. Затем их разделяют на две группы так, чтобы в каждой из них были примерно равные суммы частостей символов. Первые биты кодов всех символов одной половины устанавливаются в 0, а второй – в 1. После этого каждую группу делят еще раз пополам и так до тех пор, пока в каждой группе не останется по одному символу. Допустим, файл состоит из некоторой символьной строки *aaaaaaaaaabbbbbbbbbccccccddddeeeefff*, тогда каждый символ этой строки можно закодировать так, как показано в табл. 8.1.

Таблица 8.1

Пример построения кода Шеннона – Фано

Символ	Частость появления	Код
a	10	11
b	8	10
c	6	011
d	5	010
e	4	001
f	3	000

Можно видеть, что если раньше каждый символ кодировался 8 битами, то теперь требуется максимум три бита.

Однако способ Шеннона-Фано не всегда приводит к построению однозначного кода. Более удачен в данном отношении метод Хаффмена, позволяющий однозначно построить код с наименьшей средней длиной, приходящейся на символ.

Однако способ Шеннона-Фано не всегда приводит к построению однозначного кода. Более удачен в данном отношении метод Хаффмена, позволяющий однозначно построить код с наименьшей средней длиной, приходящейся на символ.

Суть метода Хаффмена сводится к следующему. Символы, встречающиеся в файле, выписываются в столбец в порядке убывания вероятностей (частости) их появления. Два последних символа объединяются в один с суммарной вероятностью. Из полученной новой вероятности м вероятностей новых символов, не использованных в объединении, формируется новый столбец в порядке убывания вероятностей, а две последние вновь объединяются. Это продолжается до тех пор, пока не останется одна вероятность, равная сумме вероятностей всех символов, встречающихся в файле.

Процесс кодирования по методу Хаффмена поясняется табл. 8.2. Для составления кода, соответствующего данному символу, необходимо проследить путь перехода знака по строкам и столбцам таблицы кода.

Таблица 8.2

Процесс кодирования по методу Хаффмена

Символ	Частость появления														Кодовое слово
c	22	22	22		26		32		42		58				01
e	20	20	20		22		26		32		42				00
h	16	16	16		20		22		26						111
i	16	16	16		16		20								110
a	10	10	16		16										100
k	10	10	10												1011
m	4	6													10101
b	2														10100

Недостатки метода Хаффмена. Самой большой сложностью с кодами Хаффмена, как следует из предыдущего обсуждения, является необходимость иметь таблицы вероятностей для каждого типа сжимаемых данных. Это не представляет проблемы, если известно, что сжимается английский или русский текст; мы просто предоставляем кодеру и декодеру подходящее для английского или русского текста кодовое дерево. В общем же случае, когда вероятность символов для входных данных неизвестна, статические коды Хаффмена работают неэффективно.

Решением этой проблемы является статистический анализ кодируемых данных, выполняемый в ходе первого прохода по данным, и составление на его основе кодового дерева. Собственно кодирование при этом выполняется вторым проходом.

Существует, правда, динамическая версия сжатия Хаффмена, которая может строить дерево Хаффмена "на лету" во время чтения и активного сжатия. Дерево постоянно обновляется, чтобы отражать изменения вероятностей входных данных. Однако и она на практике обладает серьезными ограничениями и недостатками и, кроме того, обеспечивает меньшую эффективность сжатия.

Еще один недостаток кодов Хаффмена - это то, что минимальная длина кодового слова для них не может быть меньше единицы, тогда как энтропия сообщения вполне может составлять и 0,1, и 0,01 бит/букву. В этом случае код Хаффмена становится существенно избыточным. Проблема решается применением алгоритма к блокам символов, но тогда усложняется процедура кодирования/декодирования и значительно расширяется кодовое дерево, которое нужно в конечном итоге сохранять вместе с кодом.

Наконец, код Хаффмена обеспечивает среднюю длину кода, совпадающую с энтропией, только в том случае, когда вероятности символов источника являются целыми отрицательными степенями двойки: 1/2 = 0,5; 1/4 = 0,25; 1/8 = 0,125; 1/16 = 0,0625 и т.д. На практике же такая ситуация встречается очень редко или может быть создана блокированием символов со всеми вытекающими отсюда последствиями.

8.2.2. Арифметическое кодирование

При арифметическом кодировании, в отличие от рассмотренных нами методов, когда кодируемый символ (или группа символов) заменяется соответствующим им кодом, *результат кодирования всего сообщения представляется одним или парой вещественных чисел в интервале от 0 до 1*. По мере кодирования исходного текста отображающий его интервал уменьшается, а количество десятичных (или двоичных) разрядов, служащих для его представления, возрастает. Очередные символы входного текста сокращают величину интервала исходя из значений их вероятностей, определяемых моделью. Более вероятные символы делают это в меньшей степени, чем менее вероятные, и, следовательно, добавляют меньше разрядов к результату. Поясним идею арифметического кодирования на простейшем примере. Пусть нам нужно закодировать следующую текстовую строку: ***РАДИОВИЗИР.***

Перед началом работы кодера соответствующий кодируемому тексту исходный интервал составляет ***[0; 1).***

Алфавит кодируемого сообщения содержит следующие символы (буквы): { ***Р, А, Д, И, О, В, З*** }.

Определим количество (встречаемость, вероятность) каждого из символов алфавита в сообщении и назначим каждому из них интервал, пропорциональный его вероятности. С учетом того, что в кодируемом слове всего 10 букв, получим табл. 8.3.

Таблица 8.3.

Вероятности появления символов

Символ	Вероятность	Интервал
А	0.1	0 – 0.1
Д	0.1	0.1 – 0.2
В	0.1	0.2 – 0.3
И	0.3	0.3 – 0.6
З	0.1	0.6 – 0.7
О	0.1	0.7 – 0.8
Р	0.2	0.8 – 1

Располагать символы в таблице можно в любом порядке: по мере их появления в тексте, в алфавитном или по возрастанию вероятностей – это совершенно не принципиально. Результат кодирования при этом будет разным, но эффект – одинаковым.

Итак, перед началом кодирования исходный интервал составляет [0 – 1).

После просмотра первого символа сообщения ***Р*** кодер сужает исходный интервал до нового - [0.8; 1), который модель выделяет этому символу. Таким образом, после кодирования первой буквы результат кодирования будет находиться в интервале чисел [0.8 - 1).

Следующим символом сообщения, поступающим в кодер, будет буква ***А***. Если бы эта буква была первой в кодируемом сообщении, ей был бы отведен интервал [0 - 0.1), но она следует за ***Р*** и поэтому кодируется новым *подынтервалом внутри уже выделенного для первой буквы*, сужая его до величины [0.80 - 0.82). Другими словами, интервал [0 - 0.1), выделенный для буквы ***А***, располагается теперь внутри интервала, занимаемого предыдущим символом (начало и конец нового интервала определяются путем прибавления к началу предыдущего интервала произведения ширины предыдущего интервала на значения интервала, отведенные текущему символу). В результате получим новый рабочий интервал [0.80 - 0.82), т.к. предыдущий интервал имел ширину в 0.2 единицы и одна десятая от него есть 0.02.

Следующему символу ***Д*** соответствует выделенный интервал [0.1 - 0.2), что применительно к уже имеющемуся рабочему интервалу [0.80 - 0.82) сужает его до величины [0.802 - 0.804).

Следующим символом, поступающим на вход кодера, будет буква ***И*** с выделенным для нее фиксированным интервалом [0,3 – 0,6). Применительно к уже имеющемуся рабочему интервалу получим [0,8026 - 0,8032).

Продолжая в том же духе, имеем:

вначале		[0.0 - 1.0)
после просмотра	***Р***	[0.8 - 1.0)
	А	[0.80 - 0.82)
	Д	[0.802 - 0.804)
	И	[0.8026 - 0.8032)
	О	[0.80302 - 0.80308)
	В	[0.803032 - 0.803038)
	И	[0.8030338 - 0.8030356)
	З	[0.80303488 - 0.80303506)
	И	[0.803034934 - 0.803034988)
	Р	[0.8030349772 - 0.8030349880)

Результат кодирования: интервал [0,8030349772 – 0,8030349880]. На самом деле, для однозначного декодирования теперь достаточно знать только одну границу интервала – нижнюю или верхнюю, то есть результатом кодирования может служить начало конечного интервала - 0,8030349772. Если быть еще более точным, то любое число, заключенное *внутри* этого интервала, однозначно декодируется в исходное сообщение. К примеру, это можно проверить с числом 0,80303498, удовлетворяющим этим условиям. При этом последнее число имеет меньшее число десятичных разрядов, чем числа, соответствующие нижней и верхней границам интервала, и, следовательно может быть представлено меньшим числом двоичных разрядов.

Нетрудно убедиться в том, что, чем шире конечный интервал, тем меньшим числом десятичных (и, следовательно, двоичных) разрядов он может быть представлен. Ширина же интервала зависит от распределения вероятностей кодируемых символов – более вероятные символы сужают интервал в меньшей

степени и, следовательно, добавляют к результату кодирования меньше бит. Покажем это на простом примере.

Допустим, нам нужно закодировать следующую строку символов: ***A A A A A A A A A A #***, где вероятность буквы ***A*** составляет 0,9. Процедура кодирования этой строки и получаемый результат будут выглядеть в этом случае следующим образом:

Входной символ	Нижняя граница	Верхняя граница
	0.0	1.0
A	0.0	0.9
A	0.0	0.81
A	0.0	0.729
A	0.0	0.6561
A	0.0	0.59049
A	0.0	0.531441
A	0.0	0.4782969
A	0.0	0.43046721
A	0.0	0.387420489
#	0.3486784401	0.387420489

Результатом кодирования теперь может быть, к примеру, число 0.35 , целиком попадающее внутрь конечного интервала 0.3486784401 – 0.387420489. Для двоичного представления этого числа нам понадобится 7 бит (два десятичных разряда соответствуют примерно семи двоичным), тогда как для двоичного представления результатов кодирования из предыдущего примера – 0,80303498 – нужно 27 бит !!!

При декодировании предположим, что все что декодер знает о тексте, – это конечный интервал [0,8030349772 - 0,8030349880]. Декодеру, как и кодеру, известна также таблица распределения выделенных алфавиту интервалов. Он сразу же понимает, что первый закодированный символ есть ***P***, так как результат кодирования целиком лежит в интервале [0.8 - 1), выделенном моделью символу ***P*** согласно таблице .

Теперь повторим действия кодера:

вначале [0.0 - 1.0);

после просмотра [0.8 - 1.0).

Исключим из результата кодирования влияние теперь уже известного первого символа ***P***, для этого вычтем из результата кодирования нижнюю границу диапазона, отведенного для ***P,*** – 0,8030349772 – 0.8 = =0,0030349772 – и разделим полученный результат на ширину интервала, отведенного для ***P,*** – 0.2. В результате получим 0,0030349772 / 0,2 = =0,015174886. Это число целиком вмещается в интервал, отведенный для буквы ***A,*** – [0 – 0,1) , следовательно, вторым символом декодированной последовательности будет ***A***.

Поскольку теперь мы знаем уже две декодированные буквы - ***PA***, исключим из итогового интервала влияние буквы ***A***. Для этого вычтем из остатка

0,015174886 нижнюю границу для буквы ***А*** 0,015174886 – 0.0 = =0,015174886 и разделим результат на ширину интервала, отведенного для буквы ***А***, то есть на 0,1. В результате получим 0,015174886/0,1=0,15174886. Результат лежит в диапазоне, отведенном для буквы ***Д***, следовательно, очередная буква будет ***Д***.

Исключим из результата кодирования влияние буквы ***Д***. Получим (0,15174886 – 0,1)/0,1 = 0,5174886. Результат попадает в интервал, отведенный для буквы ***И***, следовательно, очередной декодированный символ – ***И***, и так далее, пока не декодируем все символы:

Декодируемое число	Символ на выходе	Границы нижняя	Границы верхняя	Ширина интервала
0,8030349772	***Р***	0.8	1.0	0.2
0,015174886	***А***	0.0	0.1	0.1
0,15174886	***Д***	0.1	0.2	0.1
0,5174886	***И***	0.3	0.6	0.3
0,724962	***О***	0,7	0,8	0,1
0,24962	***В***	0,2	0,2	0,1
0,4962	***И***	0,3	0,6	0,3
0,654	***З***	0,6	0,7	0,1
0,54	***И***	0,3	0,6	0,3
0,8	***Р***	0,6	0,8	0,2
0.0	Конец декодирования			

Это основная идея арифметического кодирования, его практическая реализация несколько сложнее. Некоторые проблемы можно заметить непосредственно из приведенного примера.

Первая состоит в том, что декодеру нужно обязательно каким-либо образом дать знать об окончании процедуры декодирования, поскольку остаток 0,0 может означать букву а или последовательность аа, ааа, аааа и т.д. Решить эту проблему можно двумя способами.

Во-первых, кроме кода данных можно сохранять число, представляющее собой размер кодируемого массива. Процесс декодирования в этом случае будет прекращен, как только массив на выходе декодера станет такого размера.

Другой способ – включить в модель источника специальный символ конца блока, например *#,* и прекращать декодирование, когда этот символ появится на выходе декодера.

Вторая проблема вытекает из самой идеи арифметического кодирования и состоит в том, что окончательный результат кодирования – конечный интервал – станет известен только по окончании процесса кодирования. Следовательно, нельзя начать передачу закодированного сообщения, пока не получена последняя буква исходного сообщения и не определен окончательный интервал?

На самом деле в этой задержке нет необходимости. По мере того, как интервал, представляющий результат кодирования, сужается, старшие десятичные знаки его (или старшие биты, если число записывается в двоичной форме) перестают изменяться (посмотрите на приведенный пример кодирования).

Следовательно, эти разряды (или биты) уже могут передаваться. Таким образом, передача закодированной последовательности осуществляется, хотя и с некоторой задержкой, но последняя незначительна и не зависит от размера кодируемого сообщения.

И третья проблема – это вопрос точности представления. Из приведенного примера видно, что точность представления интервала (число десятичных разрядов, требуемое для его представления) неограниченно возрастает при увеличении длины кодируемого сообщения. Эта проблема обычно решается использованием арифметики с конечной разрядностью и отслеживанием переполнения разрядности регистров.

8.2.3. Сжатие данных по алгоритму словаря

Алгоритм словаря построен вокруг так называемой таблицы фраз (словаря), которая отображает строки символов сжимаемого сообщения в коды фиксированной длины, равные 12 бит. Таблица обладает свойством предшествования.

В настоящее время методы сжатия данных, включенные в протоколы MNP5 и MNP7, целенаправленно заменяются на метод, основанный на алгоритме словарного типа Лемпеля–Зива–Вэлча (LZW-алгоритме), который имеет два главных преимущества:

– обеспечивает достижение коэффициента сжатия 4:1 файлов с оптимальной структурой;

– LZW-метод утвержден ITU-T как составная часть стандарта V.42bis.

Метод сжатия данных LZW основан на создании древовидного словаря последовательностей символов, в котором каждой последовательности соответствует единственное кодовое слово. Входящий поток данных последовательно, символ за символом, сравнивается с имеющимися в словаре последовательностями. После того как в словаре будет найдена кодируемая последовательность, идентичная входной, модем передает соответствующее ей кодовое слово. Алгоритм динамически создает и обновляет словарь символьных последовательностей.

Согласно кодировке, приведённой в табл. 8.4, в двоичном коде с помощью 8 бит можно закодировать 256 символов.

Таблица 8.4.

Кодирование буквенно-цифровых знаков

				8 *b*7 *b*6 *b*5			0 0 1 0						0 0	0 0 1						
*b*4	*b*3	*b*2	*b*1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0			Пробел	0	@	P		p					ю	п	Ю	П
0	0	0	1	1	┤		!	1	A	Q	a	q			i	±	а	я	А	Я
0	0	1	0	2			″	*2*	B	R	b	r			¢	2	6	р	Б	Р
0	0	1	1	3			#	3	C	S	*c*	s			£	3	ц	с	Ц	С

0	1	0	0	4	↑		¤	4	D	T	d	t			$	x	д	т	Д	Т
0	1	0	1	5	↓		%	5	E	U	e	u			¥		е	у	Е	У
0	1	1	0	6	→		&	6	F	V	f	v			#		Ф	ж	Ф	Ж
0	1	1	1	7	←		,	7	G	W	g	w			§	,	г	в	Г	В
1	0	0	0	8			(	8	H	X	h	x			¤	•r	х	ь	Х	Ь
1	0	0	1	9			)	9	I	Y	i	y					и	ы	И	Ы
1	0	1	0	10			*	:	J	Z	j	z					й	з	Й	З
1	0	1	1	11	├		+	;	K	[	k	{			«	»	к	ш	К	Ш
1	1	0	0	12			˙	<	L	\	l					¼	л	э	Л	Э
1	1	0	1	13			-	=	M	]	m	}				½	м	щ	М	Щ
1	1	1	0	14			.	>	N	¬	n	-				¾	н	ч	Н	Ч
1	1	1	1	15				?	O	_	o					¿	о	ъ	О	

Эти символы (вернее, их коды изначально заносятся в словарь программы, реализующей LZW). Во время работы программа посимвольно перебирает строку, подлежащую сжатию и передаче. При этом выполняется такая последовательность действий.

– Считываемый символ добавляется в формирующую строку. Если полученная строка уже присутствует в словаре, проверяется следующий символ.

– если полученной строки в словаре нет, передается предыдущая сформированная строка, а новая заносится в словарь.

Таким образом, считываемые символы используются для формирования отсутствующих в словаре строк, длина которых с каждым выполнением цикла сжатия увеличивается. Если обнаруживается, что такой последовательности символов в словаре еще нет, последняя сформированная строка передается на выход, а новая строка добавляется в словарь. Для указания положения строки в таблице строк словаря в алгоритме LZW используется числовой код. Если сформированную строку условно назвать префиксом, а считываемый символ – суффиксом, то работу алгоритма можно описать следующим образом:

префикс + суффикс = новая строка

После формирования новой строки суффикс становится префиксом:

префикс = суффикс

В качестве примера рассмотрим, как с помощью алгоритма LZW выполняется сжатие строки *ababc,* которая была передана модему терминалом. Вначале каждому символу словаря назначается числовое кодовое значение, соответствующее десятеричному представлению этого символа в кодировке ASCII. То есть кодовое значение символа *a* равно 97, кодовое значение символа *b* – 98 и т. д.

В соответствии с алгоритмом LZW, при первой выполняемой операции (первом цикле) принимается, что префиксом является пустая строка, которую мы обозначим символом *f*. Поэтому при выполнении первой операции первый считываемый символ *a* добавляется к пустой строке, в результате чего формируется новая строка *a.* Поскольку *a* присутствует в словаре, на выход ничего не

передается. Далее, согласно алгоритму, суффикс становится префиксом – *a* становится префиксом при формировании новой строки (этот этап сжатия строки *ababc* отображен в первой строке табл. 8.5).

Таблица 8.5.

Сжатие строки *ababc* в соответствии с алгоритмом LZW

Префикс	Суффикс	Новая строка	Выход
f	*A*	*A*	–
a	*B*	*ab*	97
b	*A*	*ba*	98
a	*B*	*ab*	–
ab	*C*	*abc*	256
c	*F*	*c*	99

Следующим шагом выполнения алгоритма LZW является считывание из строки ввода второго символа – *b,* который становится суффиксом. В ходе его обработки он добавляется к префиксу *a*, и в результате образуется новая строка *ab.* Этой строки нет в словаре программы, поэтому вступает в силу второе правило, согласно которому на выход передается последняя сформированная строка *a*, кодовое значение которой равно 97, а новая строка *ab* добавляется в словарь. Ранее уже говорилось, что для представления символов в кодировке ASCII используется 8 бит, что позволяет работать с 255 символами. Из этого следует, что новым строкам можно присвоить кодовые значения, которые будут больше 255 (256 и т. д.) и которые в двоичном представлении требуют большего количества битов. Первоначальный размер лексемы, используемый для представления новых строк, согласовывается модемами во время процесса согласования, выполняемого в соответствии со стандартом V.42bis.

Однако вернемся к рассмотрению процесса сжатия. Символ *b,* который был суффиксом при формировании строки *ab*, стал префиксом для следующей операции (это отображено в третьей строке табл. 8.5).

Далее считывается следующий символ – *a*, который тут же используется как суффикс при создании новой строки *ba.* Поскольку этой строки нет в словаре, на выход передается предыдущая строка из числа еще не переданных, *b*, кодовое значение которой равно 98 (в соответствии с кодировкой ASCII). Заметьте, что сформированная перед этим строка *ab* была добавлена в словарь, а не отправлена на выход. При добавлении в словарь строки *ba* ей присваивается следующий код – 257, а символ *a*, который был суффиксом при формировании этой строки, при выполнении следующей операции становится префиксом, что отражено в четвертой строке табл. 6.4. Затем считывается очередной (четвертый) символ строки ввода – *b*, при добавлении которого в качестве суффикса к предыдущей строке (*a*) образуется новая строка *ab*. Однако поскольку она уже

была добавлена в таблицу строк (словарь), на выход ничего не передается, а сама строка становится префиксом при создании следующей строки.

Данный этап процесса сжатия отражен в пятой строке табл. 8.5 сформированная на предыдущем этапе строка *ab*, которая ранее была занесена в таблицу строк, стала префиксом при создании следующей строки, а последний символ *c* стал суффиксом. Полученная новая строка *abc* отсутствует в словаре, поэтому на выход передается последняя сформированная и не переданная строка – *ab*, точнее, передается присвоенное ей кодовое значение – 256. Символ *c* становится префиксом для создаваемой очередной строки, но так как он является последним символом строки ввода, его кодовое значение (99) передается на выход.

Декодер LZW должен использовать тот же словарь, что и кодер, строя его по аналогичным правилам при восстановлении сжатых данных. Каждый считываемый код разбирается с помощью словаря на предшествующую фразу *w* и символ *K*. Затем рекурсия продолжается для предшествующей фразы *w* до тех пор пока она не окажется кодом одного символа. При этом завершается декомпрессия этого кода. Обновление словаря происходит для каждого декодируемого кода, кроме первого. После завершения декодирования кода его последний символ, соединенный с предыдущей фразой, добавляется в словарь. Новая фраза получает то же значение кода (позицию в словаре), которое присвоил ей кодер. В результате такого процесса декодер шаг за шагом восстанавливает тот словарь, который построил кодер.

Важное значение имеют алгоритмы сжатия LZ и LZW при архивации данных. Популярные архиваторы ARJ, PAK, LHARC PKZ1P работают на основе этих алгоритмов.

8.2.4. Кодирование повторов

8.2.4.1 Кодирование последовательностей повторяющихся символов, метод RLE предусматривает замену последовательности повторяющихся символов на строку, содержащую этот символ, и число, соответствующее количеству его повторений. В качестве примера рассмотрим сжатие последовательности символов *ACCOUNTbbbbbbbMOUNT*, в которой *b* означает символ пробела. Если для обозначения выполненного сжатия символов пробела модем использует специальный символ *Sc*, то между модемами будет передана последовательность символов *ACCONTSc7MOUNT*. Символ *Sc* в этой последовательности означает, что было произведено сжатие символов пробела, а число 7 указывает, сколько именно символов пробела заменено символом *Sc*. С помощью этой информации принимающий модем может восстановить данные.

Однако в последовательности передаваемых символов может встретиться пара символов *S* и *c*, которые являются частью данных, а не специальным символом *Sc*, обозначающим сжатие. Чтобы принимающий модем воспринимал эти символы как данные, передающий модем при обнаружении пары символов *Sc* добавляет в передаваемую последовательность еще одну такую пару. Таким образом, если модем принял от терминала поток данных *XYZScABC*, то по те-

лефонному каналу он передаст следующую последовательность символов: *XYZScScABC*. На принимающем модеме при обнаружении первого специального символа *Sc* проверяется следующий символ. Если им окажется не число, а еще один такой символ, модем отбросит второй символ и восстановит первоначальный поток данных.

Сжатие позволяет увеличить пропускную способность систем передачи данных, но если один или более символов будут переданы с ошибкой, это может привести к очень печальным последствиям при восстановлении потока данных. В качестве примера покажем, к чему может привести ошибка при передаче последовательности символов *AAAAAAAA*. Предположим, что используется алгоритм сжатия RLE, в котором символ, означающий сжатие, представлен последовательностью битов 11111111, а символ *A* – последовательностью 01000001 (табл. 8.6.).

На рис. 8.6 демонстрируется, к чему может привести ошибка всего лишь в одном символе переданной последовательности битов.

Как видно из рассмотренного примера, последствия ошибки в символе при передаче сжатых данных намного серьезнее, чем при наличии такой же ошибки в случае обычной передачи. Поэтому во всех модемах, выполняющих сжатие данных, имеются встроенные программы, осуществляющие кодирование и декодирование сжатой последовательности в одном из корректирующих кодов.

– Последовательность одинаковых символов: AAAAAAAA
– Последовательность символов после сжатия: *S*c 8 *A*
– Двоичное представление передаваемых данных: 11111111 00001000 01000001

– Ошибка в символе: 11111111 00001000 01000011
– Принятая последовательность символов: *S*c 8 *C*

– Распакованная последовательность символов: CCCCCCCC

Рис. 8.6. Последствие ошибки в одном бите переданной сжатой строки

8.2.4.2. Кодирование длин повторений, может быть достаточно эффективным при сжатии двоичных данных, например, черно-белых факсимильных изображений, черно-белых изображений, содержащих множество прямых линий и однородных участков, схем и т.п. Кодирование длин повторений является одним из элементов известного алгоритма сжатия изображений JPEG.

Идея сжатия данных на основе кодирования длин повторений состоит в том, что вместо кодирования собственно данных подвергаются кодированию числа, соответствующие длинам участков, на которых данные сохраняют неизменное значение.

Предположим, что нужно закодировать двоичное (двухцветное) изображение размером 8 х 8 элементов, приведенное на рис. 8.2.

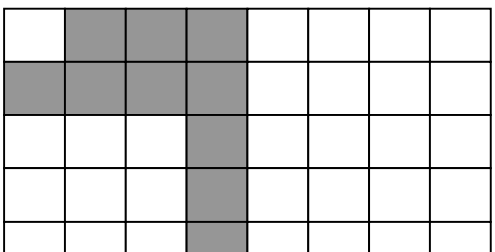


Рис.8.2. Двухцветное изображение 8 х 8 элементов

Просканируем это изображение по строкам (двум цветам на изображении будут соответствовать 0 и 1), в результате получим двоичный вектор данных

***X*=** (0111000011110000000100000001000000010000000111100011110111101111)

длиной 64 бита (скорость исходного кода составляет 1 бит на элемент изображения).

Выделим в векторе ***X*** участки, на которых данные сохраняют неизменное значение, и определим их длины. Результирующая последовательность длин участков - положительных целых чисел, соответствующих исходному вектору данных ***X*, -** будет иметь вид ***r*** = (1, 3, 4, 4, 7, 1, 7, 1, 7, 1, 7, 4, 3, 4, 1, 4, 1, 4). Теперь эту последовательность, в которой заметна определенная повторяемость (единиц и четверок гораздо больше, чем других символов), можно закодировать каким-либо статистическим кодом, например, кодом Хаффмена без памяти, имеющим таблицу кодированием (табл.8.6.)

Таблица 8.6

Таблица кодирования длин участков

Кодер	
Длина участка	Кодовое слово
4	0
1	10
7	110
3	111

Для того, чтобы указать, что кодируемая последовательность начинается с нуля, добавим в начале кодового слова префиксный символ 0. В результате получим кодовое слово ***B (r)*** = (01 0001101011010110101100111010010 0) длиной в 34 бита, то есть результирующая скорость кода ***R*** составит 34/64, или немногим более 0,5 бита на элемент изображения. При сжатии изображений большего размера и содержащих множество повторяющихся элементов эффективность сжатия может оказаться существенно более высокой.

Ниже приведен другой пример использования кодирования длин повторений, когда в цифровых данных встречаются участки с большим количеством нулевых значений. Всякий раз, когда в потоке данных встречается “*ноль*”, он

кодируется двумя числами. Первое - 0, являющееся флагом начала кодирования длины потока нулей, и второе – количество нулей в очередной группе. Если среднее число нулей в группе больше двух, будет иметь место сжатие. С другой стороны, большое число отдельных нулей может привести даже к увеличению размера кодируемого файла:

Еще одним простым и широко используемым для сжатия изображений и звуковых сигналов методом неразрушающего кодирования является метод дифференциального кодирования.

8.2.5. Дифференциальное кодирование

Работа дифференциального кодера основана на том факте, что для многих типов данных разница между соседними отсчетами относительно невелика, даже если сами данные имеют большие значения. Например, нельзя ожидать большой разницы между соседними пикселами цифрового изображения.

Следующий простой пример показывает, какое преимущество может дать дифференциальное кодирование (кодирование разности между соседними отсчетами) в сравнении с простым кодированием без памяти (кодированием отсчетов независимо друг от друга).

Просканируем 8-битовое (256-уровневое) цифровое изображение, при этом десять последовательных пикселов имеют уровни:

144, 147, 150, 146, 141, 142, 138, 143, 145, 142.

Если закодировать эти уровни пиксел за пикселом каким-либо кодом без памяти, использующим 8 бит на пиксел изображения, получим кодовое слово, содержащее 80 бит.

Предположим теперь, что прежде чем подвергать отсчеты изображения кодированию, мы вычислим разности между соседними пикселами. Эта процедура даст нам последовательность следующего вида:

144, 147, 150, 146, 141, 142, 138, 143, 145, 142.

⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓

144, 3, 3, - 4, - 5, 1, - 4, 5, 2, -3.

Исходная последовательность может быть легко восстановлена из разностной простым суммированием (дискретным интегрированием):

144, 144+3, 147+3, 150–4, 146–5, 141+1, 142–4, 138+5, 143+2, 145-3

⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓ ⇓

144, 147, 150, 146, 141, 142, 138, 143, 145, 142.

Для кодирования первого числа из полученной последовательности разностей отсчетов, как и ранее, понадобится 8 бит, все остальные числа можно закодировать 4-битовыми словами (один знаковый бит и 3 бита на кодирование модуля числа).

Таким образом, в результате кодирования получим кодовое слово длиной ***8 + 9*4 = 44*** бита или почти вдвое более короткое, нежели при индивидуальном кодировании отсчетов.

Метод дифференциального кодирования очень широко используется в тех случаях, когда природа данных такова, что их соседние значения незначительно отличаются друг от друга, притом, что сами значения могут быть сколь угодно большими.

Это относится к звуковым сигналам, особенно к речи, изображениям, соседние пиксели которых имеют практически одинаковые яркости и цвет и т.п. В то же время этот метод совершенно не подходит для кодирования текстов, чертежей или каких-либо цифровых данных с независимыми соседними значениями.

8.3. Методы сжатия с потерей информации

Как уже ранее отмечалось, существуют два типа систем сжатия данных:

– без потерь информации (неразрушающие);
– с потерями информации (разрушающие).

При неразрушающем кодировании исходные данные могут быть восстановлены из сжатых в первоначальном виде, то есть абсолютно точно. Такое кодирование применяется для сжатия текстов, баз данных, компьютерных программ и данных и т.п., где недопустимо их даже малейшее различие. Все рассмотренные выше методы кодирования относились именно к неразрушающим.

К сожалению, неразрушающее сжатие, при всей привлекательности перспективы получить *абсолютное совпадение* исходных и восстановленных данных, имеет невысокую эффективность – коэффициенты неразрушающего сжатия редко превышают 3…5 (за исключением случаев кодирования данных с высокой степенью повторяемости одинаковых участков и т.п.).

Вместе с тем очень часто нет необходимости в абсолютной точности передачи исходных данных потребителю. Во-первых, сами источники данных обладают ограниченным динамическим диапазоном и вырабатывают исходные сообщения с определенным уровнем искажений и ошибок. Этот уровень может быть большим или меньшим, но абсолютной точности воспроизведения достичь невозможно.

Во-вторых, передача данных по каналам связи и их хранение всегда производятся при наличии различного рода помех. Поэтому принятое (воспроизведенное) сообщение всегда в определенной степени отличается от передан-

ного, то есть на практике невозможна абсолютно точная передача при наличии помех в канале связи (в системе хранения).

Наконец, сообщения передаются и сохраняются для их восприятия и использования получателем. Получатели же информации - органы чувств человека, исполнительные механизмы и т.д. - также обладают конечной разрешающей способностью, то есть не замечают незначительной разницы между *абсолютно точным* и *приближенным* значениями воспроизводимого сообщения. Порог чувствительности к искажениям также может быть различным, но он всегда есть.

Кодирование с разрушением учитывает эти аргументы в пользу *приближенного восстановления данных* и позволяет получить за счет некоторой контролируемой по величине ошибки коэффициенты сжатия, иногда в десятки раз превышающие степень сжатия для неразрушающих методов.

Большинство методов разрушающего сжатия основано на кодировании не самих данных, а некоторых линейных преобразований от них, например, коэффициентов дискретного преобразования Фурье (ДПФ), коэффициентов косинусного преобразования, преобразований Хаара, Уолша и т.п.

Для того, чтобы понять, на чем основана высокая эффективность систем разрушающего сжатия и почему кодирование преобразований оказывается значительно более эффективным, нежели кодирование исходных данных, рассмотрим в качестве примера популярный метод сжатия изображений ***JPEG*** (“джипег”), который содержит в себе все необходимые атрибуты системы разрушающего сжатия. Не будем сейчас вдаваться в формульные дебри, главное – понять идею кодирования преобразований.

Нужно будет также рассмотреть и сущность самих линейных преобразований, применяемых для сжатия, поскольку без понимания их физического смысла трудно уяснить причины получаемых эффектов.

8.3.1. Кодирование преобразований. Стандарт сжатия JPEG

Популярный стандарт кодирования изображений ***JPEG*** (Joint Photographers Experts Group) является очень хорошей иллюстрацией к объяснению принципов разрушающего сжатия на основе кодирования преобразований.

Основную идею кодирования преобразований можно понять из следующих простых рассуждений. Допустим, мы имеем дело с некоторым цифровым сигналом (последовательностью отсчетов Котельникова). Если отбросить в каждом из отсчетов половину двоичных разрядов (например, 4 разряда из восьми), то *вдвое уменьшится скорость кода* ***R*** *и потеряется половина информации*, содержащейся в сигнале.

Если же подвергнуть сигнал преобразованию Фурье (или какому-либо другому подобному линейному преобразованию), разделить его на две составляющие – НЧ и ВЧ, продискретизовать, подвергнуть квантованию каждую из них и отбросить половину двоичных разрядов только в высокочастотной составляющей сигнала, то результирующая *скорость кода уменьшится на одну треть, а потеря информации составит всего 5%.*

Этот эффект обусловлен тем, что низкочастотные составляющие большинства сигналов (крупные детали) обычно гораздо более интенсивны и

несут гораздо больше информации, нежели высокочастотные составляющие (мелкие детали). Это в равной степени относится и к звуковым сигналам, и к изображениям.

Рассмотрим работу алгоритма сжатия ***JPEG*** при кодировании черно-белого изображения, представленного набором своих отсчетов (пикселов) с числом градаций яркости в 256 уровней (8 двоичных разрядов). Это самый распространенный способ хранения изображений - каждой точке на экране соответствует один байт (8 бит - 256 возможных значений), определяющий её яркость. 255 - яркость максимальная (белый цвет), 0 - минимальная (черный цвет). Промежуточные значения составляют всю остальную гамму серых цветов (рис. 8.3).

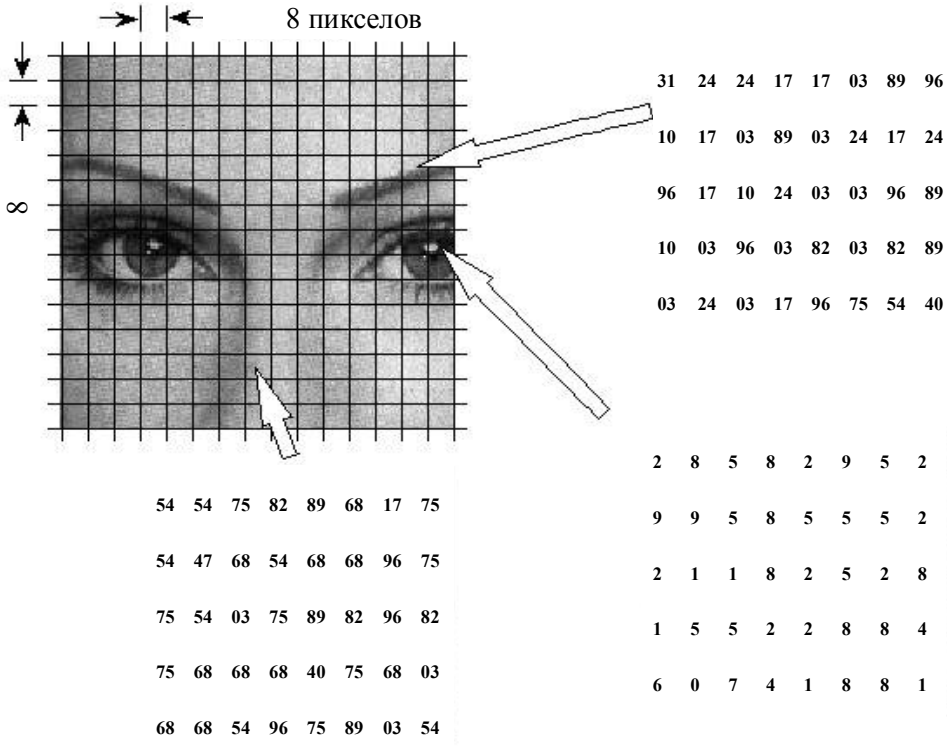


Рис. 8.3. Чёрно-белое изображение, продлежащее кодированию

Работа алгоритма сжатия ***JPEG*** начинается *с разбиения изображения на квадратные блоки* размером 8x8 = 64 пиксела. Почему именно 8x8, а не 2x2 или 32x32? Выбор такого размера блока обусловлен тем, что при его малом размере эффект кодирования будет небольшим (при размере 1x1 – вообще отсутствует), а при большом - свойства изображения в пределах блока будут сильно изменяться и эффективность кодирования снова снизится.

На рис. 8.3 изображено несколько таких блоков (в виде матриц цифровых отсчетов), взятых из различных участков изображения. В дальнейшем эти блоки будут обрабатываться, и кодироваться независимо друг от друга.

Второй этап сжатия – *применение ко всем блокам дискретного косинусного преобразования* – ***DCT***. Для сжатия данных пытались применить множество различных преобразований, в том числе специально разработанных для этих целей, например, преобразование Карунены-Лоэва, обеспечивающее макси-

мально возможный коэффициент сжатия. Но оно очень сложно реализуется на практике. Преобразование Фурье выполняется очень просто, но не обеспечивает хорошего сжатия. Выбор был остановлен на дискретном косинусном преобразовании, являющем разновидностью ПФ. В отличие от ПФ, которое применяет для разложения сигнала синусные и косинусные частотные составляющие, в ***DCT*** используются только косинусные составляющие. Дискретное косинусное преобразование позволяет перейти от пространственного представления изображения (набором отсчетов или пикселов) к спектральному представлению (набором частотных составляющих) и наоборот.

Дискретное косинусное преобразование от изображения ***IMG (x,y)*** может быть записано следующим образом:

$$DCT(u,v) = sqrt(2/N) \sum_{i,j} IMG(x_i, y_j)\cos((2i+1)\pi u/2N)\cos((2j+1)\pi v/2N), \quad (8.4)$$

где $N = 8,\ 0 < i < 7,\ 0 < j < 7,$

или же, в матричной форме,

$$RES = DCT^T * IMG * DCT, \quad (8.5)$$

где ***DCT*** - матрица базисных (косинусных) коэффициентов для преобразования размером 8x8, имеющая вид:

$$DCT = \begin{vmatrix} .353553 & .353553 & .353553 & .353553 & .353553 & .353553 & .353553 & .353553 \\ .490393 & .415818 & .277992 & .097887 & -.097106 & -.277329 & -.415375 & -.490246 \\ .461978 & .191618 & -.190882 & -.461673 & -.462282 & -.192353 & .190145 & .461366 \\ .414818 & -.097106 & -.490246 & -.278653 & .276667 & .490710 & .099448 & -.414486 \\ .353694 & -.353131 & -.354256 & .352567 & .354819 & -.352001 & -.355378 & .351435 \\ .277992 & -.490246 & .096324 & .416700 & -.414486 & -.100228 & .491013 & -.274673 \\ .191618 & -.462282 & .461366 & -.189409 & -.193822 & .463187 & -.460440 & .187195 \\ .097887 & -.278653 & .416700 & -.490862 & .489771 & -.413593 & .274008 & -.092414 \end{vmatrix} \quad (8.6)$$

Итак, в результате применения к блоку изображения размером 8x8 пикселов дискретного косинусного преобразования получим двумерный спектр, также имеющий размер 8x8 отсчетов. Иными словами, *64 числа, представляющие отсчеты изображения, превратятся в 64 числа, представляющие отсчеты его* ***DCT****-спектра.*

А теперь напомним, что такое спектр сигнала. Это – величины коэффициентов, с которыми соответствующие спектральные составляющие входят в сумму, которая в результате и дает этот сигнал. Отдельные спектральные составляющие, на которые раскладывается сигнал, часто называют базисными функциями. Для ПФ базисными функциями являются синусы и косинусы разных частот.

Для 8x8 ***DCT*** система базисных функций задается формулой

$$b[x,y] = \cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right], \qquad (8.7)$$

а сами базисные функции выглядят подобно приведенным на рис. 8.4.

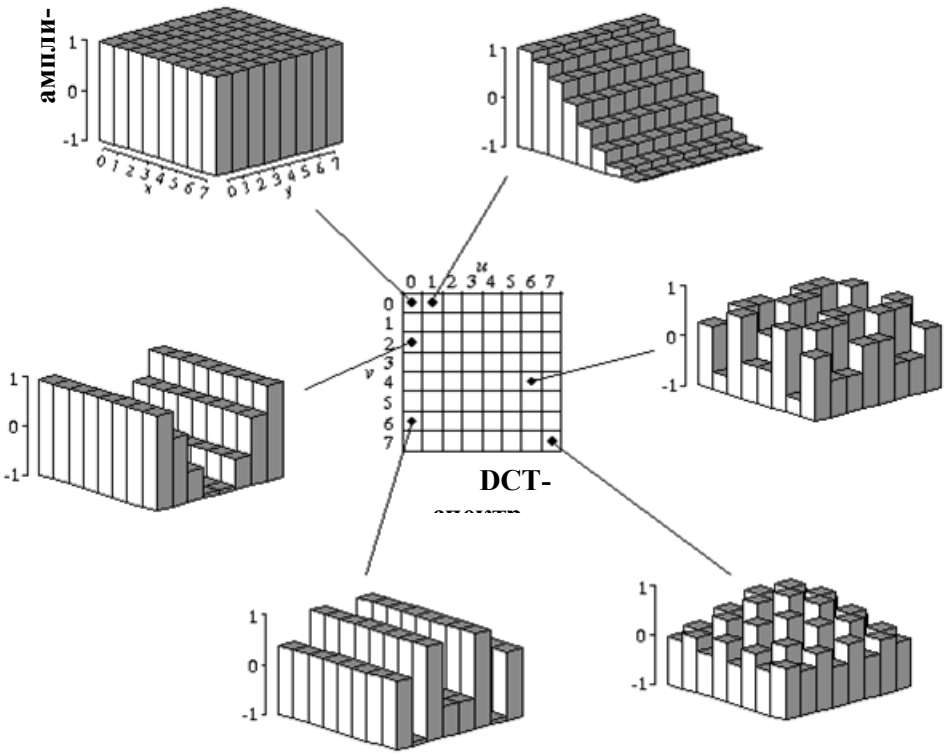


Рис. 8.4. Вид базисных функций

Самая низкочастотная базисная функция, соответствующая индексам (0,0), изображена в левом верхнем углу рисунка, самая высокочастотная – в нижнем правом. Базисная функция для (0,1) представляет собой половину периода косинусоиды по одной координате и константу - по другой, базисная функция с индексами (1,0) – то же самое, но повернута на 90^0. Дискретное косинусное преобразование вычисляется путем поэлементного перемножения и суммирования блоков изображения 8x8 пикселов с каждой из этих базисных функций. В результате, к примеру, компонента ***DCT***-спектра с индексами (0,0) будет представлять собой просто сумму всех элементов блока изображения, то есть среднюю для блока яркость. В компоненту с индексом (0,1) усредняются с одинаковыми весами все горизонтальные детали изображения, а по вертикали наибольший вес присваивается элементам верхней части изображения и т.д. Можно заметить, что чем ниже и правее в матрице ***DCT*** его компонента, тем более высокочастотным деталям изображения она соответствует. Для того, чтобы получить исходное изображение по его ***DCT-***спектру (выполнить обратное преобразование), нужно теперь базисную функцию с индексами (0,0) умножить на спектральную компоненту с координатами (0,0), прибавить к ре-

зультату произведение базисной функции (1,0) на спектральную компоненту (1,0) и т.д.

В приведенной ниже табл. 8.7 видны числовые значения одного из блоков изображения и его ***DCT***-спектра:

Таблица 2.9

Числовые значения блока изображения

Исходные данные							
139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	161	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
161	162	161	163	162	157	157	157
162	162	161	161	163	158	158	15
Результат DCT							
235,6	-1	-12,1	-5,2	2,1	-1,7	-2,7	1,3
-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1
-7,1	-1,9	0,2	1,5	0,9	-0,1	0	0,3
-0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3
1,8	-0,2	1,6	-0,3	-0,8	1,5	1	-1
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4

Отметим очень интересную особенность полученного ***DCT***-спектра: наибольшие его значения сосредоточены в левом верхнем углу табл. 8.7 (низкочастотные составляющие), правая же нижняя его часть (высокочастотные составляющие) заполнена относительно небольшими числами. Чисел этих, правда, столько же, как и в блоке изображения: 8x8 = 64, то есть пока никакого сжатия не произошло, и, если выполнить обратное преобразование, получим тот же самый блок изображения. Следующим этапом работы алгоритма JPEG является *квантование* (табл. 8.8).

Таблица 8.8

Процесс квантования

Ранее полученный результат DCT							
235,6	-1	-12,1	-5,2	2,1	-1,7	-2,7	1,3

-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1
-7,1	-1,9	0,2	1,5	0,9	-0,1	0	0,3
-0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3
1,8	-0,2	1,6	-0,3	-0,8	1,5	1	-1
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4
Таблица квантования							
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99
Результат квантования							
15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Если внимательно посмотреть на полученные в результате ***DCT*** коэффициенты, то будет видно, что добрая их половина - нулевые или имеет очень небольшие значения (1 - 2). Это высокие частоты, которые (обычно) могут быть безболезненно отброшены или, по крайней мере, округлены до ближайшего целого значения. Квантование заключается в делении каждого коэффициента ***DCT*** на некоторое число в соответствии с матрицей квантования. Эта матрица может быть фиксированной либо, для более качественного и эффективного сжатия, получена в результате анализа характера исходной картинки. Чем больше числа, на которые происходит деление, тем больше в результате деления будет нулевых значений, а значит, сильнее сжатие и заметнее потери.

Совершенно очевидно, что от выбора таблицы квантования будет в значительной степени зависеть как эффективность сжатия – число нулей в квантованном (округленном) спектре,– так и качество восстановленной картинки. Та-

ким образом, мы округлили результат ***DCT*** и получили в большей или меньшей степени искаженный поблочный спектр изображения. Следующим этапом работы алгоритма ***JPEG*** является преобразование 8x8 матрицы ***DCT***-спектра в линейную последовательность. Но делается это таким образом, чтобы сгруппировать по возможности вместе все большие значения и все нулевые значения спектра. Совершенно очевидно, что для этого нужно прочесть элементы матрицы коэффициентов ***DCT*** в порядке, изображенном на рис. 8.5, то есть зигзагообразно - из левого верхнего угла к правому нижнему. Эта процедура называется *зигзаг-сканированием*.

В результате такого преобразования квадратная матрица 8x8 квантованных коэффициентов ***DCT*** превратится в линейную последовательность из 64 чисел, большая часть из которых – это идущие подряд нули. Известно, что такие потоки можно очень эффективно сжимать путем *кодирования длин повторений.* Именно так это и делается.

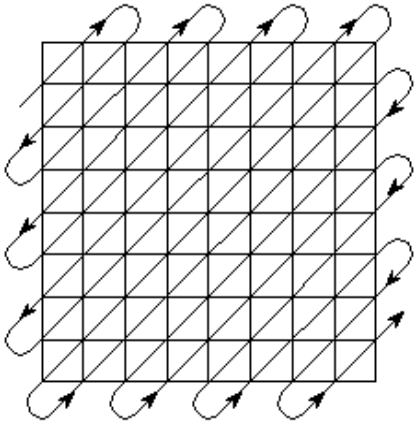


Рис. 8.5. Порядок чтения элементов матрицы коэффициентов

На следующем, пятом этапе ***JPEG***-кодирования получившиеся цепочки нулей подвергаются кодированию длин повторений.

И, наконец, последним этапом работы алгоритма ***JPEG*** является кодирование получившейся последовательности каким-либо статистическим алгоритмом. Обычно используется арифметическое кодирование или алгоритм Хаффмена. В результате получается новая последовательность, размер которой существенно меньше размера массива исходных данных.

Последние два этапа кодирования обычно называют вторичным сжатием, и именно здесь происходит неразрушающее статистическое кодирование, и с учетом характерной структуры данных - существенное уменьшение их объема.

Декодирование данных сжатых согласно алгоритму ***JPEG*** производится точно так же, как и кодирование, но все операции следуют в обратном порядке.

После неразрушающей распаковки методом Хаффмена (или LZW, или арифметического кодирования) и расстановки линейной последовательности в

блоки размером 8x8 чисел *спектральные компоненты деквантуются* с помощью сохраненных при кодировании таблиц квантования. Для этого распакованные 64 значения ***DCT*** умножаются на соответствующие числа из таблицы. После этого каждый блок подвергается обратному косинусному преобразованию, процедура которого совпадает с прямым и различается только знаками в матрице преобразования. Последовательность действий при декодировании и полученный результат иллюстрируются приведенной ниже табл. 8.9.

Таблица 8.9

Последовательность действий при декодировании

Квантованные данные							
15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
Деквантованные данные							
240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
Результат обратного DCT							
144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

Для сравнения - исходные данные							
139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	161	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
161	162	161	163	162	157	157	157
162	162	161	161	163	158	158	15

Очевидно, что восстановленные данные несколько отличаются от исходных. Это естественно, потому что ***JPEG*** и разрабатывался, как сжатие с потерями. На представленном ниже рис. 8.6 приведено исходное изображение (справа), а также изображение, сжатое с использованием алгоритма ***JPEG*** в 10 раз (слева) и в 45 раз (в центре). Потеря качества в последнем случае вполне заметна, но и выигрыш по объему тоже очевиден.

Итак, ***JPEG***-сжатие состоит из следующих этапов:

– Разбиение изображения на блоки размером 8x8 пикселов.

– Применение к каждому из блоков дискретного косинусного преобразования.

–Округление коэффициентов ***DCT*** в соответствии с заданной матрицей весовых коэффициентов.

– Преобразование матрицы округленных коэффициентов ***DCT*** в линейную последовательность путем их зигзагообразного чтения.

– Кодирование повторений для сокращения числа нулевых компнент.

– Статистическое кодирование результата кодом Хаффмена или арифметическим кодом.

Рис. 8.6. Вид изображений до и после сжатия

Декодирование производится точно так же, но в обратном порядке.

Существенными положительными сторонами алгоритма сжатия ***JPEG*** являются:

– возможность задания в широких пределах (от 2 до 200) степени сжатия;

– возможность работы с изображениями любой разрядности;

– симметричность процедур сжатия – распаковки.

К недостаткам можно отнести наличие ореола на резких переходах цветов - эффект Гиббса, а также распадение изображения на отдельные квадратики 8x8 при высокой степени сжатия.

8.3.2. Фрактальный метод

Фрактальное сжатие основано на том, что изображение представляется в более компактной форме с помощью коэффициентов системы итерируемых функций (Iterated Function System – ***IFS***).

Прежде чем рассматривать сам процесс фрактального сжатия, разберемся, как ***IFS*** строит изображение, то есть рассмотрим процесс декомпрессии. Строго говоря, ***IFS*** представляет собой набор трехмерных аффинных преобразований, в нашем случае переводящих одно изображение в другое. Преобразованию подвергаются точки в трехмерном пространстве (координата точки изображения ***X***, координата точки изображения ***Y*** и яркость точки ***I***). Упрощенно этот процесс можно пояснить следующим образом. Рассмотрим так называемую фотокопировальную машину (рис. 8.7), состоящую из экрана, на котором изображена исходная картинка, и системы линз, проецирующих изображение на другой экран. Фотокопировальная машина может выполнять следующие действия:

–линзы могут проецировать часть изображения *произвольной формы* в любое другое место нового изображения;

–области, *в которые* проецируются изображения, *не пересекаются*;

–линза может *менять яркость и уменьшать контрастность*;

–линза может *зеркально отражать и поворачивать* свой фрагмент изображения;

– линза *должна масштабировать* (уменьшать) свой фрагмент изображения.

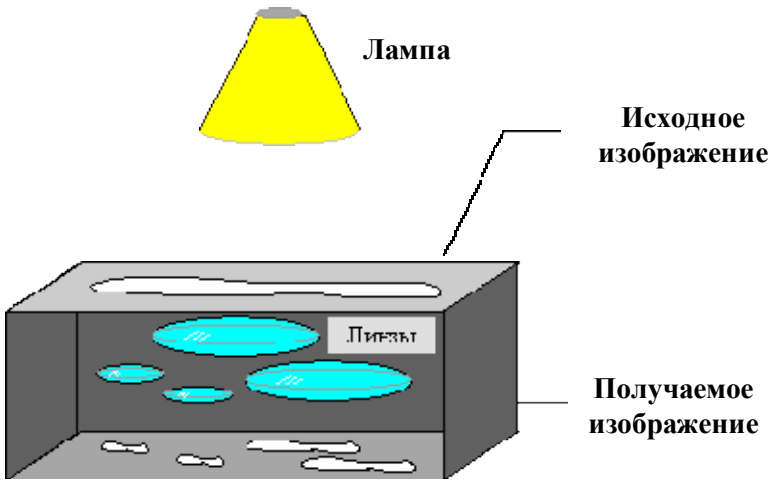


Рис. 8.7. Установка, поясняющая принцип фрактального метода

Расставляя линзы и меняя их характеристики, можно управлять получаемым изображением. Одна итерация работы машины заключается в том, что по исходному изображению с помощью проектирования строится новое, после чего новое берется в качестве исходного. Утверждается, что в процессе итераций мы получим изображение, которое перестанет изменяться. Оно будет зависеть только от расположения и характеристик линз и не будет зависеть от исходной картинки. Это изображение называется "*неподвижной точкой*", или *аттрактором* данной ***IFS***. Соответствующая теория гарантирует наличие ровно одной неподвижной точки для каждой ***IFS***.

Поскольку отображение линз является сжимающим, каждая линза в явном виде задает самоподобные области в нашем изображении. Благодаря самоподобию получаем сложную структуру изображения при любом увеличении. Таким образом, интуитивно понятно, что система итерируемых функций задает *фрактал* – самоподобный математический объект.

Наиболее известным из изображений, полученных с помощью ***IFS,*** является так называемый "папоротник Барнсли" (рис. 8.8), задаваемый четырьмя аффинными преобразованиями (или, в нашей терминологии, "линзами"). Каждое преобразование кодируется буквально считанными байтами, в то время как изображение, построенное с их помощью, может занимать и несколько мегабайт.

На этом изображении можно выделить 4 области, объединение которых покрывает все изображение и каждая из которых подобна всему изображению (не забывайте о стебле папоротника).

Рис. 8.8. "Папоротник Барнсли"

Из вышесказанного становится понятно, как работает фрактальный кодер, и также очевидно, что для сжатия ему понадобится очень много времени.

Фактически фрактальная компрессия – это поиск самоподобных областей в изображении и определение для них параметров аффинных преобразований. Для этого потребуется выполнить перебор и сравнение всех возможных фрагментов изображения разного размера.

Даже для небольших изображений при учете дискретности мы получим астрономическое число перебираемых вариантов, причем даже резкое сужение классов преобразований, например, за счет масштабирования, только в определенное количество раз не дает заметного выигрыша во времени. Кроме того, при этом теряется качество изображения. Подавляющее большинство исследований в области фрактального сжатия сейчас направлены на уменьшение времени архивации, необходимого для получения качественного изображения.

8.3.3. Рекурсивный (волновой) алгоритм

Английское название рекурсивного сжатия – *wavelet*. На русский язык оно переводится как волновое сжатие и как сжатие с использованием всплесков. Этот вид архивации известен довольно давно и напрямую исходит из идеи использования когерентности областей. Ориентирован алгоритм на цветные и черно-белые изображения с плавными переходами, идеален для картинок типа рентгеновских снимков. Коэффициент сжатия задается и варьируется в пределах 5 - 100. При попытке задать больший коэффициент на резких границах, особенно проходящих по диагонали, проявляется “лестничный эффект” – ступеньки разной яркости размером в несколько пикселов. Идея алгоритма заключается в том, что *вместо кодирования собственно изображений сохраняется разница между средними значениями соседних блоков в изображении*, которая обычно принимает значения, близкие к 0. Так, два числа a_{2i} и a_{2i+1} всегда можно представить в виде $b^1_i = (a_{2i}+a_{2i+1})/2$ и $b^2_i=(a_{2i}-a_{2i+1})/2$. Аналогично последовательность a_i может быть попарно переведена в последовательность $b^{1,2}_i$. Рассмотрим **пример**. Пусть мы сжимаем строку из восьми значений яркости пикселов (a_i): (220, 211, 212, 218, 217, 214, 210, 202). Получим следующие последовательности b_{1i}, и b_{2i}: (215.5, 215, 215.5, 206) и (4.5, -3, 1.5, 4). Заметим, что значения b_{2i} достаточно близки к 0. Повторим операцию, рассматривая b_{1i} как a_i. Данное действие выполняется как бы рекурсивно, откуда и название алгоритма. Из (215.5, 215, 215.5, 206) получим (215.25, 210.75) (0.25, 4.75). Полученные коэффициенты, округлив до целых и сжав, например, с помощью алгоритма Хаффмана, можно считать результатом кодирования. Заметим, что мы применяли наше преобразование к цепочке только два раза. Реально можно позволить себе применение *wavelet*-преобразования 4-6 раз, что позволит достичь заметных коэффициентов сжатия. Алгоритм для двумерных данных реализуется аналогично. Если у нас есть квадрат из четырех точек с яркостями $a_{2i,2j}$, a_{2i+1}, $a_{2j,}a_{2i}$, $_{2j+1}$, и a_{2i+1}, a_{2j+1}, то

$$
\begin{aligned}
b^1_{i,j} &= (a_{2i,2j} + a_{2i+1,2j} + a_{2i,2j+1} + a_{2i+1,2j+1})/4,\\
b^2_{i,j} &= (a_{2i,2j} + a_{2i+1,2j} - a_{2i,2j+1} - a_{2i+1,2j+1})/4,\\
b^2_{i,j} &= (a_{2i,2j} - a_{2i+1,2j} + a_{2i,2j+1} - a_{2i+1,2j+1})/4,\\
b^3_{i,j} &= (a_{2i,2j} - a_{2i+1,2j} - a_{2i,2j+1} + a_{2i+1,2j+1})/4.
\end{aligned}
\qquad (8.9)
$$

Используя эти формулы, для изображения 512×512 пикселов получим после первого преобразования уже 4 матрицы размером 256×256 элементов (рис. 8.8, 8.9)

Исходное изображение	B^1	B^2
	B^3	B^4

В первой, как легко догадаться, хранится уменьшенная копия изображения, во второй – усредненные разности пар значений пикселов по горизонтали, в третьей – усредненные разности пар значений пикселов по вертикали, в четвертой – усредненные разности значений пикселов по диагонали.

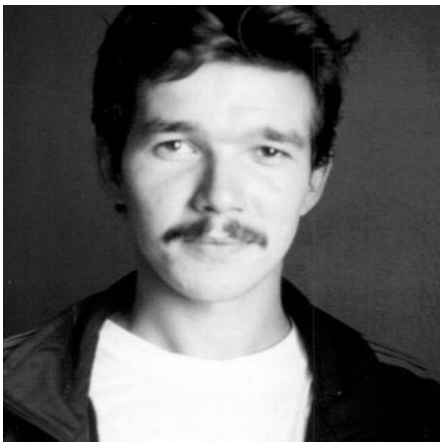


Рис. 8.8

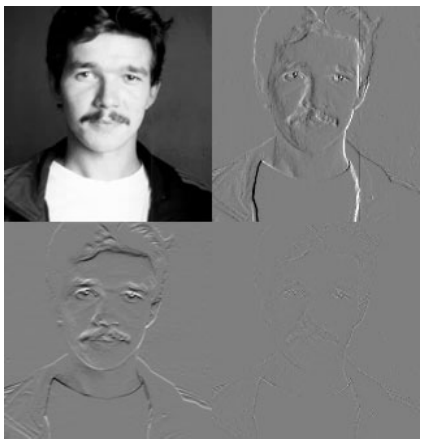


Рис. 8.9

По аналогии с двумерным случаем можно повторить преобразование и получить вместо первой матрицы 4 матрицы размером 128×128.

Повторив преобразование в третий раз, получим в итоге 4 матрицы 64×64, 3 матрицы 128×128 и 3 матрицы 256×256. Дальнейшее сжатие происходит за счет того, что в разностных матрицах имеется большое число нулевых или близких к нулю значений, которые после квантования эффективно сжимаются.

К достоинствам этого алгоритма можно отнести то, что он очень легко позволяет реализовать возможность постепенного “проявления” изображения при передаче изображения по сети. Кроме того, поскольку в начале изображения мы фактически храним его уменьшенную копию, упрощается показ “огрубленного” изображения по заголовку.

В отличие от ***JPEG*** и фрактального алгоритма данный метод не оперирует блоками, например 8x8 пикселов. Точнее, мы оперируем блоками 2x2, 4x4, 8x8 и т.д. Однако за счет того, что коэффициенты для этих блоков сохраняются независимо, можно достаточно легко избежать дробления изображения на “мозаичные” квадраты.

8.4. Методы сжатия подвижных изображений (видео)

Основной проблемой в работе с подвижными изображениями являются большие объемы данных, с которыми приходится иметь дело. Например, при записи на компакт-диск в среднем качестве на него можно поместить несколько тысяч фотографий, более 10 часов музыки и всего полчаса видео. Видео телевизионного формата – 720x576 точек и 25 кадров в секунду в системе ***RGB*** - требует потока данных примерно 240 Мбит/с (1,8 Гбит/мин). При этом обычные методы сжатия, ориентированные на кодирование отдельных кадров (в том

числе и ***JPEG***), не спасают положения, поскольку даже при уменьшении битового потока в 10 - 20 раз он остается чересчур большим для практического использования.

При сжатии подвижных изображений учитывается наличие в них нескольких типов избыточности:

- *когерентность (одноцветность) областей изображения* – незначительное изменение цвета изображения в его соседних пикселах; это свойство изображения используется при его разрушающем сжатии всеми известными методами;

- *избыточность в цветовых плоскостях,* отражающую высокую степень связи интенсивностей различных цветовых компонент изображения и важность его яркостной компоненты;

- *подобие между кадрами* – использование того факта, что при скорости 25 кадров в секунду (а это минимальная их частота, при которой незаметно мелькание изображения, связанное со сменой кадров) различие в соседних кадрах очень незначительно.

С середины 80-х гг. многие западные университеты и лаборатории фирм работали над созданием алгоритма компрессии цифрового видеосигнала. Появилось достаточно большое число внутрифирменных стандартов. Область эта очень специфична и динамична - международные стандарты появляются буквально через 2-3 года после создания алгоритма. Рассмотрим существующие стандарты в области цифрового видео.

В 1988 году в рамках Международной организации по стандартизации (ISO) начала работу группа ***MPEG*** (Moving Pictures Experts Group) - группа экспертов в области цифрового видео. В сентябре 1990 года был представлен предварительный стандарт кодирования ***MPEG-I.*** В январе 1992 года работа над MPEG-I была завершена.

Работа эта была начата не на пустом месте, и как алгоритм ***MPEG*** имеет несколько предшественников. Это прежде всего ***JPEG*** - универсальный алгоритм, предназначенный для сжатия статических полноцветных изображений. Его универсальность означает, что алгоритм дает неплохие результаты на широком классе изображений. Алгоритм использует конвейер из нескольких преобразований. Ключевым является дискретное косинусное преобразование (ДКП), позволяющее в широких пределах менять степень сжатия без заметной потери качества изображения. Последняя фраза означает, что различить на глаз восстановленное и исходное изображения практически невозможно. Идея алгоритма состоит в том, что в реальных изображениях малы амплитуды высоких частот при разложении матрицы изображения в двойной ряд по косинусам. Можно достаточно свободно огрублять их представление, не сильно ухудшая изображение. Кроме того, используется перевод в другое цветовое пространство (***YUV***), групповое кодирование и кодирование Хаффмана.

Алгоритм сжатия. Технология сжатия видео в ***MPEG*** распадается на две части: уменьшение избыточности видеоинформации во временном измерении, основанное на том, что соседние кадры, как правило, отличаются не сильно, и

сжатие отдельных изображений.

Уменьшение избыточности во временном измерении. Чтобы удовлетворить противоречивым требованиям и увеличить гибкость алгоритма, в последовательности кадров, составляющих подвижное изображение, выделяют четыре типа кадров:

– ***I-кадры*** - независимо сжатые (*I-Intrapictures*); – ***P-кадры*** - сжатые с использованием ссылки на одно изображение (*P-Predicted*);

– ***B-кадры*** - сжатые с использованием ссылки на два изображения (*B-Bidirection*);

– ***BC-кадры*** - независимо сжатые с большой потерей качества (используются только при быстром поиске).

I-кадры обеспечивают возможность произвольного доступа к любому кадру, являясь своеобразными входными точками в поток данных для декодера.

P-кадры используют при архивации ссылку на один ***I-*** или ***P-кадр***, повышая тем самым степень сжатия фильма в целом.

B-кадры, используя ссылки на два кадра, находящихся впереди и позади, обеспечивают наивысшую степень сжатия; сами в качестве ссылки использоваться не могут.

Последовательность кадров в фильме может быть, например, такой: ***I B B P B B P B B P B B P B B I B B P ....,*** или ***I P B P B P B I P B P B ...***

Частоту ***I-кадра*** выбирают исходя из требований на время произвольного доступа и надежности потока при передаче по каналу с помехами, соотношение между ***P*** и ***B-кадрами*** – исходя из необходимой степени сжатия и сложности декодера, поскольку для того, чтобы распаковать ***B-кадр***, нужно уже иметь как предшествующий, так и следующий за ним кадры.

Одно из основных понятий при сжатии нескольких изображений - макроблок. Макроблок - это матрица пикселов 16x16 элементов (размер изображения должен быть кратен 16). Такая величина выбрана не случайно - ***ДКП*** работает с матрицами размером 8×8 элементов. При сжатии каждый макроблок из цветового пространства ***RGB*** переводится в цветовое пространство ***YUV***. Матрица, соответствующая ***Y*** (яркостному компоненту), превращается в четыре исходные матрицы для ***ДКП***, а матрицы, соответствующие компонентам ***U*** и ***V***, прореживаются на все четные строки и столбцы, превращаясь в одну матрицу для ***ДКП***.

Таким образом, мы сразу получаем сжатие в два раза, пользуясь тем, что глаз человека хуже различает цвет отдельной точки изображения, чем ее яркость.

Отдельные макроблоки сжимаются независимо, т.е. в ***B-кадрах*** можно сжать макроблок как ***I-блок***, ***P-блок*** со ссылкой на предыдущий кадр, ***P-блок*** со ссылкой на последующий кадр и, наконец, как ***B-блок***.

Сжатие отдельных кадров. Существует достаточно много алгоритмов, сжимающих статические изображения. Из них чаще всего используются алгоритмы на базе дискретного косинусного преобразования. Алгоритм сжатия отдельных кадров в ***MPEG*** похож на соответствующий алгоритм для статических изображений - ***JPEG***. Напомним, как выглядит процедура ***JPEG*** -кодирования.

К макроблокам, которые готовит алгоритм уменьшения избыточности во временном измерении, применяется ***ДКП***. Само преобразование заключается в разложении значений дискретной функции двух переменных в двойной ряд по косинусам некоторых частот. Дискретное косинусное преобразование переводит матрицу значений яркостей в матрицу амплитуд спектральных компонент, при этом амплитуды, соответствующие более низким частотам, записываются в левый верхний угол матрицы, а те, которые соответствуют более высоким, - в правый нижний. Поскольку в реалистичных изображениях высокочастотная составляющая очень мала по амплитуде, в результирующей матрице значения под побочной диагональю либо близки, либо равны нулю.

К полученной матрице амплитуд применяется операция квантования. Именно на этапе квантования - группового кодирования - в основном и происходит адаптивное сжатие, и здесь же возникают основные потери качества фильма. Квантование - это целочисленное поэлементное деление матрицы амплитуд на матрицу квантования (МК). Подбор значений МК позволяет увеличивать или уменьшать потери по определенным частотам и регулировать качество изображения и степень сжатия. Заметим, что для различных компонентов изображения могут быть свои МК.

Следующий шаг алгоритма заключается в преобразовании полученной матрицы 8×8 в вектор из 64 элементов. Этот этап называется зигзаг-сканированием, т.к. элементы из матрицы выбираются, начиная с левого верхнего, зигзагом по диагоналям, параллельным побочной диагонали. В результате получается вектор, в начальных позициях которого находятся элементы матрицы, соответствующие низким частотам, а в конечных - высоким. Следовательно, в конце вектора будет очень много нулевых элементов.

Далее повторяются все действия, соответствующие стандартному алгоритму сжатия неподвижных изображений ***JPEG***.

Использование векторов смещений блоков. Простейшим способом учета подобия соседних кадров было бы вычитание каждого блока текущего кадра из каждого блока предыдущего. Однако гораздо более эффективным является алгоритм поиска векторов, на которые сдвинулись блоки текущего кадра по отношению к предыдущему.

Алгоритм состоит в том, что для каждого блока изображения мы находим блок, близкий к нему в некоторой метрике (например, по минимуму суммы квадратов разностей пикселов), в предыдущем кадре в некоторой окрестности текущего положения блока. Если минимальное расстояние между блоками в этой метрике меньше некоторого порога, то вместе с каждым блоком в выходном потоке сохраняется вектор смещения - координаты смещения максимально похожего блока в предыдущем ***I*** или ***P- кадре***. Если различия больше этого порога, блок сжимается независимо.

8.5. Методы сжатия речевых сигналов

Что бы ни говорили, а основным способом общения и связи между людьми были и остаются речь и передача речевых сообщений. Основные объемы передаваемой в системах связи информации сегодня приходится на речь – это и проводная телефония, и системы сотовой и спутниковой связи, и т.д. Поэтому эффективному кодированию, или сжатию речи, в системах связи уделяется исключительное внимание.

История сжатия речевых сигналов в системах связи насчитывает уже не один десяток лет. Так, например, в те времена, когда время ожидания заказанного телефонного разговора составляло десятки часов, экономические ограничения привели к установке на трансконтинентальных линиях США и атлантическом кабеле так называемой аппаратуры J2, каналы которой имели полосу 0,3 - 1,7 кГц при необходимой для нормального качества связи полосе 0,3 – 3,5 кГц. Такая аппаратура некогда работала и на линии Москва -Владивосток. Качество ее каналов едва достигало двух баллов MOS, но решающим оказалось двукратное увеличение числа телефонных соединений. Потребности пользователей в каналах сделали тогда вопросы качества речи второстепенными. Сегодня же фактор качества является не менее важным, чем экономия пропускной способности каналов связи.

Рассмотрим основные свойства речевого сигнала как объекта экономного кодирования и передачи по каналам связи и попытаемся пояснить, на каких свойствах сигнала основывается возможность его сжатия.

Речь представляет собой колебания сложной формы, зависящей от произносимых слов, тембра голоса, интонации, пола и возраста говорящего. Спектр речи весьма широк (примерно от 50 до 10000 Гц), но для передачи речи в аналоговой телефонии когда-то отказались от составляющих, лежащих за пределами полосы 0,3 - 3,4 кГц, что несколько ухудшило восприятие ряда звуков (например шипящих, существенная часть энергии которых сосредоточена в верхней части речевого спектра), но мало затронуло разборчивость. Ограничение частоты снизу (до 300 Гц) также немного ухудшает восприятие из-за потерь низкочастотных гармоник основного тона.

На приведенных ниже рисунках изображены фрагменты речевых сигналов, содержащих *гласные* (рис. 8.10) и *согласные* (рис. 8.11) звуки, а также спектры этих сигналов (рис. 8.12 и 8.13). Хорошо видны разница в характере соответствующих сигналов, а также то, что как в первом, так и во втором случаях ширина спектра сигнала не превышает 3,5 кГц. Кроме этого, можно отметить, что уровень низкочастотных (то есть медленных по времени) составляющих в спектре речевого сигнала значительно выше уровня высокочастотных (быстрых) составляющих. Эта существенная неравномерность спектра, кстати, является одним из факторов сжимаемости таких сигналов.

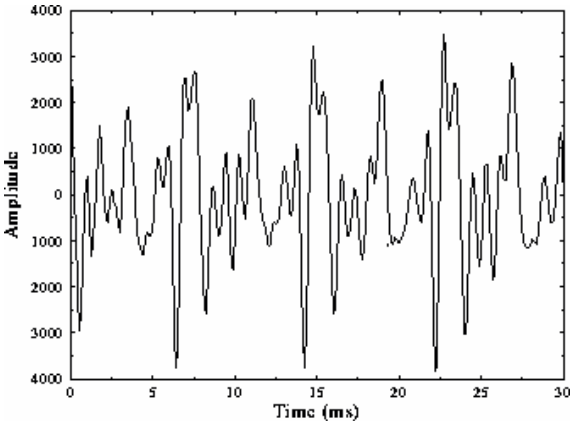


Рис. 8.10. Речевой сигнал, содержащий гласные звуки

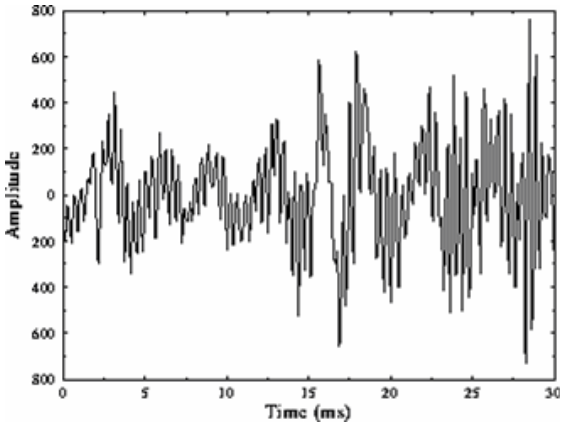


Рис. 8.11. Речевой сигнал, содержащий согласные звуки

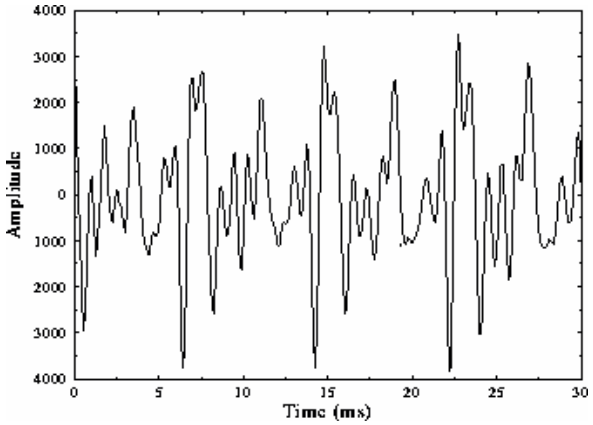


Рис. 8.12. Спектр сигнала, содержащий гласные звуки

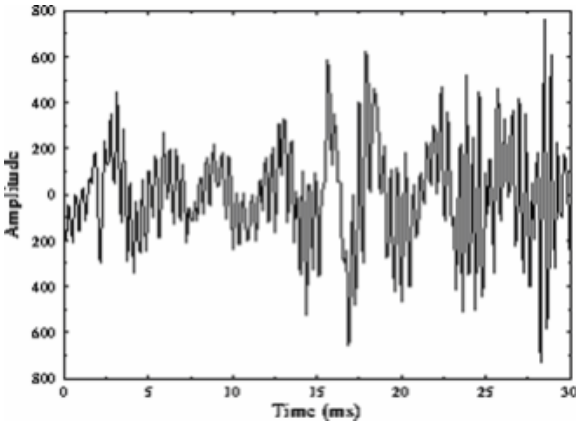


Рис. 8.13. Спектр сигнала, содержащий согласные звуки

Второй особенностью речевых сигналов, как это можно отметить из приведенных примеров, является неравномерность распределения вероятностей (плотности вероятности) мгновенных значений сигнала. Малые уровни сигнала значительно более вероятны, чем большие. Особенно это заметно на фрагментах большой длительности с невысокой активностью речи. Этот фактор, как известно, также обеспечивает возможность экономного кодирования – более вероятные значения могут кодироваться короткими кодами, менее вероятные – длинными.

Еще одна особенность речевых сигналов – их существенная нестационарность во времени: свойства и параметры сигнала на различных участках значительно различаются. При этом размер интервала стационарности составляет порядка нескольких десятков миллисекунд. Это свойство сигнала значительно затрудняет его экономное кодирование и заставляет делать системы сжатия адаптивными, то есть подстраивающимися под значения параметров сигнала на каждом из участков.

Наконец, исключительно важным для организации сжатия речевых сигналов является понимание физики механизма речеобразования. Его упрощенная схема приведена н рис. 8.14.

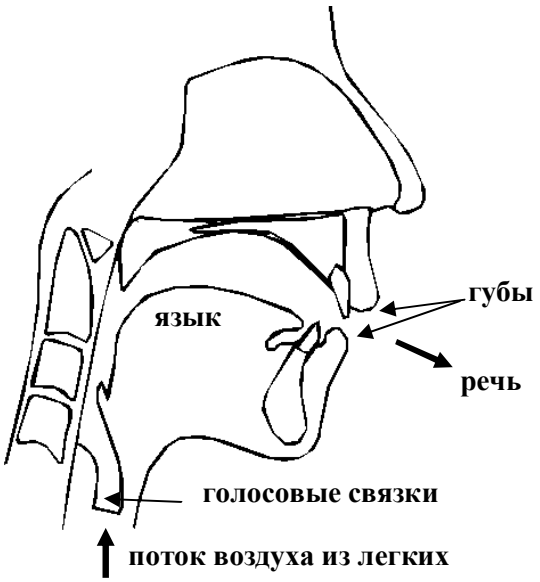


Рис. 8.14. Схема поясняющая физику механизма речеобразования

Речь формируется при прохождении выталкиваемого легкими потока воздуха через голосовые связки и голосовой тракт. Голосовой тракт начинается от голосовых связок и заканчивается губами и в среднем имеет длину порядка 15 - 17 сантиметров. Голосовой тракт в силу своих резонансных свойств вносит в формируемый сигнал набор характерных для каждого человека частотных составляющих, называемых *формантами*. Частоты и полосы этих формант могут управляться изменением формы голосового тракта, например, изменением положения языка.

Важной частью многих голосовых кодеров/декодеров является моделирование голосового тракта как кратковременного фильтра с изменяемыми параметрами. Поскольку форма голосового тракта может изменяться сравнительно медленно (трудно предположить, что можно изменять положение языка чаще, чем 20 – 30 раз в секунду), то параметры такого фильтра должны обновляться (или изменяться) также сравнительно редко (обычно – через каждые 20 миллисекунд или даже реже).

Таким образом, голосовой тракт возбуждается потоком воздуха, направляемым в него через голосовые связки. В зависимости от способа возбуждения возникающие при этом звуки можно разделить на три класса

Гласные звуки, возникающие, когда голосовые связки вибрируют, открываясь и закрываясь, прерывая тем самым поток воздуха от легких к голосовому тракту. Возбуждение голосового тракта при этом производится квазипериодическими импульсами. Скорость (частота) открывания и закрывания связок

определяют высоту возникающего звука (тона). Она может управляться *изменением формы и напряжения голосовых связок, а также изменением давления подводимого воздушного потока.*

Гласные звуки имеют высокую степень периодичности основного тона с периодом 2 - 20 мс. Эта долговременная периодичность хорошо видна на рис. 8.11, где приведен фрагмент речевого сигнала с гласным звуком.

Согласные звуки, возникающие при возбуждении голосового тракта шумоподобным турбулентным потоком, формируемым проходящим с высокой скоростью через открытые голосовые связки потоком воздуха. В таких звуках, как это видно из рис. 8.11, практически отсутствует долговременная периодичность, обусловленная вибрацией голосовых связок, однако кратковременная корреляция, обусловленная влиянием голосового тракта, имеет место.

Звуки взрывного характера, возникающие, когда закрытый голосовой тракт с избыточным давлением воздуха внезапно открывается.

Некоторые звуки в чистом виде не подходят ни под один из описанных выше классов, но могут рассматриваться как их смесь. *Таким образом, процесс речеобразования можно рассматривать как фильтрацию речеобразующим трактом с изменяющимися во времени параметрами сигналов возбуждения, также с изменяющимися характеристиками* (рис. 8.15).

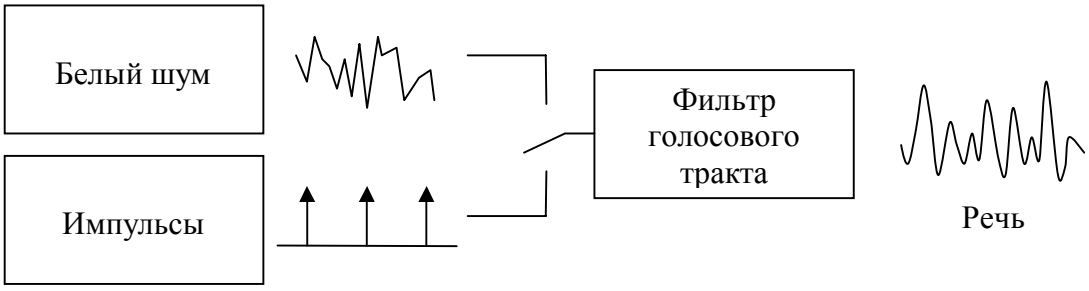


Рис. 8.15. Схема, поясняющая процесс речеобраования как фильтрацию сигналов возбуждения

При этом, несмотря на исключительное разнообразие генерируемых речевых сигналов, *форма и параметры голосового тракта, а также способы и параметры возбуждения достаточно однообразны и изменяются сравнительно медленно.* Из рис. 8.10 и 8.11 хорошо видно, что речевой сигнал обладает высокой степенью кратковременной и долговременной предсказуемости из-за периодичности вибраций голосовых связок и резонансных свойств голосового тракта. *Большинство кодеров/декодеров речи и используют эту предсказуемость, а также медленность изменения параметров модели системы речеобразования для уменьшения скорости кода.* При этом все известные способы экономного кодирования речевых сигналов можно условно разделить на три класса, описанные ниже.

8.5.1. Кодирование формы сигнала

Простейшими кодерами/декодерами речи, вообще не использующими информацию о том, как был сформирован кодируемый сигнал, а просто стараю-

щимися максимально приблизить восстанавливаемый сигнал по форме к оригиналу, являются *кодеры/декодеры формы сигнала*. Теоретически они инвариантны к характеру сигнала, подаваемого на их вход, и могут использоваться для кодирования любых, в том числе и неречевых, сигналов. Эти кодеры - самые простые по принципу действия и устройству, но больших степеней сжатия (низких скоростей кода) обеспечить не могут.

Простейшим способом кодирования формы сигнала является так называемая *импульсно-кодовая модуляция* – ***ИКМ*** (или ***PCM*** – Pulse Code Modulation), при использовании которой производятся просто дискретизация и равномерное квантование входного сигнала, а также преобразование полученного результата в равномерный двоичный код.

Для речевых сигналов со стандартной для передачи речи полосой 0,3 – 3,5 кГц обычно используют частоту дискретизации $\boldsymbol{F_{дискр}} \geq 2\boldsymbol{F_{max}} = 8$ кГц. Экспериментально показано, что при равномерном квантовании для получения практически идеального качества речи нужно квантовать сигнал не менее чем на ± 2000 уровней, иными словами, для представления каждого отсчета понадобится 12 бит, а результирующая скорость кода будет составлять

$$\boldsymbol{R} = 8000 \text{ отсчетов/с} * 12 \text{ бит/отсчет} = 96000 \text{ бит/с} = 96 \text{ кбит/с}.$$

Используя *неравномерное квантование* (более точное для малых уровней сигнала и более грубое для больших его уровней, таким образом, *чтобы относительная ошибка квантования была постоянной для всех уровней сигнала*), можно достичь того же самого субъективного качества восстановления речевого сигнала, но при гораздо меньшем числе уровней квантования – порядка ± 128 . В этом случае для двоичного представления отсчетов сигнала понадобится уже 8 бит и результирующая скорость кода составит 64 кбит/с.

С учетом статистических свойств речевого сигнала (вида распределения вероятностей мгновенных значений), а также нелинейных свойств слуха, гораздо лучше различающего слабые звуки, оптимальной является логарифмическая шкала квантования, которая и была принята в качестве стандарта еще в середине 60-х годов и сегодня повсеместно используется. Правда, в США и Европе стандарты нелинейного квантования несколько различаются (μ-law companding и A-law compression), что приводит к необходимости перекодирования сигналов.

Таким образом, исходной для любого сравнения эффективности и качества кодирования речевых сигналов может служить скорость кода, равная 64 кбит/с.

Следующим приемом, позволяющим уменьшить результирующую скорость кода, может быть попытка предсказать значение текущего отсчета сигнала по нескольким предыдущим его значениям, и далее, кодирование уже не самого отсчета, а ошибки его предсказания – *разницы между истинным значением текущего отсчета и его предсказанным значением*. Если точность предсказания достаточно высока, то ошибка предсказания очередного отсчета будет значительно меньше величины самого отсчета и для ее кодирования понадо-

бится гораздо меньшее число бит. Таким образом, *чем более предсказуемым будет поведение кодируемого сигнала, тем более эффективным будет его сжатие.*

Описанная идея лежит в основе так называемой *дифференциальной импульсно-кодовой модуляции* - ***ДИКМ*** (***DPCM***) – способа кодирования, при котором кодируются не сами значения сигнала, а их *отличия от некоторым образом предсказанных значений.* Простейшим способом предсказания является *использование предыдущего отсчета сигнала в качестве предсказания его текущего значения*:

$$x^*_i = x_{i-1}, \qquad \varepsilon_i = x_i - x^*_1. \tag{8.10}$$

Это так называемое *предсказание нулевого порядка*, самое простое, но и наименее точное. Более точным, очевидно, будет предсказание текущего отсчета на основе линейной комбинации двух предшествующих и т.д.:

$$x^*_i = \sum a_k x_{i-k}, \quad \varepsilon_i = x_i - x^*_1. \tag{8.11}$$

К сожалению, точность предсказания не всегда растет с ростом порядка предсказания, поскольку свойства сигнала между отсчетами начинают уже изменяться, поэтому обычно ограничиваются предсказанием не выше 2 – 3-го порядка.

На рис. 8.16 и 8.17 приведены схемы ***ДИКМ*** кодера и декодера.

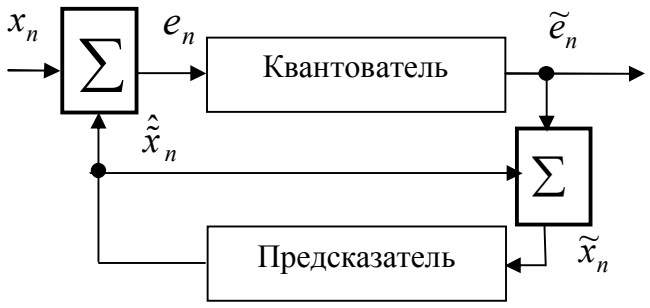


Рис. 8.16.Кодер ДИКМ

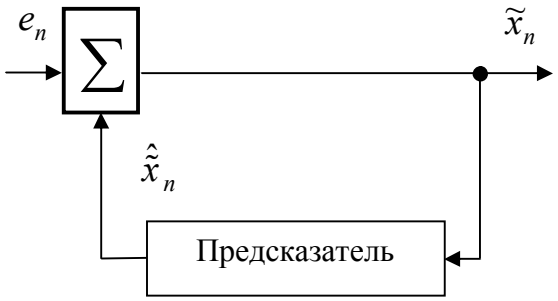


Рис. 8.17. Декодер ДИКМ

При кодировании речевых сигналов с учетом степени их кратковременной (на несколько очередных отсчетов) предсказуемости результирующая скорость кода для ***ДИКМ*** (***DPCM***) обычно составляет 5 – 6 бит на отсчет или 40 – 48 кбит/с.

Эффективность ***ДИКМ*** может быть несколько повышена, если предсказание и квантование сигнала будет выполняться не на основе некоторых усредненных его характеристик, а с учетом их текущего значения и изменения во времени, то есть адаптивно. Так, если скорость изменения сигнала стала большей, можно увеличить шаг квантования, и, наоборот, если сигнал стал изменяться гораздо медленнее, величину шага квантования можно уменьшить. При этом ошибка предсказания уменьшится и, следовательно, будет кодироваться

меньшим числом бит на отсчет. Такой способ кодирования называется *адаптивной* ***ДИКМ***, или ***АДИКМ*** (***ADPCM***). Сегодня такой способ кодирования стандартизован и широко используется при сжатии речи в междугородных цифровых системах связи, в системе микросотовой связи ***DECT***, в цифровых бесшнуровых телефонах и т.д. Использование ***АДИКМ*** со скоростью кода 4 бита/отсчет или 32 кбит/с обеспечивает такое же субъективное качество речи, что и 64 кбит/с - ***ИКМ***, но при вдвое меньшей скорости кода.

На сегодня стандартизованы также ***АДИКМ*** – кодеки для скоростей 40, 24 и 16 кбит/с (в последнем случае с несколько худшим, чем для 32 кбит/с – АДИКМ, качеством сигнала). Таким образом, видно, что сжатие речевых сигналов на основе кодирования их формы обеспечивает в лучшем случае двух - трехкратное уменьшение скорости кода. Дальнейшее снижение скорости ведет к резкому ухудшению качества кодируемого сигнала.

Описанные выше кодеры формы сигнала использовали чисто временной подход к описанию этого сигнала. Однако возможны и другие подходы. Примером может служить так называемое *кодирование поддиапазонов* (Sub-Band Coding - ***SBC***), при котором входной сигнал разбивается (или расфильтровывается) на несколько частотных диапазонов (поддиапазонов - sub-bands) и сигнал в каждом из этих поддиапазонов кодируется по отдельности, например, с использованием техники ***АДИКМ***.

Поскольку каждый из частотных поддиапазонов имеет более узкую полосу (все поддиапазоны в сумме дают полосу исходного сигнала), то и частота дискретизации в каждом поддиапазоне также будет меньше. В результате суммарная скорость всех кодов будет по крайней мере не больше, чем скорость кода для исходного сигнала. Однако у такой техники есть определенные преимущества. Дело в том, что субъективная чувствительность слуха к сигналам и их искажениям различна на разных частотах. Она максимальна на частотах 1 - 1,5 кГц и уменьшается на более низких и более высоких частотах. Таким образом, если в диапазоне более высокой чувствительности слуха квантовать сигнал более точно, а в диапазонах низкой чувствительности более грубо, то можно получить выигрыш в результирующей скорости кода. Действительно, при использовании технологии *кодирования поддиапазонов* получено хорошее качество кодируемой речи при скорости кода 16 – 32 кбит/с. Кодер получается несколько более сложным, чем при простой АДИКМ, однако гораздо проще, нежели для других эффективных способов сжатия речи.

Упрощенная схема подобного кодера (с разбиением на 2 поддиапазона) приведена на рис. 8.18.

Близким к кодированию поддиапазонов является метод сжатия, основанный на применении к сигналу линейных преобразований, к примеру, дискретного косинусного или синусного преобразования. Для кодирования речи используется так называемая технология ***ATC*** (Adaptive Transform Coding), при которой сигнал разбивается на блоки, к каждому блоку применяется дискретное косинусное преобразование и полученные коэффициенты адаптивно, в соответствии с характером спектра сигнала, квантуются.

Чем более значимыми являются коэффициенты преобразования, тем большим числом бит они кодируются. Техника очень похожа на ***JPEG***, но применяется к речевым сигналам. Достигаемые при таком кодировании скорости кодов составляют 12 – 16 кбит/с при вполне удовлетворительном качестве сигнала. Широкого распространения для сжатия речи этот метод не получил, поскольку известны гораздо более эффективные и простые в исполнении методы кодирования.

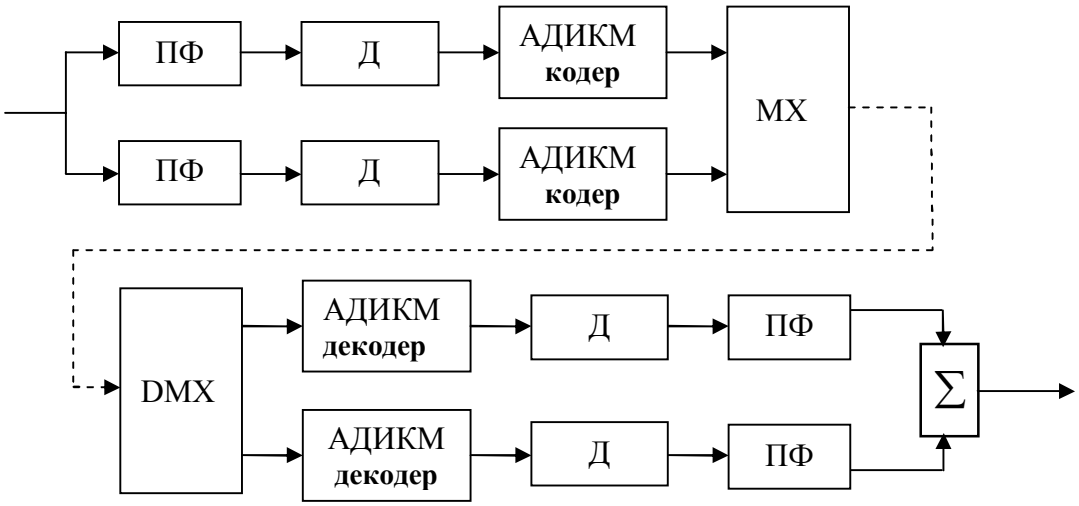


Рис. 8.18. Схема, поясняющая кодирование поддиапазонов

Следующим большим классом кодеров речевых сигналов являются кодеры источника.

8.5.2. Кодирование источника

В отличие от *кодеров формы сигнала*, вообще не использующих информацию о том, как был сформирован кодируемый сигнал, *кодеры источника* основываются *именно на модели источника* и из кодируемого сигнала извлекают информацию о параметрах этой модели. При этом *результатом кодирования являются не коды сигналов, а коды параметров источника этих сигналов.* Кодеры источника для кодирования речи называются *вокодерами* (VOice CODERS) и работают примерно следующим образом. Голосообразующий тракт представляется как линейный фильтр с переменными во времени параметрами, возбуждаемый либо источником белого шума (при формировании согласных звуков), либо последовательностями импульсов с периодом основного тона (при формировании гласных звуков) – рис. 8.19 .

Линейная модель системы речеобразования и ее параметры могут быть найдены различными способами. И от того, каким способом они определяются, зависит тип вокодера.

Информация, которую получает вокодер в результате анализа речевого сигнала и передает декодеру, это *параметры речеобразующего фильтра, указатель гласный/негласный звук, мощность сигнала возбуждения и период ос-*

новного тона для гласных звуков. Эти параметры должны обновляться каждые 10 – 20 мс, чтобы отслеживать нестационарность речевого сигнала.

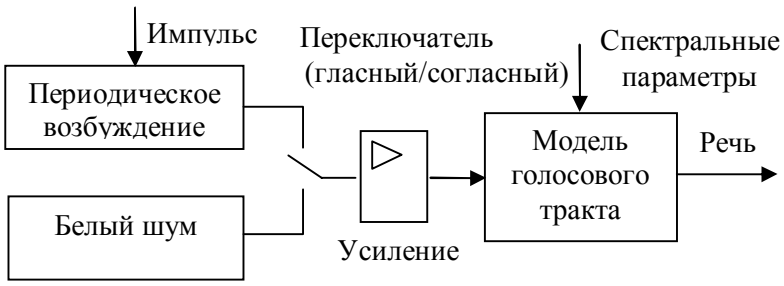


Рис. 8.19. Представление голосообразующего тракта линейным фильтром с перемещением во времени параметрами

Вокодер, в отличие от кодера формы сигнала, пытается сформировать сигнал, звучащий как оригинальная речь, и не обращает внимания на отличие формы этого сигнала от исходного. При этом *результирующая скорость кода на его выходе обычно составляет не более* 2,4 кбит/с, то есть *в пятнадцать раз меньше, чем при* ***АДИКМ***! К сожалению, качество речи, обеспечиваемой вокодерами, очень далеко от идеального, ее звучание хотя и достаточно разборчиво, но абсолютно ненатурально. При этом даже существенное увеличение скорости кода практически не улучшает качества речи, поскольку для кодирования была выбрана слишком простая модель системы речеобразования. Особенно грубым является предположение о том, что речь состоит лишь из гласных и согласных звуков, не допускающее каких либо промежуточных состояний.

Основное применение вокодеры нашли в военной области, где главное – это не натуральность речи, а большая степень ее сжатия и очень низкая скорость кода, позволяющая эффективно защищать от перехвата и засекречивать передаваемую речь. Кратко рассмотрим основные из известных типов вокодеров.

Канальные вокодеры. Это наиболее древний тип вокодера, предложенный еще в 1939 году. Этот вокодер использует слабую чувствительность слуха человека к незначительным фазовым (временным) сдвигам сигнала.

Для сегментов речи длиной примерно в 20 - 30 мс с помощью набора узкополосных фильтров определяется амплитудный спектр. Чем больше фильтров, тем лучше оценивается спектр, но тем больше нужно бит для его кодирования и тем больше результирующая скорость кода. Сигналы с выходов фильтров детектируются, пропускаются через ФНЧ, дискретизуются и подвергаются двоичному кодированию (рис. 8.20).

Таким образом, определяются медленно изменяющиеся параметры голосообразующего тракта и, кроме того, с помощью детекторов основного тона и гласных звуков, – период основного тона возбуждения и признак - гласный/негласный звук.

Канальный вокодер может быть реализован как в цифровой, так и в аналоговой форме и обеспечивает достаточно разборчивую речь при скорости кода

на его выходе порядка 2,4 кбит/с.

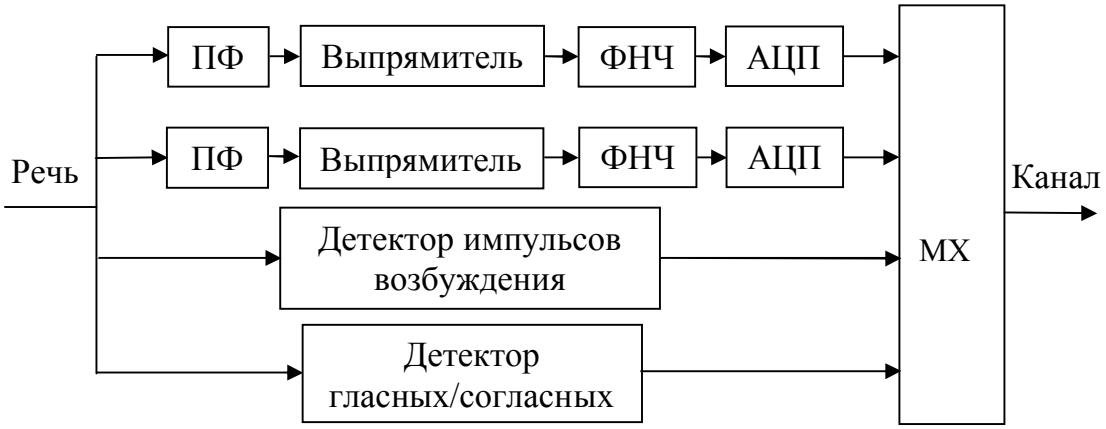


Рис. 8.20. Схема начального вокодера

Декодер (рис. 8.21), получив информацию, вырабатываемую кодером, обрабатывает ее в обратном порядке, синтезируя на своем выходе речевой сигнал, в какой-то мере похожий на исходный.

Учитывая простоту модели, трудно ожидать от вокодерного сжатия хорошего качества восстановленной речи. Действительно, канальные вокодеры используются в основном только там, где главным образом необходимы разборчивость и высокая степень сжатия: в военной связи, авиации, космической связи и т.д.

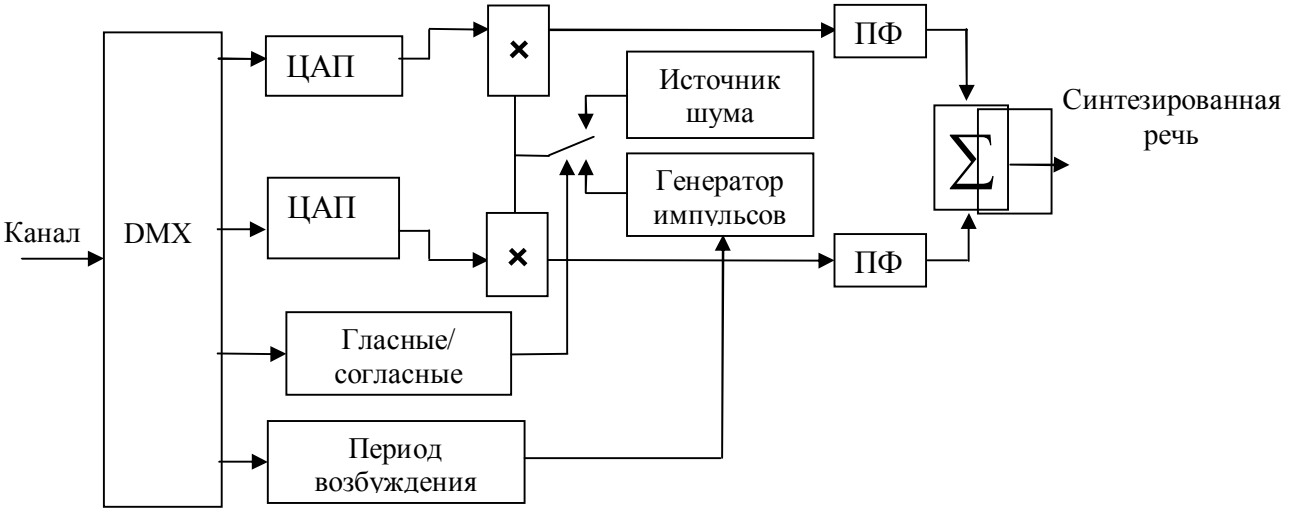


Рис. 8.21. Декодер сжатой речи

Гомоморфный вокодер. Гомоморфная обработка сигналов представляет собой один из нелинейных методов обработки, который может эффективно применяться к сложным сигналам, например к речевым.

С учетом используемой в вокодерах модели системы голосообразования речевой сигнал можно представить как временную свертку импульсной переходной характеристики голосового тракта с сигналом возбуждения. В частотной

области это соответствует произведению частотной характеристики голосового тракта и спектра сигнала возбуждения. Наконец, если взять логарифм от этого произведения, то получим сумму логарифмов спектра сигнала возбуждения и частотной характеристики голосового тракта. Поскольку человеческое ухо практически не чувствительно к фазе сигнала, можно оперировать с амплитудными спектрами:

$$\boldsymbol{log(|S(e^{jw})|) = log(|P(e^{jw})|) + log(|V(e^{jw})|,} \qquad (8.12)$$

где $\boldsymbol{S(e^{jw})}$ - спектр речи, ***P(ejw)*** спектр сигнала возбуждения и ***V(ejw) -*** частотная характеристика голосового тракта.

Если теперь выполнить над $\boldsymbol{log(|S(e^{jw})|)}$ обратное преобразование Фурье (***ОПФ***), то получим так называемый *кепстр сигнала.* Параметры голосового тракта изменяются во времени сравнительно медленно (их спектр находится в области низких частот - НЧ), тогда как сигнал возбуждения – быстроосциллирующая функция (ее спектр сосредоточен в области высоких частот - ВЧ). Поэтому *в кепстре речевого сигнала* эти составляющие разделяются (рис. 8.22) и могут быть закодированы по отдельности.

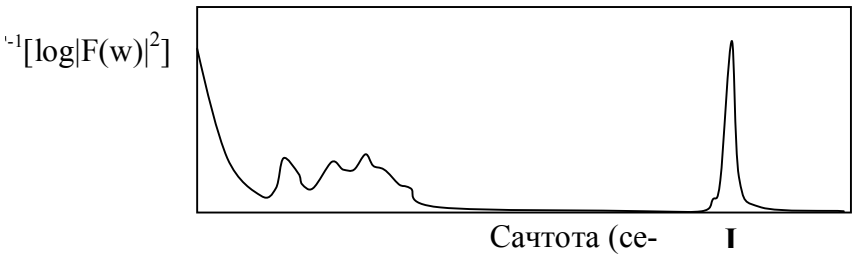


Рис. 8.22. Представление речевого сигнала в виде НЧ и ВЧ составляющих

Схема гомоморфного кодера/декодера речи приведена на рис. 8.23, с его использованием можно получить скорость кода порядка 4 кбит/с.

Формантные вокодеры. Как уже отмечалось ранее, основная информация о речевом сигнале содержится в положении и ширине составляющих его формант. Если с высокой точностью определять и кодировать параметры этих формант, можно получить очень низкую результирующую скорость кода – менее 1 кбит/с. К сожалению, сделать это очень трудно, поэтому формантные кодеры речи пока не нашли широкого распространения.

Вокодеры с линейным предсказанием. Вокодеры на основе *линейного предсказания* используют такую же модель речеобразования, что и остальные из рассмотренных. Что их отличает – это метод определения параметров тракта. Линейные предсказывающие кодеры, или ЛПК, полагают голосовой тракт линейным фильтром с непрерывной импульсной переходной характеристикой, в котором каждое очередное значение сигнала может быть получено как линейная комбинация некоторого числа его предыдущих значений.

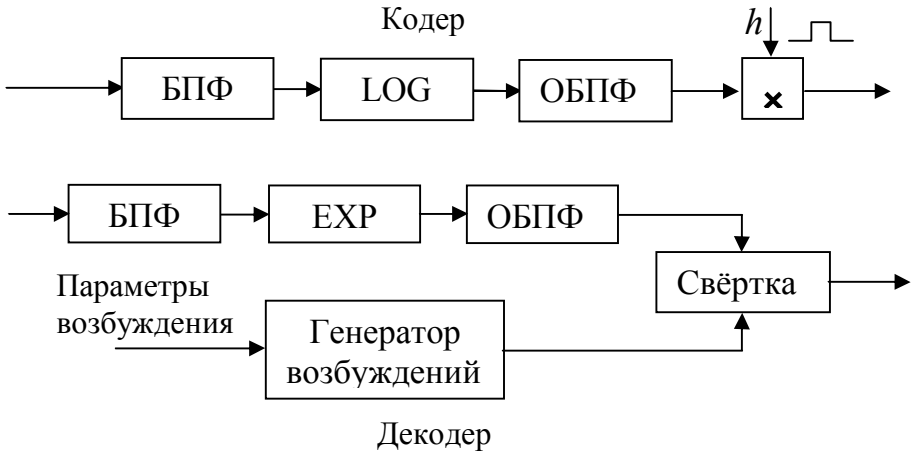


Рис. 8.23. Схема гомоморфного кодера/декодера

В ЛПК-вокодере речевой сигнал делится на блоки длиной около 20 мс, для каждого из которых определяются коэффициенты предсказывающего фильтра. Эти коэффициенты квантуются и передаются декодеру. Затем речевой сигнал пропускается через фильтр, частотная характеристика которого обратна частотной характеристике голосового тракта. На выходе фильтра получается ошибка предсказания. Назначение предсказателя – устранить корреляцию между соседними отсчетами сигнала. В результате гораздо отчетливее проявляется долговременная корреляция в сигнале, что позволяет точнее определить частоту основного тона и выделить признак гласный/согласный звук.

Вокодеры на основе линейного предсказания сейчас наиболее популярны, поскольку все используемые ими фильтровые модели речевого тракта работают очень хорошо. Получаемые с их помощью скорости кодов при неплохом качестве речи составляют до 2,4 кбит/с.

8.5.3. Гибридные методы кодирования речи

Гибридные, или комбинированные, методы кодирования речи заполняют разрыв между *кодерами формы сигнала*, совершенно не учитывающими его природы, и *кодерами источника*, кодирующими, по сути, не сигнал, а параметры модели порождающего его источника. Как отмечалось ранее, кодеры формы сигнала обеспечивают очень хорошее качество речи при скоростях кодирования выше 16 кбит/с, но вообще не работают при более низких скоростях, тогда как вокодеры обеспечивают разборчивую речь при скоростях кодирования 2,4 кбит/с и ниже, но не могут дать хорошего качества при любой скорости кода.

Наиболее распространенными в настоящее время являются гибридные методы кодирования, работающие во временной области (то есть с сигналом, а не его спектром или другими линейными преобразованиями), основанные на анализе сигнала через его синтез (так называемые ***ABS***-кодеки). Эти кодеры так же, как и вокодеры, используют модель голосового тракта, но несколько иным образом – для *подбора сигнала возбуждения, обеспечивающего наилучшее совпадение синтезированного на ее основе речевого сигнала с исходным.*

ABS-кодеры были впервые предложены сравнительно недавно – в 1982 году - и в своем первоначальном виде получили название ***MPE***-кодеров (Multi-Pulse Excited - кодеры с многоимпульсным возбуждением). Позднее были предложены более совершенные ***RPE***-кодеры (Regular-Pulse Excited – кодеры с регулярным импульсным возбуждением) и ***CELP***-кодеры (Codebook-Excited Linear Predictive – с возбуждением на основе кодовых книг). Сегодня существуют и другие их разновидности, но все они используют общую идею.

Чтобы понять, на чем основаны эффективность и качество ***ABS***-кодера, сначала рассмотрим работу так называемого ***RELP***-кодера (Residual Excited Linear Prediction - RELP).

Если речевой сигнал (имеющий спектр рис. 8.24, а) пропустить через линейный предсказатель (с частотной характеристикой вида рис. 8.24, б), то корреляция между отсчетами выходного сигнала (*ошибки предсказания*) значительно уменьшится. Если предсказание выполнялось достаточно хорошо, то выходом предсказателя будет практически белый шум с равномерным спектром (рис. 8.24, в).

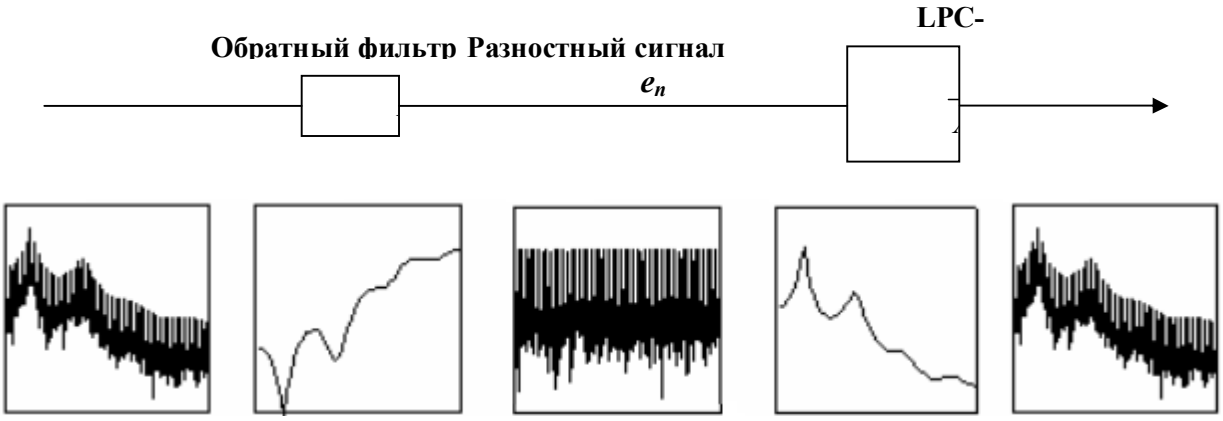


Рис. 8.24

Вместе с тем этот белый шум (*ошибка предсказания*) несет всю информацию о кодируемом речевом сигнале, и если его пропустить снова через LPC-фильтр (с частотной характеристикой - рис. 8.24,г), то мы абсолютно точно восстановим исходный речевой сигнал. Поскольку эта информация распределена по спектру ошибки предсказания более или менее равномерно, то возникла идея кодировать и передавать только небольшую часть спектра ошибки предсказания ***E(ω)***, а остальное восстанавливать в декодере.

В ***RELP***-кодере сигнал ошибки предсказания пропускается через низкочастотный фильтр с частотой среза около 1 кГц. Сигнал с выхода фильтра кодируется по форме, например ДИКМ-кодером. В декодере ошибка предсказания восстанавливается путем ее переноса в область удаленных низкочастотным фильтром кодера частот.

RELP-кодер работал бы идеально, если бы в процессе линейного предсказания мы получали белый шум. Однако из за наличия в речевом сигнале квазипериодических формантных составляющих линейный предсказатель не может устранить долговременной корреляции с периодом основного тона формант и они будут явно присутствовать в спектре ошибки предсказания. Если теперь пропустить ***E(ω)*** через ФНЧ, то высокочастотные формантные составляющие будут утеряны и в дальнейшем не смогут быть восстановлены.

RELP-кодеры позволяют получить неплохое качество сигнала при скорости кода порядка 9.6 кбит/с, однако им в некоторой степени присущ недостаток вокодеров – синтетический характер восстановленной речи. В связи с этим на смену им практически повсеместно пришли похожие по принципу работы ***ABS***-кодеры в их разновидностях.

ABS-кодер работает следующим образом. Кодируемый входной сигнал (уже в цифровой форме, в виде потока отсчетов) разбивается на фрагменты длиной порядка 20 мс, в пределах которых свойства сигнала изменяются незначительно. Для каждого из этих фрагментов определяются текущие параметры синтезирующего фильтра (аналога голосового тракта) и далее подбирается сигнал возбуждения, который, будучи пропущенным через синтезирующий фильтр, минимизирует ошибку между входным и синтезированным сигналами.

Таким образом, название метода Analysis-by-Synthesis состоит в том, что кодер анализирует входную речь посредством синтеза множества приближений к ней. В конечном итоге кодер передает декодеру информацию, представляющую собой комбинацию текущих параметров синтезирующего фильтра и сигнала возбуждения. Желательно, чтобы этих данных было поменьше. Декодер по этим параметрам восстанавливает закодированную речь, причем делает это так же, как это делал кодер в процессе анализа через синтез. Различие между ***ABS***-кодерами разного типа состоит в том, *как в каждом из них подбирается сигнал возбуждения синтезирующего фильтра* ***u(n).*** Теоретически на вход синтезирующего фильтра нужно подать бесконечно большое число различных сигналов возбуждения, чтобы посмотреть, какой сигнал получится на его выходе, и сравнить его с кодируемым. Сигнал возбуждения, который даст минимум взвешенной ошибки между оригиналом и синтезированной речью, выбирается в качестве результата кодирования. Именно эта замкнутая схема определения сигнала возбуждения (рис. 8.25) и обеспечивает ***ABS***-кодерам высокое качество кодируемой речи при низких скоростях кода.

Проблема состоит в большом количестве вычислительных операций, необходимых для подбора наилучшего сигнала возбуждения. Но для сегодняшних возможностей вычислительной и микропроцессорной техники это вполне разрешимая задача.

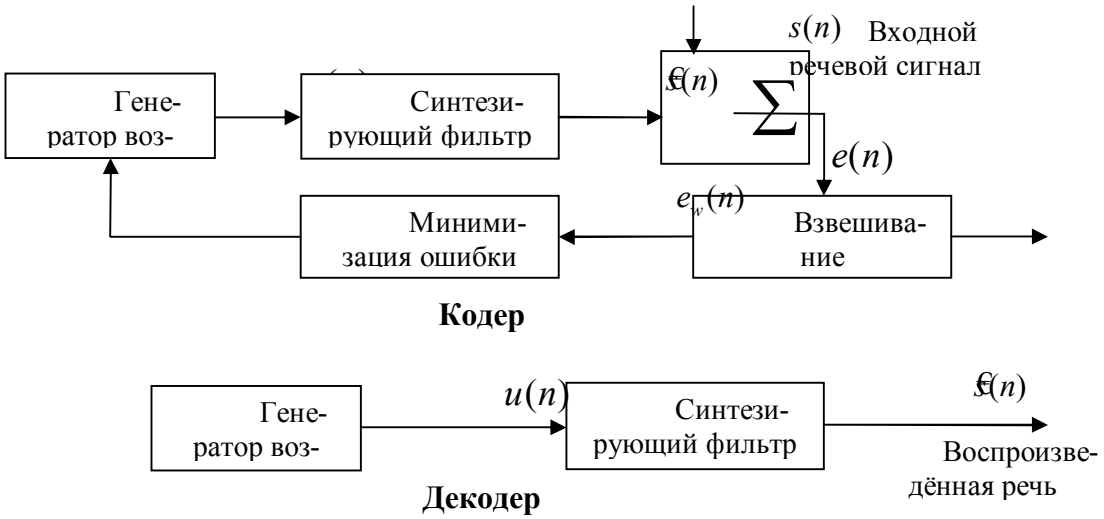


Рис. 8.25. Кодер и декодер гибридного метода кодирования речи

Многоимпульсные кодеры (MPE-кодеры). Как уже говорилось, при прохождении речевого сигнала через предсказывающий фильтр корреляция между его соседними отсчетами значительно уменьшается. Однако для гласных звуков наличие формантных составляющих приводит к появлению в *речевом сигнале* квазипериодичности и высокой долговременной корреляции. Эта периодичность не устраняется линейным предсказанием и приводит к появлению в сигнале *ошибки предсказания* высокоамплитудных спайков. Чтобы устранить долговременную корреляцию, можно пропустить сигнал ошибки предсказания через второй линейный предсказатель. Этот линейный предсказатель должен устранить корреляцию уже *не между соседними отсчетами речевого сигнала*, а между *соседними периодами ошибки предсказания.* Это достигается введением в предсказатель временной задержки на величину периода основного тона речевого сигнала:

$$\boldsymbol{P(z) = 1 - \sum_i \beta_i z^{-M-i}}, \tag{8.13}$$

где $\boldsymbol{M}$ – период основного тона.

На приведенном ниже рис. 8.26 изображены: а - исходный речевой сигнал; б - сигнал ошибки кратковременного линейного предсказания (увеличенный в 3 раза); в - сигнал на выходе двухкаскадного (кратковременного + долговременного) предсказателя (увеличенный в 10 раз).

Если теперь подать результирующий сигнал ошибки предсказания в качестве возбуждения на последовательно соединенные кратковременный и долговременный фильтры-предсказатели, то на выходе получим исходный неискаженный речевой сигнал. Можно было бы кодировать и передавать по каналу связи полученный сигнал ошибки предсказания, и уже это обеспечивало бы определенную экономию из-за существенно меньшей его амплитуды по сравнению с исходным речевым сигналом. Однако форма сигнала (рис. 8.26,в) все

же остается довольно сложной, что требует для его кодирования достаточно много бит.

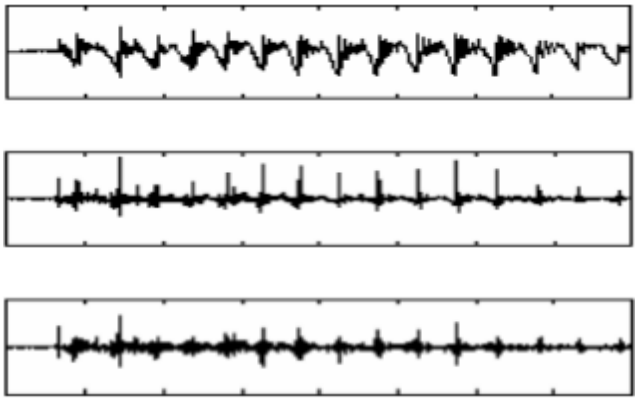


Рис. 8.26

В *многоимпульсных кодерах* (***MPE***) в качестве сигнала возбуждения ***u(n)*** берут *не ошибку предсказания* (рис. 8.26,в), а просто *последовательность из четырех - шести коротких импульсов*. Временное положение каждого из этих импульсов и их амплитуды определяются в процессе процедуры *анализа через синтез (**ABS**)* до достижения минимальных различий между исходным и синтезированным речевыми сигналами. Параметры импульсов возбуждения, минимизирующие ошибку, подбирают последовательно, сначала для первого импульса, затем для второго и т.д. На практике достаточно задавать положение импульсов с шагом около 1 мс и точностью амплитуд до 5 %, и это обеспечивает хорошее качество синтезируемого звука при скорости кода около 10 кбит/с. (Для фрагмента речевого сигнала длительностью в 20 мс используется 6 импульсов возбуждения, положение каждого задают с точностью 1мс = 1/20 от длительности фрагмента = 5 бит на импульс, амплитуду импульса - с точностью 5 % = =5 бит на импульс, в результате получим минимальную скорость кода *сигнала возбуждения* 6 × 10 = 60 бит/20 мс. Кроме этого, нужно будет добавить в код параметры фильтров долговременного и кратковременного предсказания для данного фрагмента, что составит примерно 80 – 100 бит/ 20мс, в результате получим скорость кода 160 бит/20 мс = 8 кбит/с.

Кодеры с регулярным импульсным возбуждением (RPE-кодеры). Так же как и ***MPE***-кодек, Regular Pulse Excited, или ***RPE***-кодек, использует в качестве сигнала возбуждения ***u(n)*** фиксированный набор коротких импульсов. Однако в этом кодеке импульсы расположены регулярно на одинаковых расстояниях друг от друга, и кодеру необходимо определить лишь положение первого импульса и амплитуды всех импульсов. Таким образом, декодеру нужно передавать меньше информации о положении импульсов, следовательно, в сигнал возбуждения можно включить их большее количество и тем самым улучшить приближение синтезированного сигнала к оригиналу. К примеру, если при скорости кода 10 кбит/с в ***MPE-***кодеке используется четырехимпульсный сиг-

нал возбуждения, то в ***RPE***-кодеке можно использовать уже десятиимпульсный сигнал. При этом существенно повышается качество речи. Метод регулярного импульсного возбуждения **RPE** сегодня широко применяется, в том числе в системе сотовой связи **GSM.** Кодеры с возбуждением на основе кодовых книг (CELP–кодеры). Методы кодирования **MPE** и **RPE** обеспечивают хорошее качество кодируемой речи при скоростях кода порядка 10 кбит/с и выше, но начинают сильно искажать сигнал при более низких скоростях. Дело в том, что для описания необходимых параметров сигнала возбуждения – временного положения и амплитуд импульсов - с требуемой точностью просто не хватает бит.

В связи с этим был предложен метод, использующий в качестве сигнала возбуждения не импульсные последовательности, *задаваемые набором своих параметров,* а *библиотеки (кодовые книги) специальным образом подготовленных и записанных в запоминающее устройство сигналов возбуждения различной формы* - Codebook Excited Linear Prediction (***CELP***).

Схема формирования сигнала возбуждения ***CELP***-кодера приведена на рис. 8.27.

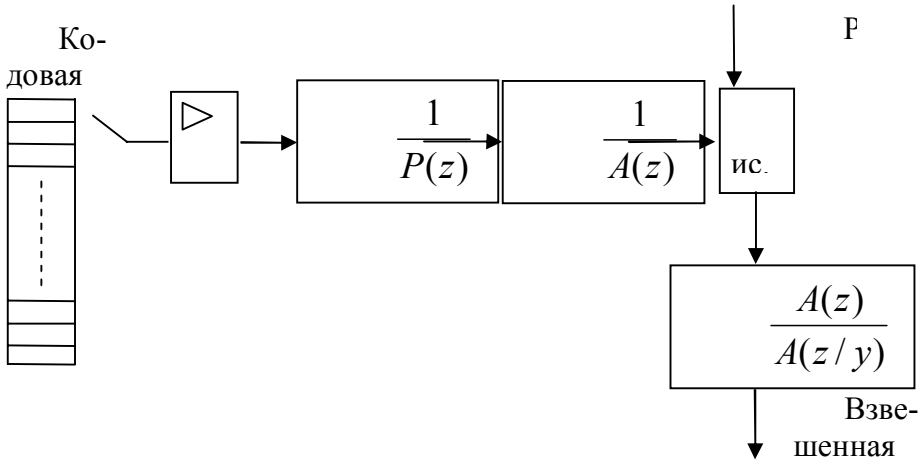


Рис. 8.27. Схема формирования сигнала возбуждения ***CELP***-кодера

Результатом кодирования при этом являются не параметры импульсов сигнала возбуждения, а индекс кодовой книги (номер хранимого в ней образца сигнала возбуждения), а также его амплитуда. Если кодовая книга содержит, к примеру, 1024 сигнала, а амплитуда сигнала кодируется с точностью 2 – 3 %, то необходимое число бит составит 10 (для индекса) + 5 (для амплитуды) = 15 бит на фрагмент сигнала длительностью в 20 мс (в сравнении с 47 битами, используемыми в ***GSM RPE-***кодеке). Правда, процедура кодирования требует очень больших вычислительных затрат, поэтому реализация ***CELP***-кодеров стала возможной только в последнее время с использованием специализированных сигнальных процессоров с производительностью порядка 300 млн. операций в секунду и более. Кодирование на основе алгоритма ***CELP*** с успехом используется в современных системах связи при скоростях кода от 16 до 4,8 кбит/с. При этом для скорости кода 16 кбит/с ***CELP*** обеспечивается такое же качество

речи, как и для 64 кбит/с ***ИКМ***, а при скорости кода 4,8 кбит/с - как для 13 кбит/с ***GSM RPE.***

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите типы систем сжатия.
2. Поясните принцип работы систем сжатия без потерь.
3. Назовите основные характеристики систем сжатия сообщений.
4. Поясните принцип работы систем сжатия с потерями и назовите их основные характеристики.
5. Поясните принцип кодирования повторов.
6. Поясните вероятностные методы сжатия.
7. Поясните идею арифметического кодирования на текстовой строке ТЕЛЕМЕХАНИКА.
8. Выполнить сжатия строки ИНФОРМАЦИЯ с помощью алгоритма LZW.
9. Поясните принцип дифференциального кодирования.
10. Поясните стандарт сжатия JPEG.
11. Поясните принцип фрактального сжатия.
12. В чём сущность волнового алгоритма сжатия.
13. Какие избыточности учитываются при сжатии подвижных изображений?
14. Назовите методы сжатия речевых сигналов.
15. Поясните принцип работы кодера/декодера формы сигнала.
16. На чём основывается принцип работы кодера источника?
17. В чём сущность гомоморфной обработки сигналов?
18. Поясните принцип гибридных методов кодирования речи.

9. КОДИРОВАНИЕ КАК СРЕДСТВО КРИПТОГРАФИЧЕСКОГО ЗАКРЫТИЯ ИНФОРМАЦИИ

В настоящее время все большее развитие получают вычислительные сети коллективного пользования. В таких системах концентрируются большие объемы данных, хранимые на машинных носителях, и осуществляется автоматический межмашинный обмен данными, в том числе на больших расстояниях.

Во многих случаях хранимая и передаваемая информация может представлять интерес для лиц, желающих использовать ее в корыстных целях. Последствия от такого несанкционированного использования информации могут быть весьма серьезными. Поэтому уже в настоящее время возникла проблема защиты информации от несанкционированного доступа [3]. В данном разделе ограничимся рассмотрением методов защиты информации от несанкционированного доступа при передаче ее по каналам связи. Рассматриваемые методы защиты обеспечивают такое преобразование сообщений, при котором их исходное содержание становится доступным лишь при наличии у получателя некоторой специфической информации (ключа) и осуществления с ее помощью

обратного преобразования. Эти методы называют методами криптографического закрытия информации. Они применяются как для защиты информации в каналах передачи, так и для защиты ее в каналах хранения.

Преобразования, выполняемые в системах, где используются методы криптографического закрытия информации, можно считать разновидностями процессов кодирования и декодирования, которые получили специфические названия шифрования и дешифрования. Зашифрованное сообщение называют криптограммой (шифртекстом).

Известно большое число различных методов криптографического закрытия информации. В настоящее время утвердились в практике следующие основные криптографические методы защиты: замены (подстановки); перестановки; использования генератора псевдослучайных чисел (гаммирование); перемешивания (алгоритмитические); использование систем с открытым ключом. Классификация методов преобразования информации приведена на рисунке 9.1. Рассмотрим некоторые из них в порядке возрастания сложности и надежности закрытия.

9.1. Метод замены

Шифрование методом замены (подстановки) основано на алгебраической операции, называемой подстановкой. В криптографии рассматриваются четыре типа подстановки: моноалфавитная, гоммофоническая, полиалфавитная и полиграммная. При моноалфавитной простой подстановке буквы кодируемого сообщения прямо заменяются другими буквами того же или другого алфавита. Если сообщения составляются из K различных букв, то существует $K!$ способов выражения сообщения K буквами этого алфавита, т.е. существует $K!$ различных ключей.

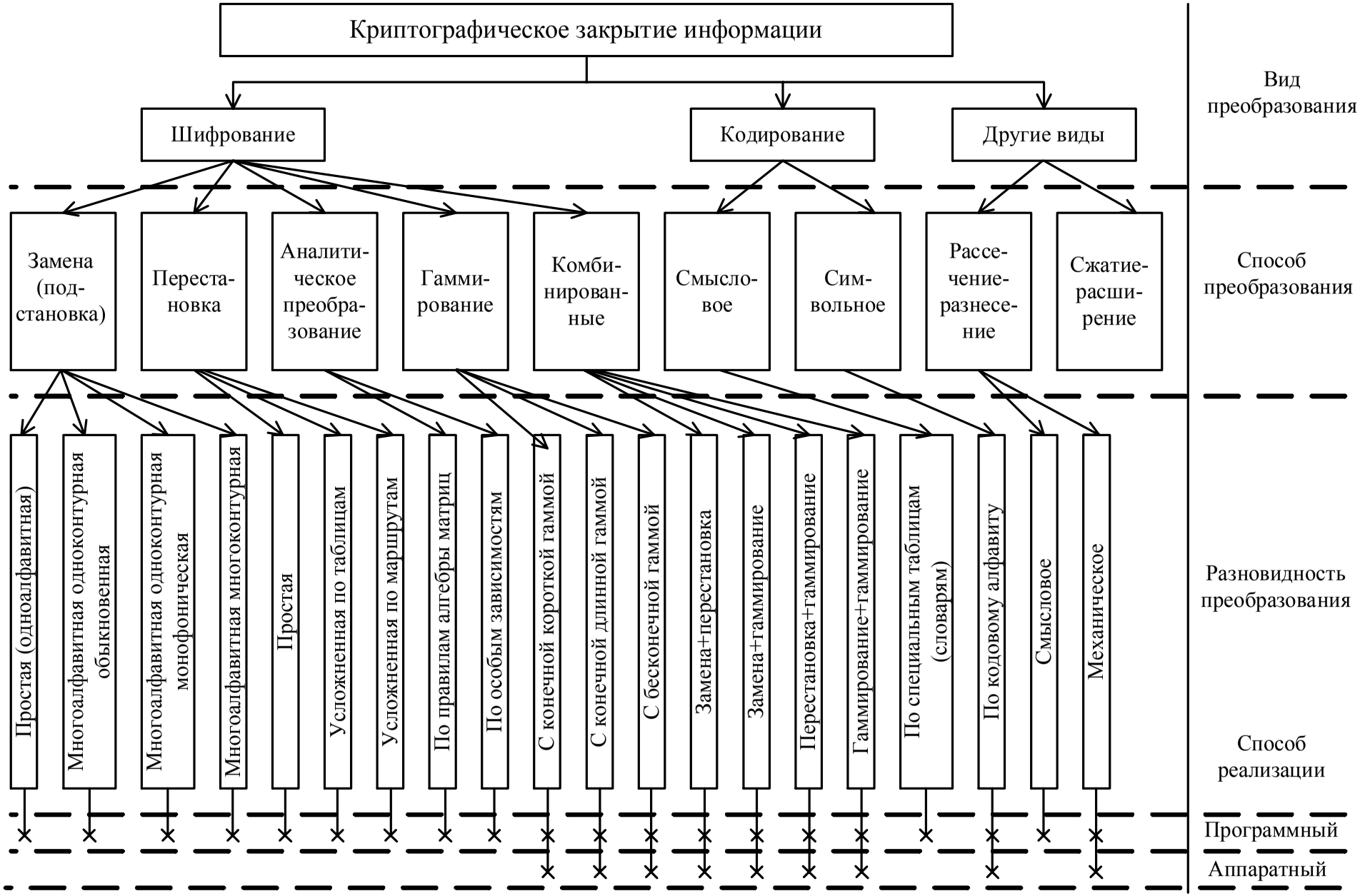

Криптографическое закрытие информации
Шифрование
Кодирование
Другие виды
Вид преобразования
Замена (под-становка)
Переста-новка
Аналити-ческое преобра-зование
Гамми-рование
Комби-нирован-ные
Смысло-вое
Сим-вольное
Рассе-чение-разнесе-ние
Сжатие-расши-рение
Способ преобразования
Простая (одноалфавитная)
Многоалфавитная одноконтурная обыкновенная
Многоалфавитная одноконтурная монофоническая
Многоалфавитная многоконтурная
Простая
Усложненная по таблицам
Усложненная по маршрутам
По правилам алгебры матриц
По особым зависимостям
С конечной короткой гаммой
С конечной длинной гаммой
С бесконечной гаммой
Замена+перестановка
Замена+гаммирование
Перестановка+гаммирование
Гаммирование+гаммирование
По специальным таблицам (словарям)
По кодовому алфавиту
Смысловое
Механическое
Разновидность преобразования
Способ реализации
Программный
Аппаратный

Рисунок 9.1 -

Пример 9.1. Зашифровать сообщение «ИНФОРМАЦИЯ», используя в качестве ключа для шифрования русского текста буквы русского алфавита в соответствии с таблицей 9.1.

Таблица 9.1 - Замена одних букв другими

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О
П	Р	С	Т	У	Ф	Х	Ч	Ц	Ы	Ъ	Ь	Э	Ю

П	Р	С	Т	У	Ф	Х	Ч	Ц	Ы	Ъ	Ь	Э	Ю	Я
Я	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О

Подставляя новые буквы, получаем криптограмму «ЦЭДЮАЬПЗЦО».

Метод шифрования прост, но не позволяет обеспечить высокой степени защиты информации. Это связано с тем, что буквы русского языка (как, впрочем, и других языков) имеют вполне определенные и различные вероятности появления(таблица 9.2 и 9.3).

Так как в зашифрованном тексте статистические свойства исходного сообщения сохраняются, то при наличии криптограммы достаточной длины можно с большой достоверностью определить вероятности отдельных букв, а по ним и буквы исходного сообщения.

Таблица 9.2 - Частота появления букв английского языка в тексте

Буква и частота её появления			
A 0.08167	H 0.06094	O 0.07507	V 0.00978
B 0.01492	I 0.06966	P 0.01929	W 0.0236
C 0.02782	J 0.00153	Q 0.00095	X 0.0015
D 0.04253	K 0.00772	R 0.05987	Y 0.01974
E 0.12702	L 0.04025	S 0.06327	Z 0.00074
F 0.0228	M 0.02406	T 0.09056	
G 0.02015	N 0.06749	U 0.02758	

Таблица 9.3 - Частота появления букв русского языка в тексте

Буква и частота её появления				
А 0.07821	Ж .01082	Н 0.0685	Ф 0.00132	Ы 0.0185
Б 0.01732	З 0.01647	О 0.11394	Х 0.00833	Ь 0.02106
В 0.04491	И 0.06777	П 0.02754	Ц 0.00333	Э 0.0031
Г 0.01698	Й 0.01041	Р 0.04234	Ч 0.01645	Ю 0.0054
Д 0.03103	К 0.03215	С 0.05382	Ш 0.0077	Я 0.01979
Е 0.08567	Л 0.04813	Т 0.06443	Щ 0.0033	
Ё 0.0007	М 0.0313	У 0.02882	Ъ 0.00023	

Ещё одна классическая система шифрования, изображенная на рисунке 9.2, называется квадратом Полибиуса (Polybius square).

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	IJ	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Рисунок 9.2 – Квадрат Полибиуса

Вначале объединяются буквы I и J и трактуются как один символ (в дешифрованном сообщении значение этой « двойной буквы» легко определяется из контекста). Получившиеся 25 символов алфавита размещаются в таблицу размером 5×5. Шифрование любой буквы производится с помощью выбора соответствующей пары числе – строки и столбца (или столбца и строки).

Пример 9.2. Зашифровать сообщение «now is the time» с помощью квадрата Полибиуса. Схема шифрования приведена в таблице 9.4.

Таблица 9.4 - Схема шифрования с помощью квадрата Полибиуса

Исходный текст:	N O W I S T H E T I M E
Криптограмма:	33 43 25 42 34 44 32 51 44 42 23 51

Код изменяется путем перестановки букв в таблице 5×5.

Шифр Виженера (Vigener key method) – этот шифр является одним из наиболее распространенных. Степень надежности закрытия информации повышается за счет того, что метод шифрования предусматривает нарушение статистических закономерностей появления букв алфавита.

Каждая буква алфавита нумеруется. Например, буквам русского алфавита ставятся в соответствие цифры от 0 до 33 (таблица 9.5).

Ключ представляет собой некоторое слово или просто последовательность букв, которая подписывается с повторением под сообщением. Цифровой эквивалент каждой буквы y_i криптограммы определяется в результате сложения с приведением по модулю 33 цифровых эквивалентов буквы сообщения x_i и лежащей под ней буквы ключа k_i, т.е. $y_i = x_i + k_i \,(\mathrm{mod}33)$.

Таблица 9.5 - Кодирование букв русского алфавита

Буква	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л
Цифра	01	02	03	04	05	06	07	08	09	10	11	12

Буква	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
Цифра	13	14	15	16	17	18	19	20	21	22	23	24

Продолжение таблицы 9.5

Буква	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	□ (ПРОБЕЛ)
Цифра	25	26	27	28	29	30	31	32	33

Пример 9.3. Зашифруем сообщение «ИНФОРМАЦИЯ» кодом Виженера с ключом «НЕМАН».

Запишем буквы сообщения, расположив под ними их цифровые эквиваленты. Аналогично внизу запишем ключ, повторяя его необходимое число раз:

И	Н	Ф	О	Р	М	А	Ц	И	Я
9	14	21	15	17	13	1	23	9	32
Н	Е	М	А	Н	Н	Е	М	А	Н
14	6	13	1	14	14	6	13	1	14

Складывая верхние и нижние цифровые эквиваленты с приведением по модулю 33, получим следующую последовательность чисел

23 20 1 16 31 27 7 3 10 13 , что соответствует криптограмме

«Ц У А П Ю Ъ Ж В Й М» .

Шифр Вижинера обладает достаточно высокой надежностью закрытия только при использовании весьма длинных ключей.

Шифр Вижинера с ключом, состоящим из одной буквы, известен как шифр Цезаря, а с неограниченным неповторяющимся ключом – как шифр Верната.

Разновидностью шифра Вижинера является шифр Бофора, но при этом при определении цифрового эквивалента используют формулы

$$y_i = x_i - k_i \,(\mathrm{mod}\, 33) \text{ и } y_i = k_i - x_i \,(\mathrm{mod}\, 33).$$

При рассмотрении этих видов шифров становится очевидным, что чем больше длина ключа (например в шифре Вижинера), тем лучше шифр. Существенного улучшения свойств шифртекста можно достигнуть при использовании шифров с автоключом.

Шифр, в котором сам открытый текст или получающаяся криптограмма используются в качестве ключа, называется шифром с автоключом. Шифрование в этом случае начинается с ключа, называемого первичным, и продолжается с помощью открытого текста или криптограммы, смещенной на длину первичного ключа.

Пример 9.4. Открытый текст: «ШИФРОВАНИЕ ЗАМЕНОЙ».

Первичный ключ «КЛЮЧ» Схема шифрования с автоключом при использовании открытого текста представлена в таблице 9.6.

Таблица 9.6 - Схема шифрования с автоключом при использовании открытого текста

Ш	И	Ф	Р	О	В	А	Н	И	Е	□	З	А	М	Е	Н	0	И
К	Л	Ю	Ч	Ш	И	Ф	Р	0	В	А	Н	И	Е	□	З	А	М
36	21	52	41	40	12	22	31	24	09	34	22	10	19	39	22	16	23
В	Ф	Т	З	Ж	Л	Х	Ю	Ч	И	А	Х	И	Т	Е	Х	П	Ц

Схема шифрования с автоключом при использовании криптограммы представлена в таблице 9.7

Таблица 9.7 - Схема шифрования с автоключом при использовании криптограммы

Ш	И	Ф	Р	О	В	А	Н	И	Е	□	З	А	М	Е	Н	О	И
К	Л	Ю	Ч	В	Ф	Т	З	С	Ч	У	Х	Ъ	Э	У	Э	Ы	И
36	21	52	41	18	24	20	22	27	30	53	30	24	43	26	44	39	20
В	Ф	Т	З	С	Ч	У	Х	Ъ	Э	У	Э	Ы	И	Щ	К	И	У

Для шифрования используются и другие методы подстановки символов открытого текста в соответствии с некоторыми правилами.

Гомофоническая замена одному символу открытого текста ставит в соответствие несколько символов шифртекста. Этот метод применяется для искажения статистических свойств шифртекста.

Пример 9.5. Открытый текст «ЗАМЕНА». Подстановка задана таблицей 9.8.

Таблица 9.8 - Подстановка алфавита гомофонической замены

Алфавит открытого текста	А	Б	…	Е	Ж	З	…	М	Н	…
Алфавит шифртекста	17 31 48	23 44 63	…	97 51 15	47 67 33	76 19 59	…	32 28 61	55 84 34	…

Шифртекст «76-17-32-97-55-31».

Таким образом, при гомофонической замене каждая буква открытого текста заменяется по очереди цифрами соответствующего столбца.

Полиалфавитная подстановка использует несколько алфавитов шифртекста. Пусть используется k алфавитов. Тогда открытый текст

$$X = x_1 x_2 \ldots x_k x_{k+1} \ldots x_{2k} x_{2k+1}.$$

заменяется шифртекстом

$$Y = f_1(x_1)f_2(x_2)...f_k(x_k)f_1(x_{k+1})...f_k(x_{2k})f_1(x_{2k+1})...,$$

где $f_i(x_j)$ означает символ шифртекста алфавита i для символа открытого текста x_j.

Пример 9.6. Открытый текст «ЗАМЕНА» $k = 3$ Подстановка задана таблицей из примера 9.5. Шифртекст «76 31 61 97 84 48».

Прогрессивный ключ Тритемиуса, который изображен на рисунке 9.3, является примером полиалфавитного шифра. Строка, обозначенная как сдвиг 0, совпадает с обычным порядком букв в алфавите. Буквы в следующей строке сдвинуты на один символ влево с циклическим сдвигом оставшихся позиций. Каждая последующая строка получается с помощью такого же сдвига алфавита на один символ влево относительно предыдущей строки. Это продолжается до тех пор, пока в результате циклических сдвигов алфавита не будет смещен на все возможные позиции. Один из методов использования такого алфавита заключается в выборе первого символа шифрованного сообщения из строки, полученной при сдвиге на 1 символ, второго символа – из строки, полученной при сдвиге на 2 символа, и т.д.

Пример 9.7. Зашифровать сообщение «NOWISTHETIME» ключом Тритемиуса. Результат шифрования приведен в таблице 9.9.

Открытый текст:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Сдвиг: 0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	O	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	O	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	O	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Рисунок 9.3 – Прогрессивный ключ Тритемиуса.

Таблица 9.9 - Подстановка алфавита гомофонической замены

Исходный текст	N	O	W	I	S	T	H	E	T	I	M	E
Шифрованный текст	O	Q	Z	M	X	Z	O	M	C	S	X	Q

Полиграммная замена формируется из одного алфавита с помощью специальных правил. В качестве примера рассмотрим шифр Плэйфера [12].

В этом шифре алфавит располагается в матрице. Открытый текст разбивается на пары символов x_1, x_{k+1}. Каждая пара символов открытого текста заменяется на пару символов из матрицы следующим образом:

–если символы находятся в одной строке, то каждый из символов пары заменяется на стоящий правее его (за последним символом в строке следует первый);

–если символы находятся в одном столбце, то каждый символ пары заменяется на символ расположенный ниже его в столбце (за последним нижним символом следует верхний);

–если символы пары находятся в разных строках и столбцах, то они считаются противоположными углами прямоугольника. Символ, находящийся в левом углу заменяется на символ, стоящий в другом левом углу. Замена символа, находящегося в правом углу осуществляется аналогично;

–если в открытом тексте встречаются два одинаковых символа подряд, то перед шифрованием между ними вставляется специальный символ (например, тире).

Пример 9.8. Открытый текст «ШИФР ПЛЭЙФЕРА» Матрица алфавита представлена в таблице 9.6

Таблица 9.8 - Матрица алфавита шифра Плэйфера

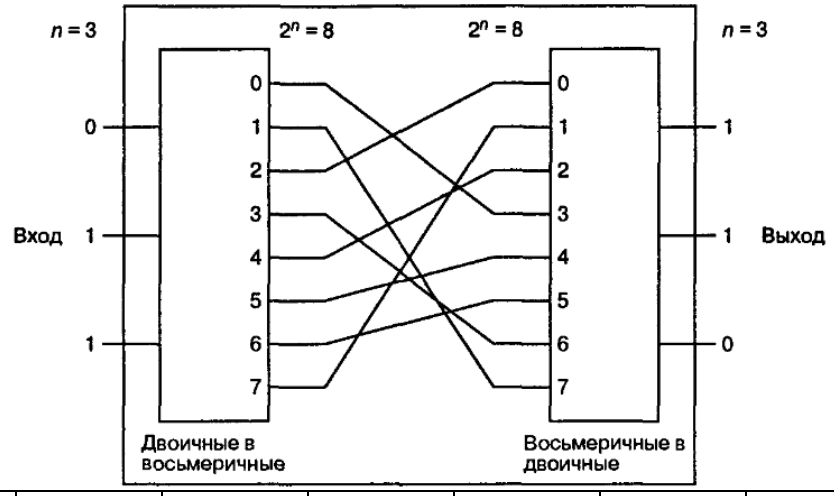
А	Ж	Б	М	Ц	В
Ч	Г	Н	Ш	Д	0
Е	Щ	,	Х	У	П
·	З	Ъ	Р	И	Й
С	Ь	К	Э	Т	Л
Ю	Я	□	Ы	Ф	–

Шифртекст «РДЫИ,-СТ-И. ХЧС».

Технология шифрования с помощью подстановки, например использование шифра Цезаря и прогрессивного ключа шифрования Тритемиуса, широко используется в головоломках. Такие простые подстановочные шифры дают малую защищенность. Чтобы к подстановочной технологии можно было приме-

нить концепцию *смещения,* требуется более сложное соотношение. Смещение – это подстановки, которые делают взаимосвязь между ключом и шифрованным текстом как можно более сложной. На рисунке 9.4 изображен пример создания большей подстановочной сложности с помощью использования нелинейного преобразования. В общем случае *n* входных битов сначала представляются как один из 2^n различных символов (на приведенном рисунке n=3). Затем множество из 2^n символов перемешивается так, чтобы каждый символ заменялся другим символом множества. После этого символ снова превращается в n-битовый.

Можно легко показать, что существует $(2^n)!$ Различные подстановки или связанные с ними возможные модели. Задача криптоаналитика становится вычислительно невозможной для больших n. Пусть n=128, тогда $2^n=10^{38}$ и $(2^n)!$ представляет собой астрономическое число. Видим, что для n=128 это преобразование с помощью блока подстановки (substitution block, S-блок) является сложным (запутывающим). Впрочем, хотя S-блок с n=128 можно считать идеальным, его реализация является невозможной, поскольку она потребует блока с $2^n = 10^{38}$ контактами.



Вход	000	001	010	011	100	101	110	111
Выход	011	111	000	110	010	100	101	001

Рисунок 9.4 – Блок подстановки

Чтобы убедиться, что S-блок, приведенный на рисунке 9.4, представляет собой *нелинейное преобразование*, достаточно использовать теорему о суперпозиции, которая формулируется ниже. Предположим, что

$$\begin{aligned} C &= Ta + Tb \\ C &= T(a+b), \end{aligned} \qquad (9.1)$$

где a и b – входные элементы, C и C’ – выходные элементы, а T - преобразование. Тогда если T линейно, C=C’ для всех входных элементов, а если T нелинейно, $C \neq C'$.

Предположим, a=001 и b=010; тогда, используя преобразование T, показанное на рисунке 9.4, получим следующее:

$$C = T(001) \oplus T(010) = 111 \oplus 000 = 111,$$
$$C' = T(001 \oplus 010) = T(011) = 110.$$

Здесь символ $\oplus$ обозначает сложение по модулю 2. Поскольку $C \neq C'$, S – блок является нелинейным.

9.2. Шифрование перестановкой

При этом методе открытый текст разбивается на группы определенной длины. Ключом задается порядок перестановки букв в группе.

При перестановке (транспозиции) буквы исходного открытого текста в сообщении не заменяются другими буквами алфавита, как в классических шифрах, а просто переставляются. Например, слово “THINK” после перестановки может выглядеть как шифрованный текст HKTNI. На рис. 9.5 приведен пример бинарной перестановки данных (линейная операция). Видно, что входные данные просто перемешиваются или переставляются. Преобразование выполняется с помощью блока перестановки (permutation block, P-блок). Технология, используемая сама по себе, имеет один основной недостаток: она уязвима по отношению к обманным сообщениям. Обманное сообщение изображено на рисунке 9.5. Подача на вход единственной 1 (при остальных 0) позволяет обнаружить одну из внутренних связей. Если криптоаналитику необходимо выполнить криптоанализ такой системы с помощью атаки открытого текста , он отправит последовательность таких обманных сообщений, при каждой передаче смещая единственную 1 на одну позицию. Таким образом, обнаруживаются все связи входа и выхода. Данный пример показывает, почему защищенность системы не должна зависеть от ее архитектуры.

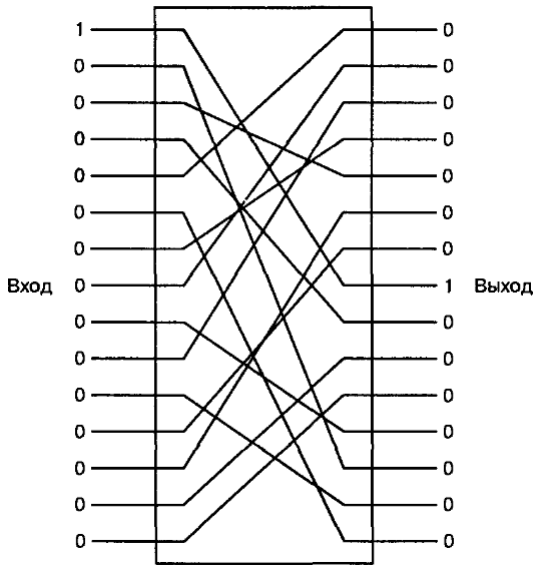


Рисунок 9.5 – Блок перестановки

Пример 9.9. Открытый текст «СДАЧА ЗАЧЕТА ПО ТЕЛЕМЕХАНИКЕ» правила перестановки группы из семи букв с порядковыми номерами 1 – 2 – 3 – 4 – 5 – 6 – 7 переставить в порядок – 7 – 1 – 5 – 3 – 6 – 4 – 2.

Шифртекст (криптограмма) «ЗСАА□ЧДПААЕ□ТЧМОЛТЕЕ□ЕЕИАКНХ».

Можно использовать и усложненную перестановку. Для этого открытый текст записывается в матрицу по определенному ключу *К*1. Шифртекст образуется при считывании из этой матрицы по ключу *К*2.

Пример 9.10. Открытый текст «ШИФРОВАНИЕ ПЕРЕСТАНОВКОЙ».

Матрица из четырех столбцов приведена в таблице 9.11, где запись по строкам в соответствии с ключом *К*1: 5 – 3 – 1 – 2 – 4 – 6, а чтение по столбцам в соответствии с ключом *К*2: 4 – 2 – 3 – 1

Таблица 9.11 - Матрица алфавита с перестановкой из четырех столбцов

1	И	Е	□	П
2	Е	Р	Е	С
3	0	В	А	Н
4	Т	А	Н	О
5	Ш	И	Ф	Р
6	В	К	О	Й
К1/К2	1	2	3	4

Шифртекст «ПСНОРЙЕРВАИК□ЕАНФОИЕОТШВ».

Наиболее сложные перестановки осуществляются по гамильтоновым путям, которых в графе может быть несколько.

Пример 9.11. Открытый текст «ШИФРОВАНИЕ ПЕРЕСТАНОВКОЙ» Ключ – гамильтонов путь на графе (рисунок 9.6).

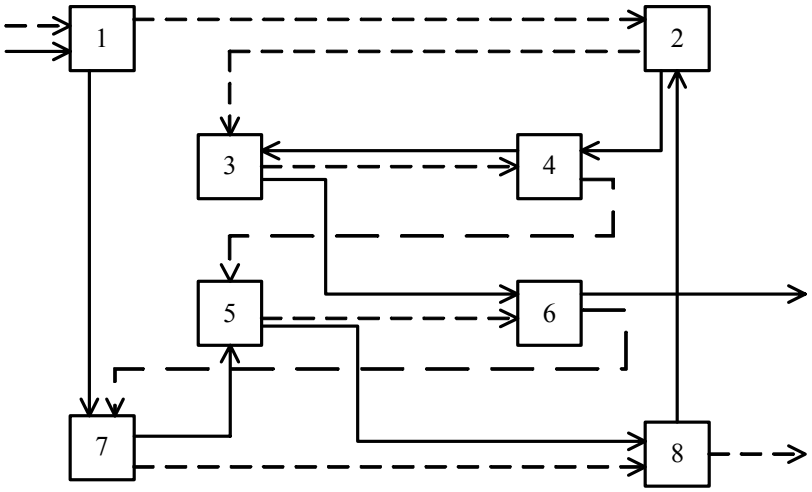


Рисунок 9.6 – Гамильтонов путь на графе

Шифртекст «ШАОНИРФВИЕЕСЕП□РТОВЙАОНК»

Чтение криптограммы (1 – 7 – 5 – 8 – 2 – 4 – 3 – 6).

Запись открытого текста (1 – 2 – 3 – 4 – 5 – 6 – 7 – 8).

Необходимо отметить, что для данного графа из восьми вершин можно предложить несколько маршрутов записи открытого текста и несколько гамильтоновых путей для чтения криптограмм.

В 1991 г. В.М. Кузьмич предложил схему перестановки, основанную на кубике Рубика. Согласно этой схеме открытый текст записывается в ячейки граней куба по строкам. После осуществления заданного числа заданных поворотов слоев куба считывание шифртекста осуществляется по столбцам. Сложность расшифрования в этом случае определяется числом ячеек на гранях куба и сложностью выполненных поворотов слоев. Перестановка, основанная на кубике Рубика, получила название объемной (многомерной) перестановки [12].

В 1992-1994 гг. идея применения объемной перестановки для шифрования открытого текста получила дальнейшее развитие. Усовершенствованная схема перестановок по принципу кубика Рубика, в которой наряду с открытым текстом перестановке подвергаются и функциональные элементы самого алгоритма шифрования легла в основу секретной системы «Рубикон». В качестве прообразов пространственных многомерных структур, на основании объемных преобразований которых осуществляются перестановки, в системе «Рубикон» используются трехмерный куб и тетраэдр.

9.3. Шифрование гаммированием

В процессе шифрования цифровые эквиваленты знаков криптографически закрываемого сообщения складываются с псевдослучайной последовательностью чисел, именуемой гаммой, и приводятся по модулю K, где K – объем алфавита знаков. Таким образом, псевдослучайная последовательность выполняет здесь роль ключа.

Пример 9.12. Открытый текст «ПЕРЕДАЧА» («16-06-17-05-06-01-24-01» согласно табл. 9.5). Гамма «04–11–14–30–02–10–25».

Операцию сложения по mod 33:

$y_1 = 16 + 04 = 20$, $y_2 = 06 + 11 = 17$, $y_3 = 17 + 14 = 31$, $y_4 = 06 + 30 = 03$, y_5=05+02=07, $y_6 = 01 + 10 = 11$, $y_7 = 24 + 25 = 16$, $y_8 = 01 + 04 = 05$.

Криптограмма «УРЮВЖКПД» («20 – 17 – 31 – 03 – 07 – 11 – 16 – 05»).

Наиболее широко гаммирование используется для криптографического закрытия сообщений, уже выраженных в двоичном коде.

Пример 9.13. Открытый текст «ИНФОРМАЦИЯ» («09 – 14 – 21 – 15 – 17 – 13 – 01 – 23 – 09 – 32» согласно табл. 9.5). Псевдослучайная последовательность чисел (гамма)

«02 – 13 – 24 – 04 – 11 – 17 – 14 – 15 – 09 – 06 – 03 – 21».

Запишем код каждой буквы открытого текста и каждую цифру гаммы в двоич-

ном виде, используя шесть разрядов, получим код буквы:

001001-001110-010101-001111-010001-001101-000001-010111-001001-100000;

Код цифры гаммы: 000010-001101-011000-000100-001011-010001-001110-001111-001001-000110-000011-010101.

Сложим цифровые эквиваленты в двоичном коде буквы и гаммы по модулю два. В результате чего получим:

$$\begin{array}{ccccc} 001001 & 001110 & 010101 & 001111 & 010001 \\ \oplus & \oplus & \oplus & \oplus & \\ \underline{000010} & \underline{001101} & \underline{011000} & \underline{000100} & \underline{001011} \\ 001011 & 000011 & 001101 & 001011 & 011010 \end{array}$$

$$\begin{array}{ccccc} 001101 & 000001 & 010111 & 001001 & 100000 \\ \oplus & \oplus & \oplus & \oplus & \\ \underline{010001} & \underline{001110} & \underline{010111} & \underline{001001} & \underline{000110} \\ 011100 & 001111 & 011000 & 000000 & 100110 \end{array}$$

Таким образом, в канал связи будет передана последовательность «001011-000011-001101-001011-011010-011100-001111-011000-000000-100110».

Расшифрование данных сводится к повторной генерации гаммы шифра при известном ключе и наложению этой гаммы на зашифрованные данные. В нашем случае на приемной стороне генерируется гамма «000010-001101-011000-000100-001011-010001-001110-001111-001001-000110-000011-010101», которая складывается по модулю два с принятой кодовой комбинацией. Считаем, что канал связи не внес искажений в переданную последовательность, поэтому сложим ее с гаммой сгенерированной на приемной стороне. В результате чего получим:

$$\begin{array}{ccccc} 001011 & 000011 & 001101 & 001011 & 011010 \\ \oplus & \oplus & \oplus & \oplus & \\ \underline{000010} & \underline{001101} & \underline{011000} & \underline{000100} & \underline{001011} \\ 001001 & 001110 & 010101 & 001111 & 010001 \end{array}$$

$$\begin{array}{ccccc} 011100 & 001111 & 011000 & 000000 & 100110 \\ \oplus & \oplus & \oplus & \oplus & \\ \underline{010001} & \underline{001110} & \underline{001111} & \underline{001001} & \underline{000110} \\ 001101 & 000001 & 010111 & 001001 & 100000 \end{array}$$

Таким образом получим последовательность цифр в двоичном коде «001001-001110-010101-001111-010001-001101-000001-010111-001001-100000» или в десятичном коде «09-14-21-15-17-13-01-23-09-32», что соответствует тексту «ИНФОРМАЦИЯ» совпадающему с открытым текстом.

Надежность криптографического закрытия методом гаммирования опре-

деляется главным образом длиной неповторяющейся части гаммы. Если она превышает длину закрываемого текста, то раскрыть криптограмму, опираясь только на результаты статистической обработки этого текста, теоретически невозможно.

Однако если удается получить некоторое число двоичных символов исходного текста и соответствующих им двоичных символов криптограммы, то сообщение нетрудно раскрыть, так как преобразование, осуществляемое при гаммировании, является линейным. Для полного раскрытия достаточно всего $2n$, где n – число разрядов регистра, формирующего псевдослучайную последовательность двоичных символов зашифрованного и соответствующего ему исходного текста.

9.4. Стандарт шифрования данных DES

DES(Data Encryption Standart) – государственный стандарт США. Стандарт DES стал одним из первых «открытых» шифроалгоритмов. Все схемы используемые для его реализации, были опубликованы и тщательно проверены. Секретным был только ключ, с помощью которого осуществляется кодирование и декодирование информации.

Алгоритм DES базируется на научной работе Шеннона 1949 г., связавшей криптографию с теорией информации. Шеннон выделил два общих принципа используемых в практических шифрах рассеивание и перемешивание. Рассеиванием он назвал распространение влияния одного знака открытого текста на множество знаков шифротекста, что позволяет скрыть статистические свойства открытого текста. Под перемешиванием Шеннон понимал использование взаимосвязи статистических свойств открытого и шифрованного текста. Однако шифр должен не только затруднять раскрытие, но и обеспечивать легкость шифрования и дешифрования при известном секретном ключе. Поэтому была принята идея использовать произведение простых шифров, каждый из которых вносит небольшой вклад в значительное суммарное рассеивание и перемешивание.

В составных шифрах в качестве элементарных составляющих чаще всего используются простые подстановки и перестановки. При многократном чередовании простых перестановок и подстановок можно получить очень стойкий шифр (криптоалгоритм) с хорошим рассеиванием и перемешиванием.

Стандарт шифрованных данных DES – один из наиболее удачных примеров криптоалгоритма, разработанного в соответствии с принципами рассеивания и перемешивания. В нем открытый текст, криптограмма и ключ являются двоичными последовательностями длиной соответственно $M = 64$, $N = 64$, $K = 56$ бит. Криптоалгоритм DES представляет собой суперпозицию элементарных шифров, состоящую из 16 последовательных шифроциклов, в каждом из которых довольно простые перестановки с подстановками в четырехбитовых группах B каждом проходе используются лишь 48 бит ключа, однако они выби-

раются внешне случайным образом из полного 56-битового ключа.

Операции шифрования и дешифрования осуществляются по схеме, представленной на рисунке 9.7.

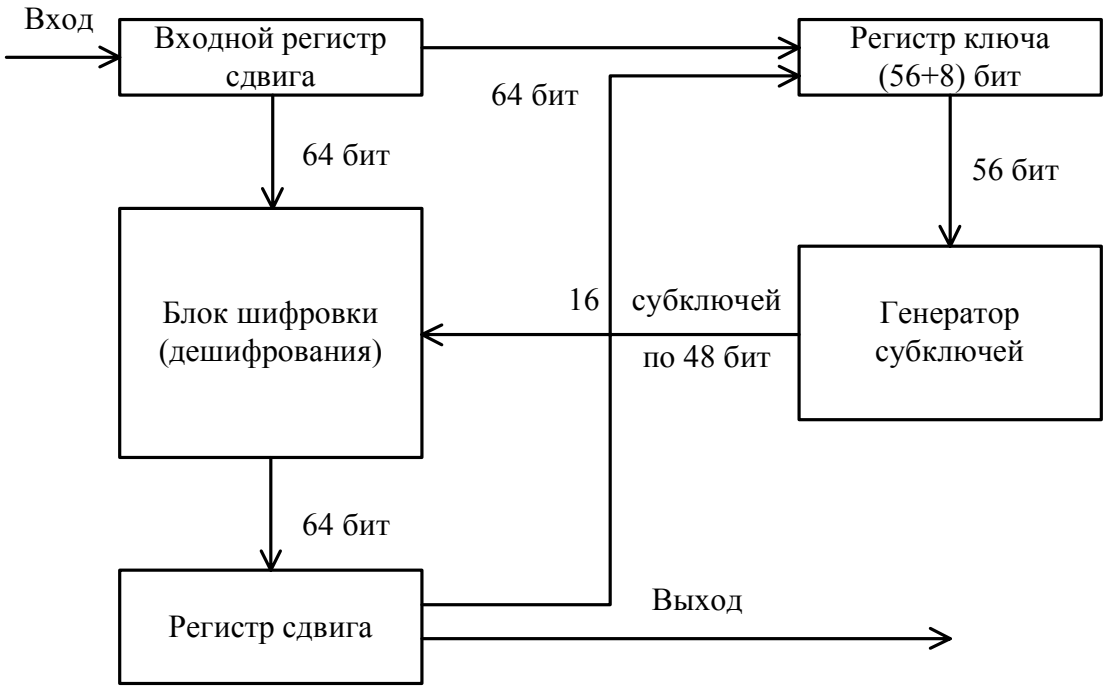


Рисунок 9.7 - Шифратор (дешифратор) в стандарте DES

Перед началом шифрования в специализированный регистр устройства через входной регистр вводится ключ, содержащий 64 бит, из которых 56 используется для генерации субключей, а 8 являются проверочными. Ключ из устройства вывести нельзя. Предусмотрена возможность формирования нового ключа внутри устройства. При этом ключ, вводимый в устройство, шифруется ранее использовавшимся ключом и затем через выходной регистр вводится в специализированный регистр в качестве нового ключа. Шестнадцать субключей по 48 бит каждый, сформированных в генераторе субключей, используется для шифрования блока из 64 символов, поступающих во входной регистр устройства. Шифрование осуществляется из 16 логически идентичных шагов, на каждом из которых используется один из субключей.

Процесс дешифрования выполняется по тому же алгоритму, что и процесс шифрования, с той лишь разницей, что субключи генерируются в обратном порядке.

Алгоритм DES используется как для шифрования, так и для установления подлинности (аутентификации) данных. С точки зрения системы ввода-вывода DES может считаться блочной системой шифрования с алфавитом в 2^{64} символа. Входной блок из 64 бит, который является в этом алфавите символом открытого текста, заменяется новым символом шифрованного текста. На рисунке 9.8 в виде блочной диаграммы показаны функции системы. Алгоритм шифро-

вания начинается с начальной перестановки 64 бит открытого текста, описанной в таблице начальной перестановки (таблица 9.12). Таблица начальной перестановки читается слева направо и сверху вниз, так что после перестановки биты $x_1, x_{2,} ..., x_{64}$ превращаются в $x_{58}, x_{50}, ..., x_7$. После этой начальной перестановки начинается основная часть алгоритма шифрования, состоящая из 16 итераций, которые используют стандартный блок, показанный на рисунке 9.9. Для преобразования 64 бит входных данных в 64 бит выходных, определенных как 32 бит левой половины и 32 бит правой, стандартный блок использует 48 бит ключа. Выход каждого стандартного блока становится входом следующего стандартного блока. Входные 32 бит правой половины (R_{i-1}) без изменений подаются на выход и становятся 32 бит левой половины (L_i). Эти R_{i-1} бит с помощью таблицы расширения (таблица 9.13) также расширяются и преобразуются в 48 бит, после чего суммируются по модулю 2 с 48 бит ключа. Как и в случае таблицы начальной перестановки, таблица расширения читается слева направо и сверху вниз.

Таблица 9.12 – Начальная перестановка

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 9.13 – Таблица расширения

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	12	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Данная таблица расширяет биты

$$R_{i-1} = x_{1,}\, x_{2,\ldots,} x_{32}$$

в биты

$$(R_{i-1})_E = x_{32}, ..., x_1, x_2, ..., x_{32}, x_{1.} \qquad (9.2)$$

Отметим, что биты, обозначенные в первом и последнем столбцах таблицы расширения, - это те битовые разряды, которые дважды использовались для расширения от 32 до 48 бит.

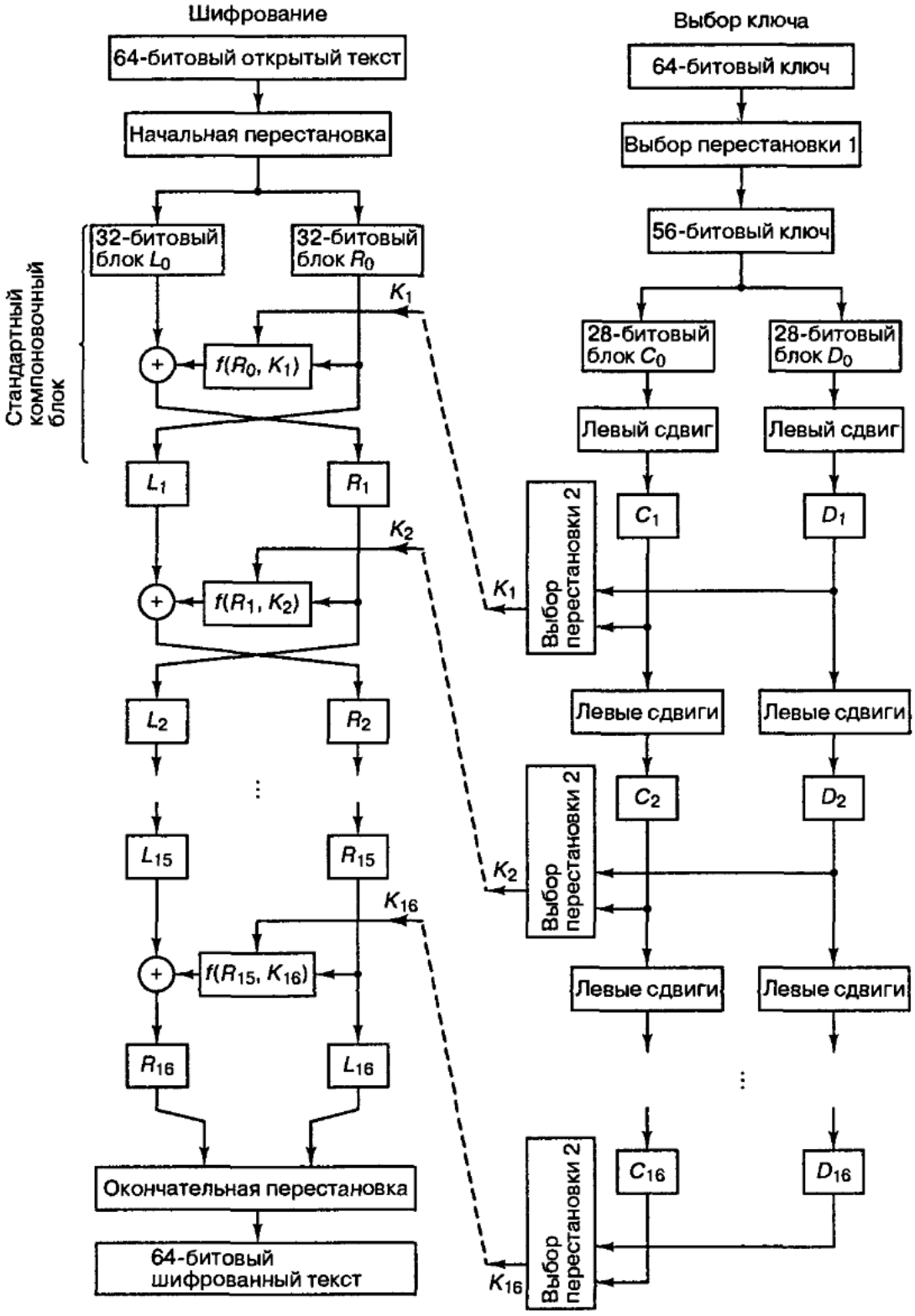


Рисунок 9.8 – Стандарт шифрования данных DES

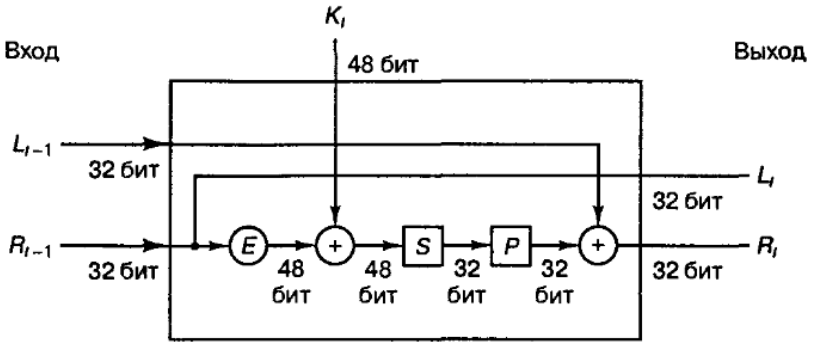


Рисунок 9.9 – Стандартный компоновочный блок

Далее $(R_{i-1})_E$ суммируется по модулю 2 с i-м ключом, выбор которого описывается позднее, а результат разделяется на восемь 6-битовых блоков.

$$B_1, B_2, ..., B_8$$

Иными словами,

$$(R_{i-1})_E \oplus K_i = B_1, B_2, ..., B_8 \tag{9.3}$$

Каждый из восьми 6-битовых блоков B_i используется как вход функции S – блока, возвращающей 4 – битовый блок $S_j(B_j)$. Таким образом, входные 48 бит с помощью функции S – блока преобразуются в 32 бит. Функция отображения S – блока S_j определена в таблице 9.14. Преобразование $B_j = b_1, b_2, b_3, b_4, b_5, b_6$ выполняется следующим образом. Нужная строка – это $b_1 b_6$ а нужный столбец - $b_2 b_3 b_4 b_5$. Например, если $b_1 = 110001$, то преобразование S_1 возвращает значение из строки 3, столбца 8, т.е. число 5 (в двоичной записи 0101). 32-битовый блок, полученный на выходе S – блока, переставляется с использованием таблицы перестановки (таблица 9.15). Как и другие таблицы, P – таблица читается слева направо и сверху вниз, так что в результате перестановки битов $x_1, x_2, ..., x_{32}$ получаем $x_{16}, x_7, ..., x_{25}$. 32-битовый выход P – таблицы суммируется по модулю 2 с 32 бит левой половины (L_{i-1}), образуя выходные 32 бит правой половины (R_i).

Алгоритм стандартного блока может быть представлен следующим образом:

$$L_i = R_{i-1} \tag{9.4}$$

$$R_i = L_{i-1} \oplus f(R_{i-1,} K_i). \tag{9.5}$$

Здесь $f(R_{i-1,} K_i)$ обозначает функциональное соотношение, включающее описанные выше расширение, преобразование в S – блоке и перестановку. По-

сле 16 итераций в таких стандартных блоках данные размещаются согласно окончательной обратной перестановке, описанной в таблице 9.16, где, как и ранее, выходные биты читаются слева направо и сверху вниз.

Для дешифрования применяется тот же алгоритм, но ключевая последовательность, используемая в стандартном блоке, берется в обратном порядке. Отметим, что значение $f(R_{i-1,}K_i)$ которое может быть также выражено через выход i-го блока как $f(L_{i,}K_i)$, делает процесс дешифрования возможным.

Таблица 9.14 – Функции выбора S-блока

	Столбец																
Строка	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	S_1
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	S_2
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	S_3
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	S_4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	S_5
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	S_6
3	4	3	2	12	9	5	15	0	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	S_7
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	S_8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Таблица 9.15 – Таблица перестановки

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Таблица 9.16 – Окончательная перестановка

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29

Продолжение таблицы 9.16

36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Выбор ключа. Выбор ключа также происходит в течение 16 итераций, как показано в соответствующей части рис.9.8. Входной ключ состоит из 64-битового блока с 8 бит четности в разрядах 8, 16, …, 64. Перестановочный выбор 1 отбрасывает биты четности и переставляет оставшиеся 56 бит согласно табл. 9.17. Выход данной процедуры делится пополам на два элемента – C и D, каждый из которых состоит из 28 бит. Выбор ключа проходит в 16 итерациях, проводимых для создания различных множеств 48 ключевых бит для каждой итерации шифрования. Блоки C и D последовательно сдвигаются согласно следующим выражениям:

Таблица 9.17 – Круговая перестановка

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

$$C_i = LS_i(C_{i-1})\, u\, D_i = LS_i(D_{i-1}). \quad (9.6)$$

Здесь LS_i - левый циклический сдвиг на число позиций, показанных в таблице 9.18. Затем последовательность C_i, D_i переставляется согласно пере-

становочному выбору 2, показанному в таблице 9.19. Результатом является ключевая последовательность K_i, которая используется в i-й итерации алгоритма шифрования.

Таблица 9.18 – Ключевая последовательность сдвигов влево

Итерация i	Количество сдвигов влево
1	1
2	1
3	2
4	2
5	2
6	2

Продолжение таблицы 9.18

Итерация i	Количество сдвигов влево
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Таблица 9.19 – Ключевая перестановка 2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

DES может реализовываться подобно блочной системе шифрования (см. рисунок 9.8), что иногда называют методом *шифровальной книги*. Основным недостатком этого метода является то, что (при использовании одного ключа) данный блок входного открытого текста будет всегда давать тот же выходной шифрованный блок. Еще один способ шифрования, называемый способом шифрования с обратной связью, приводит к шифрованию отдельных битов, а не символов, что дает поточное шифрование. В системе шифрования с обратной связью (описанной ниже) шифрование сегмента открытого текста зависит не

только от ключа и текущих данных, но и от некоторых предшествующих данных.

9.5. Симметричные криптосистемы. Алгоритм IDEA.

Алгоритм IDEA (International Data Encryption Algorithm) относится к классу симметричных шифраторов. Данный алгоритм был разработан в 1990 г. в качестве альтернативы алгоритму DES (Data Encryption Standard). В основе алгоритма лежит идея смешанного преобразования, которое случайным образом равномерно распределяет исходный текст по всему пространству шифротекста.

Смешанные преобразования реализуются при помощи перемежающихся последовательностей замен и простых операций перестановок. Преобразование данных производится по блокам, размер которых равен 64 битам. Длина ключа в алгоритме IDEA составляет 128 бит.

Каждый 64-битный блок рассматривается как четыре 16-битных подблока, которые преобразуются с использованием следующих целочисленных операций:

- Побитное сложение по модулю 2 (XOR) двух 16-битных операндов, которое будем обозначать как $\oplus$.
- Сложение двух целых 16-битных операндов по модулю 216, обозначенное как ⊠.
- Умножение двух чисел без знака по модулю $2^{16}+1$. Результат операции умножения усекается до длины в 16 бит. При вычислении данной операции существует исключение для кода со всеми нулями, который при умножении рассматривается как число 2^{16}. Данную операцию будем обозначать как $\odot$.

Процедура шифрования состоит из 8-ми одинаковых раундов и дополнительного 9-го выходного раунда (рисунок 9.10, а).

На выходе 9-го раунда формируется содержимое четырёх 16-битных подблоков, образующих блок шифротекста.

Основной частью каждого раунда является мультипликативно-аддитивная структура (рисунок 9.10, б).

Здесь F1 и F2 – 16-битные значения, полученные из открытого текста, Z5 и Z6 – 16-битные подключи.

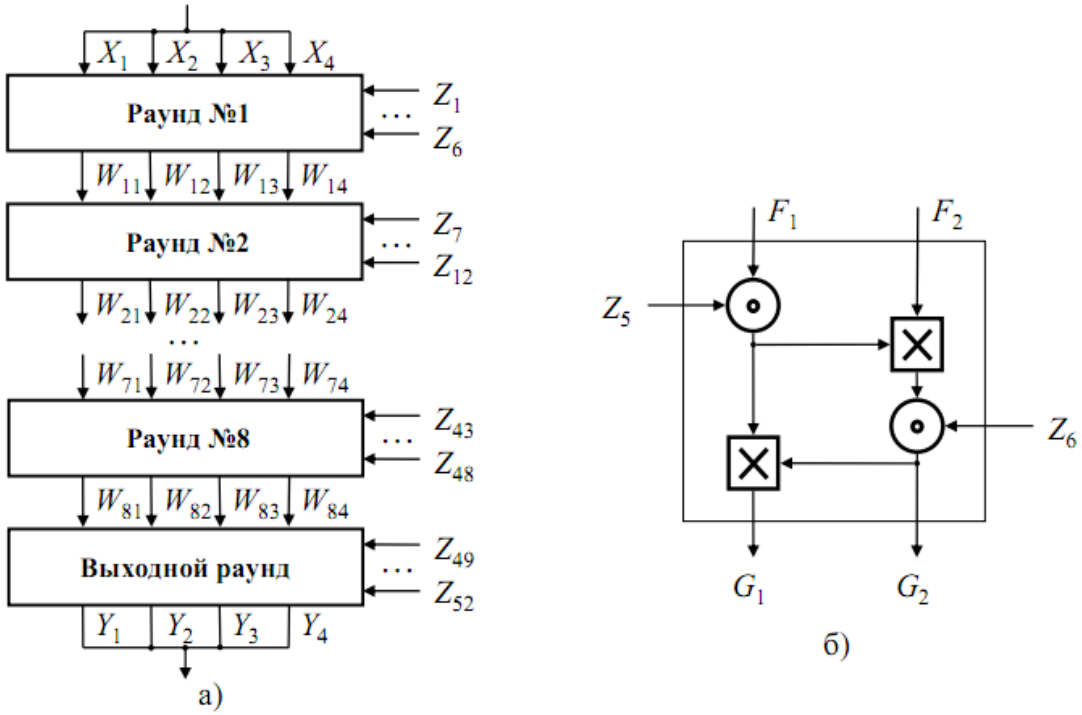


Рисунок 9.10 - Алгоритм IDEA: а) схема процедуры шифрования; б) мультипликативно-аддитивная структура

Все операнды, участвующие в выполнении процедуры шифрования, имеют размерность 16 бит.

На рисунке 9.11 приведена схема выполнения первого раунда алгоритма IDEA.

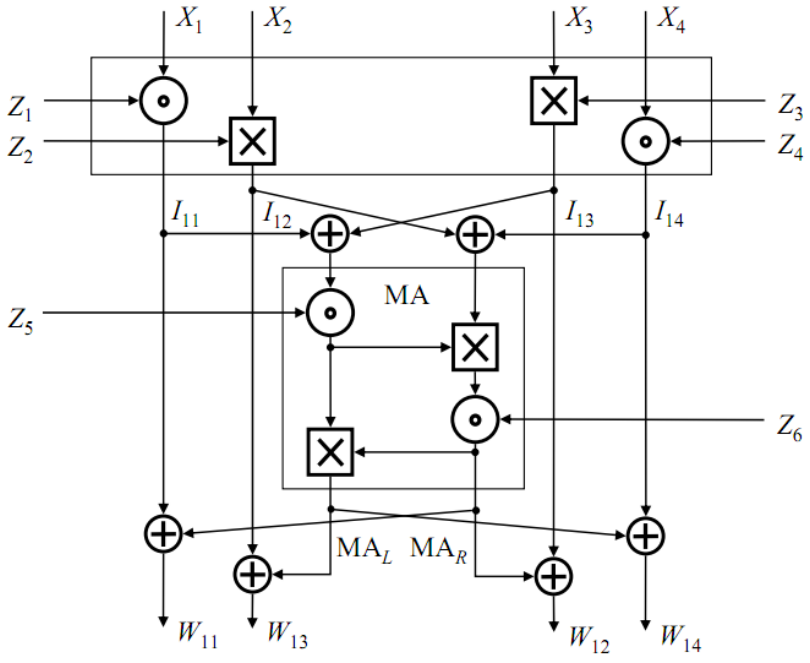


Рисунок 9.11 – первый раунд шифрования алгоритма IDEA

Данные, получаемые на выходе i-го раунда шифрования, подаются на вход (i+1)-го раунда. Входными данными 1-го раунда являются четыре 16-битных подблока (X_1, X_2, X_3, X_4) 64-битного блока исходного текста.

Схема выполнения 9-го раунда шифрования приведена на рисунке 9.12.

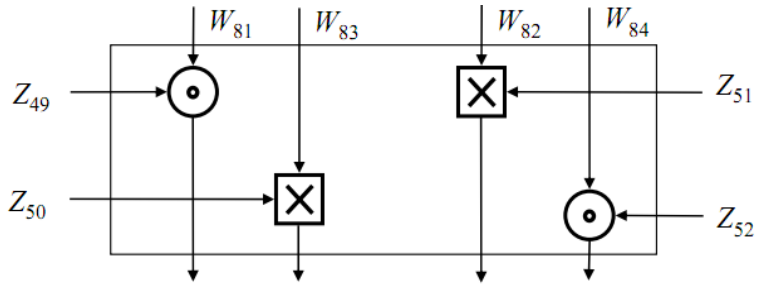


Рисунок 9.12 – Девятый раунд шифрования алгоритма IDEA

Следует обратить внимание на то, что 2-й и 3-й подблоки промежуточного значения W меняются местами после выполнения всех раундов кроме восьмого.

На каждом из девяти раундов используются значения 16-битных итерационных ключей Z_i, которые получаются путём преобразования исходного 128-битного ключа K.

Первые 8 итерационных ключей $Z_1 ... Z_8$ берутся как восемь последовательных частей 128-битного ключа. Для получения следующих 8-ми итерационных ключей 128-битное значение ключа K циклически сдвигается на 25 бит влево и ключи $Z_9 ... Z_{16}$ вновь берутся как его 8 последовательных частей. Данный процесс повторяется до тех пор, пока не будут получены все 52 итерационных ключа.

Процедура расшифрования состоит из тех же девяти раундов, но только выполняемых с использованием иных значений итерационных ключей. Итерационные ключи расшифрования получают из итерационных ключей шифрования на основе таблицы соответствия (таблица 9.20).

Таблица 9.20 – Значения ключей, используемых в алгоритме IDEA для дешифрования

Итерация (раунд)	Обозначение	Эквивалентное обозначение
1	$U_1, U_2, U_3, U_4, U_5, U_6$	$Z_{49}^{-1}, -Z_{50}, -Z_{51}, Z_{52}^{-1}, Z_{47}, Z_{48}$
2	$U_7, U_8, U_9, U_{10}, U_{11}, U_{12}$	$Z_{43}^{-1}, -Z_{45}, -Z_{44}, Z_{46}^{-1}, Z_{41}, Z_{42}$
3	$U_{13}, U_{14}, U_{15}, U_{16}, U_{17}, U_{18}$	$Z_{37}^{-1}, -Z_{39}, -Z_{38}, Z_{40}^{-1}, Z_{35}, Z_{36}$
4	$U_{19}, U_{20}, U_{21}, U_{22}, U_{23}, U_{24}$	$Z_{31}^{-1}, -Z_{33}, -Z_{32}, Z_{34}^{-1}, Z_{29}, Z_{30}$
5	$U_{25}, U_{26}, U_{27}, U_{28}, U_{29}, U_{30}$	$Z_{25}^{-1}, -Z_{27}, -Z_{26}, Z_{28}^{-1}, Z_{23}, Z_{24}$
6	$U_{31}, U_{32}, U_{33}, U_{34}, U_{35}, U_{36}$	$Z_{19}^{-1}, -Z_{21}, -Z_{20}, Z_{22}^{-1}, Z_{17}, Z_{18}$
7	$U_{37}, U_{38}, U_{39}, U_{40}, U_{41}, U_{42}$	$Z_{13}^{-1}, -Z_{15}, -Z_{14}, Z_{16}^{-1}, Z_{11}, Z_{12}$
8	$U_{43}, U_{44}, U_{45}, U_{46}, U_{47}, U_{48}$	$Z_7^{-1}, -Z_9, -Z_8, Z_{10}^{-1}, Z_5, Z_6$
9	$U_{49}, U_{50}, U_{51}, U_{52}$	$Z_1^{-1}, -Z_2, -Z_3, Z_4^{-1}$

При этом выполняются следующие соотношения:

$$Z_j^{-1} \odot Z_j = 1 \bmod (2^{16}+1); \tag{9.7}$$

$$-Z_j \boxtimes Z_j = 0 \bmod 2^{16}. \tag{9.8}$$

Таким образом, для ключа Z_j значение, обозначаемое как $-Z_j$, является аддитивным инверсным по модулю 2^{16}, а значение, обозначаемое как Z_j^{-1} – мультипликативным инверсным по модулю $2^{16}+1$.

Порядок использования итерационных ключей при шифровании показан на рисунке 9.13.

При выполнении расшифрования раунды алгоритма выполняются в таком же порядке. На вход первого раунда подаётся четыре 16-битных подблока 64-битного блока шифротекста. Значения, полученные после выполнения выходного раунда, являются подблоками 64-битного блока исходного текста. Отличие от процедуры шифрования заключается в том, что вместо ключей $Z_1 \ldots Z_{52}$ используются ключи $U_1 \ldots U_{52}$.

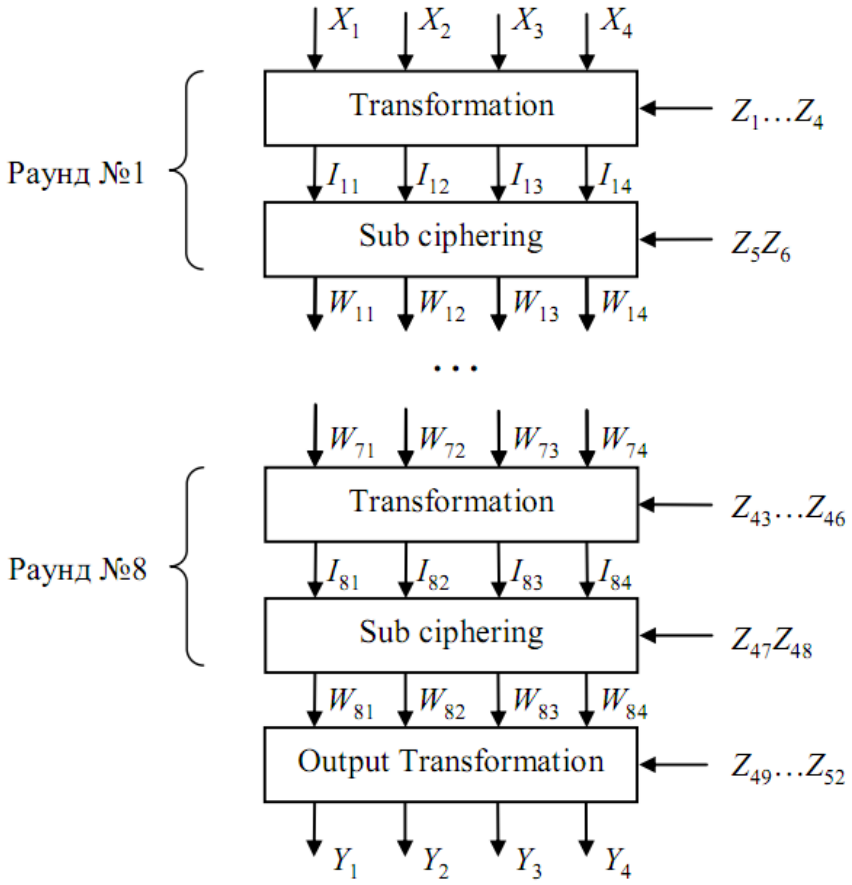


Рисунок 9.13 - Порядок использования итерационных ключей алгоритма IDEA

9.6. Криптосистема без передачи ключей

В информационных сетях использование традиционных систем шифрования с ключом затрудненно необходимостью иметь специальный особо защищенный способ для передачи ключа. В 1976 году У. Диффи (Diffie W.) и М. Хеллман (Hellman M.) - инженеры-электрики из Станфордского университета, а также студент Калифорнийского университета Р. Меркль (Merkle R.), предложили новый принцип построения криптосистем, не требующий передачи ключа принимающему сообщение и сохранения в тайне метода шифрования. В дальнейшем, в качестве примеров, рассмотрим три системы, основанные на идеях Диффи и Хеллмана: без передачи ключей, с открытым ключом и электронную подпись - все они в свою очередь основаны на математическом фундаменте теории чисел.

Пусть абоненты *A, B, C,...* условились организовать между собой секретную переписку. Для этой цели они выбирают достаточно большое простое число p такое, что $p-1$ хорошо разлагается на не очень большие простые множители. Затем каждый из абонентов независимо один от другого выбирает себе некоторое натуральное число, взаимно простое с $p-1$. Пусть число абонента *A* - a, абонента *B* - b и т.д. Числа a, b,... составляют первые секретные ключи соответствующих абонентов. Вторые секретные ключи *(*α для *A,* β для *B* и т.д.) находятся из уравнений: для *A* из $a\alpha \equiv 1 \pmod{\varphi(p)}$, $0 < \alpha < p-1$; для *B* – из $b\beta \equiv 1 \pmod{\varphi(p)}$, $0 < \beta < p-1$ и т.д. Пересылаемые сообщения, коды-числа, должны быть меньше $p-1$. В случае, когда сообщение больше или равно $p-1$, оно разбивается на части таким образом, чтобы каждая часть была числом, меньшим $p-1$.

Предположим абонент *A* решил отправить сообщение $m\,(m < p-1)$ *B*. Для этого он сначала зашифровывает свое сообщение ключом a а, получая по формуле $m_1 \equiv m^a \pmod{p}$ шифрованное сообщение m_1, которое отправляется *B*. *B,* получив m_1, зашифровывает его своим ключом b, получая по формуле $m_2 \equiv m_1^b \pmod{p}$ шифрованное сообщении m_2, которое отправляется обратно *к A*. *A* шифрует полученное сообщение ключом α по формуле $m_3 \equiv m_2^b \pmod{p}$ и окончательно отправляет m_3 *к B*. *B,* используя ключ β, сможет теперь расшифровать исходное сообщение *m*. Действительно, $m_4 \equiv m_3^\beta \equiv m^{a\alpha b\beta} \equiv m \pmod{p}$, т.к. $a\alpha b\beta \equiv 1 \pmod{\varphi(p)}$, следовательно, $a\alpha b\beta \equiv k\varphi(p)+1$ для некоторого целого k и $m^{k\varphi(p)+1} \equiv (m^{\varphi(p)})^k m \equiv m \pmod{p}$, т.к. $m^{\varphi(p)} \equiv 1 \pmod{p}$ по теореме Эйлера-Ферма.

Пример 9.12. Абоненты *A и B* вместе выбрали $p = 23$ $(\varphi(23) = 22)$, *A* выбрал $a = 5$, а *B* - $b = 7$. Затем из уравнения $5\alpha \equiv 1 \pmod{\varphi(23)}$ *A* находит $\alpha = 9$, а *B* из подобного уравнения находит $\beta = 19$. При передаче сообщения $m = 17$ от *A* к *B*

сначала A отправляет к B $m_1 \equiv 17^5 \equiv 21 \pmod{23}$, из $m_1 = 21$ B вычисляет $m_2 \equiv 21^7 \equiv 10 \pmod{23}$ и отправляет его обратно A, из $m_2 = 10$ A вычисляет для B $m_3 \equiv 10^9 \equiv 20 \pmod{23}$, наконец, B может прочитать посланное ему сообщение $20^{19} \equiv 17 \pmod{23}$.

9.7. Криптосистема с открытым ключом

Первую и наиболее известную систему с открытым ключом разработали в 1978 году американцы Р. Ривест (Rivest R.), Э. Шамир (Shamir A.) и Л. Адлеман (Adleman L.). По их именам эта система получила название RSA.

Пусть абоненты A и B решили организовать для себя возможность секретной переписки. Для этого каждый из них независимо выбирает два больших простых числа (pA_1, pA_2 и pB_1, pB_2), находит их произведение (r_A и r_B), функцию Эйлера от этого произведения ($\varphi(r_A)$ и $\varphi(r_B)$) и случайное число (a и b), меньшее вычисленного значения функции Эйлера и взаимно простое с ним. Кроме того, A из уравнения $a\alpha \equiv 1 \pmod{\varphi(r_A)}$ находит α ($0 < a < \varphi(r_A)$), а B из уравнения $b\beta \equiv 1 \pmod{\varphi(r_B)}$ находит β ($0 < \beta < \varphi(r_B)$). Затем A и B печатают доступную всем книгу паролей вида:

A: r_A, α

B: r_B, b

Теперь кто-угодно может отправлять конфиденциальные сообщения A или B. Например, если пользователь книги паролей хочет отправить сообщение m для B (m должно быть меньшим r_B, или делиться на куски, меньшие r_B), то он использует ключ b из книги паролей для получения шифрованного сообщения m_1 по формуле $m_1 \equiv m^b \pmod{r_B}$, которое и отправляется B. B для дешифровки m_1 использует ключ β в формуле $m_1^{\,b} \equiv m^{b\beta} \equiv m \pmod{r_B}$, т. к. $b\beta \equiv 1 \pmod{\varphi(r_B)}$, следовательно, $b\beta \equiv k\varphi(r_B)+1$ для некоторого целого k и $m^{k\varphi(rB)+1} \equiv (m^{\varphi(rB)})^k m \equiv m \pmod{r_B}$, т. к. $m^{\varphi(rB)} \equiv 1 \pmod{r_B}$ по теореме Эйлера-Ферма. Доказано [12], что задача нахождения секретного ключа β по данным из книги паролей имеет ту же сложность, что и задача разложения числа r_B на простые множители.

Пример 9.13. Пусть для A $pA_1 = 7$ и $pA_2 = 23$, тогда $r_A = pA_1\, pA_2 = 161$, $\varphi(161) = 6 * 22 = 132$, $\alpha = 7$, $\alpha = 19$ (из уравнения $7\alpha \equiv 1 \pmod{132}$). Следовательно, запись в книге паролей для A будет иметь вид A: 161, 7. Если кто-то захочет отправить A секретное сообщение $m = 3$, то он должен сначала превратить его в шифровку m_1 по формуле $m_1 \equiv 3^7 \equiv 94 \pmod{161}$. Когда A получит $m_1 = 94$ он дешифрует его по формуле $m \equiv 94^{19} \equiv 3 \pmod{161}$.

9.8. Электронная подпись

Криптосистема с открытым ключом открыта для посылки сообщений для абонентов из книги паролей для любого желающего. В системе с электронной подписью сообщение необходимо “подписывать”, т.е. явно указывать на отправителя из книги паролей.

Пусть W_1, W_2,... , W_n - абоненты системы с электронной подписью. Все они независимо друг от друга выбирают и вычисляют ряд чисел точно так же как и в системе с открытым ключом. Пусть i-ый абонент ($1 \neq i \leq n$) выбирает два больших простых числа p_{i1} и p_{i2}, затем вычисляет их произведение - $r_i = p_{i1}p_{i2}$ и функцию Эйлера от него – $\varphi(r_i)$, затем выбирает первый ключ a_i из условий $0 < a_i < \varphi(r_i)$, НОД$(a_i\ \varphi(r_i)) = 1$ и, наконец, вычисляет второй ключ a_i из уравнения $a_i a_i \equiv 1 \pmod{\varphi(r_i)}$. Записи в книге паролей будут иметь вид:

W_1: r_1, a_1
W_2: r_2, a_2
...
W_n: r_n, a_n

Если абонент W_1 решает отправить секретное письмо m W_2, то ему следует проделать следующую последовательность операций:

1) Если $m > \min(r_1,r_2)$, то m разбивается на части, каждая из которых меньше меньшего из чисел r_1 и r_2;

2) Если $r_1 < r_2$, то сообщение m сначала шифруется ключом α_1 ($m_1 \equiv m^{\alpha 1} \pmod{r_1}$), а затем - ключом α_2 ($m_2 \equiv m_1^{\alpha 2} \pmod{r_2}$), если же $r_1 > r_2$, то сообщение m сначала шифруется ключом α_2 ($m_1 \equiv m^{\alpha 2} \pmod{r_2}$), а затем - ключом α_1 ($m_2 \equiv m_1^{\alpha 1} \pmod{r_1}$);

3) Шифрованное сообщение m_2 отправляется W_2.

W_2 для дешифровки сообщения m_2 должен знать, кто его отправил, поэтому к m_2 должна быть добавлена электронная подпись, указывающая на W_1. Если $r_1 < r_2$, то для расшифровки m_2 сначала применяется ключ α_2, а затем - α_1, если же $r_1 > r_2$, то для расшифровки m_2 сначала применяется ключ α_2, а затем - α_1. Рассмотрим случай $r_1 < r_2$: $m_2^{\alpha 2} \equiv m_1^{\alpha 2 \alpha 2} \equiv m_1 \pmod{r_2}$ и $m_1^{\alpha 1} \equiv m^{\alpha 1 \alpha 1} \equiv m \pmod{r_1}$ по теореме Эйлера-Ферма.

Пример 9.14. Пусть W_1 выбрал и вычислил следующие числа $p_{11} = 7$, $p_{12} = 13$, $r_1 = p_{11}p_{12} = 91$, $\varphi(91) = 72$, $a_i = 5$, $\alpha_1 = 29$, а W_2 - следующие $p_{21} = 11$, $p_{22} = 23$, $r_2 = 253$, $\varphi(253) = 220$, $\alpha_2 = 31$, $\alpha_2 = 71$. После занесения записей о W_1 и W_2 в открытую книгу паролей, W_2 решает послать сообщение m = 41 для W_1. Т.к. $r_2 > r_1$, то сообщение сначала шифруется ключом a_i, а затем ключом α_2: $m \equiv 41^5 \equiv 6 \pmod{91}$, $m_2 \equiv 6^{71} \equiv 94 \pmod{253}$. Сообщение m_2 отправляется W_1. Получив $m_2 \equiv 94$, W_1, зная, что оно пришло от W_2, дешифрует его сначала ключом α_2, а затем ключом α_1: $94^{31} \pmod{253} \equiv 6$, $6^{29} \pmod{91} \equiv 41$.

Если подписать сообщение открытым образом, например, именем отправителя, то такая “подпись” будет ничем не защищена от подделки. Защита электронной подписи обычно реализуется с использованием таких же методов, что в криптосистеме с открытым ключом.

Электронная подпись генерируется отправителем по передаваемому сообщению и секретному ключу. Получатель сообщения может проверить его аутентичность по прилагаемой к нему электронной подписи и открытому ключу отправителя.

Стандартные системы электронной подписи считаются настолько надежными, что электронная подпись юридически приравнена к рукописной. Электронная подпись часто используется с открытыми, незашифрованными электронными документами.

В заключение следует отметить, что прежде чем подписать документ его «сжимают» до нескольких десятков или сотен байт с помощью так называемой хеш-функции. Здесь термин «сжатие» вовсе не аналогичен термину «архивация», значение хеш-функции лишь только сложным образом зависит от документа, но не позволяет восстановить сам документ. Эта хеш-функция должна удовлетворять ряду условий:

–быть чувствительна к всевозможным изменениям в тексте, таким, как вставки, выбросы, перестановки и т.п.;

–обладать свойством необратимости, т.е. задача подбора документа, который обладал бы требуемым значением хеш-функции, вычислительно неразрешима.

Вероятность того, что значения хеш-функций двух различных документов (вне зависимости от их длин) совпадут, должна быть ничтожна мала.

Далее к полученному значению хеш-функции применяют то или иное математическое преобразование (в зависимости от выбранного алгоритма ЭЦП) и получают собственно подпись документа. Эта подпись может иметь вполне читаемый, «буквенный» вид, но зачастую ее представляют в виде последовательности произвольных «нечитаемых» символов. ЭЦП может храниться вместе с документом, например стоять в его начале или конце, либо в отдельном файле. Естественно, в последнем случае при проверке подписи необходимо располагать как самим документом, так и файлом, содержащим его подпись.

При проверке подписи проверяющий должен располагать открытым ключом абонента, поставившего подпись. Этот ключ должен быть аутентифицирован, то есть проверяющий должен быть полностью уверен, что данный открытый ключ соответствует тому абоненту, который выдает себя за его «хозяина». В случае, когда абоненты самостоятельно обмениваются ключами, эта уверенность может подкрепляться связью по телефону, личным контактом или любым другим способом. В случае, когда абоненты действуют в сети с выделенным центром, открытые ключи абонентов подписываются (сертифицируются) центром, и непосредственный контакт абонентов между собой (при передаче или подтверждении подлинности ключей) заменяется на контакт каждого из них в отдельности с центром.

Процедура проверки ЭЦП состоит из двух этапов вычисления хеш-функции документа и собственно математических вычислений, предусмотренных в данном алгоритме подписи. Последние заключаются в проверке того или иного соотношения, связывающего хеш-функцию документа, подпись под этим документом и открытый ключ подписавшего абонента. Если рассматриваемое соотношение оказывается выполненным, то подпись признается правильной, а сам документ – подлинным, в противоположном случае документ считается измененным, а подпись под ним – недействительной.

9.9. Построение и использование хеш-функций

Под термином хеш-фунция понимается функция, отображающая электронные сообщения произвольной длины (иногда длина сообщения ограничена, но достаточно большим числом), в значения фиксированной длины. Последние часто называют хеш-кодами. Таким образом, у всякой хеш-функции h имеется большое количество коллизий, т.е. пар значений x и y таких, что $h(x)=h(y)$. Основное требование, предъявляемое криптографическими приложениями к хеш-функциям, состоит в отсутствии эффективных алгоритмов поиска коллизий. Хеш-функция, обладающая таким свойством, называется хеш-функцией, свободной от коллизий. Кроме того, хеш-функция должна быть односторонней, т.е. функцией, по значению которой вычислительно трудно найти ее аргумент, в то же время, функцией, для аргумента которой вычислительно трудно найти другой аргумент, который давал бы то же самое значение функции [16].

Схемы электронной цифровой подписи – основная сфера применения хеш-функций. Поскольку используемые на практике схемы электронной подписи не приспособлены для подписания сообщений произвольной длины, а процедура, состоящая в разбиении сообщения на блоки и генерации подписи для каждого блока по отдельности, крайне неэффективна, единственным разумным решением представляется применение схемы подписи к хеш-коду сообщения. Таким образом, хеш-функции вместе со схемами электронной цифровой подписи предназначены для решения задач обеспечения целостности и достоверности передаваемых и хранимых на носителях информации электронных сообщений. В прикладных информационных системах требуется применение так называемых криптографически стойких хеш-функций. Под термином «криптографически стойкая хэш-функция» понимается функция h, которая является односторонней и свободной от коллизий.

Введем следующие обозначения Хеш-функция h обозначается как $h(\alpha)$ и $h(\alpha,\beta)$ для одного и двух аргументов, соответственно. Хеш-код функции h обозначается как H. При этом $H_0 = 1$ обозначает начальное значение (вектор инициализации) хеш-функции. Под обозначением $\oplus$ будет пониматься операция сложения по модулю 2 или логическая операция XOR («Исключающая ИЛИ»). Результат шифрования блока B блочным шифром на ключе k обозначается $E_k(B)$.

Для лучшего понимания дальнейшего материала приведем небольшой пример построения хеш-функции

Предположим нам необходимо подписать при помощи заданного алгоритма электронной цифровой подписи достаточно длинное сообщение M. В качестве шифрующего преобразования в хеш-функции будет использоваться процедура шифра DES с ключом k. Тогда, чтобы получить хеш-код H сообще-

ния M при помощи хеш-функции h, необходимо выполнить следующую итеративную операцию:

$$H_i = E_{H_{i-1}}(M_i) \oplus M_i,$$

где $i = \overline{1,n}$; $H_0 = 1$; $M = M_1, M_2, ..., M_n$, сообщение M разбито на n 64-битных блока.

Хеш-кодом данной хеш-функции является значение $H = h(M,I) = H_n$.

Таким образом, на вход схемы электронной цифровой подписи вместо длинного сообщения M (как правило, несколько сотен или тысяч байтов) подается хэш-код H_n, длина которого ограничена длиной блока шифра DES, т.е. 64 битами. При этом в силу криптографической стой кости используемой хеш-функции практическая стойкость самой схемы подписи будет оставаться той же, что и при подписи сообщения M, в то время как эффективность всего процесса подписи электронного сообщения будет резко увеличена.

Были предприняты попытки построения хеш-функций на базе блочного шифра с размером хеш-кода в k раз (как правило, $k = 2$) большим, чем размер блока алгоритма шифрования.

В качестве примера можно привести хеш-функции MDC2 и MDC4 фирмы IBM. Данные хеш-функции используют блочный шифр (в оригинале DES) для получения хеш-кода, длина которого в 2 раза больше длины блока шифра. Алгоритм MDC2 работает несколько быстрее, чем MDC4, но представляется несколько менее стойким.

В качестве примера хеш-функций, построенных на основе вычислительно трудной математической задачи, можно привести функцию из рекомендаций МККТТ X.509.

Криптографическая стойкость данной функции основана на сложности решения следующей труднорешаемой теоретико-числовой задачи. Задача умножения двух больших (длиной в несколько сотен битов) простых чисел является простой с вычислительной точки зрения, в то время как факторизация (разложение на простые множители) полученного произведения является труднорешаемой задачей для указанных размерностей.

Следует отметить, что задача разложения числа на простые множители эквивалентна следующей труднорешаемой математической задаче. Пусть $n = pq$ произведение двух простых чисел p и q. В этом случае можно легко вычислить квадрат числа по модулю n: $x^2 (\mathrm{mod}\ n)$, однако вычислительно трудно извлечь квадратный корень по этому модулю.

Таким образом, хеш-функцию МККТТ X.509 можно записать следующим образом:

$$H_i = [(H_{i-1} \oplus M_i)^2] (\mathrm{mod}\ n),$$

где $i = \overline{1,n}$; $H_0 = 1$; $M = M_1, M_2, ..., M_n$.

Длина блока M_i представляется в октетах, каждый октет разбит пополам и к каждой половине спереди приписывается полуоктет, состоящий из двоичных единиц: n – произведение двух больших (512-битных) простых чисел p и q.

Ниже приведен числовой пример использования данной хеш-функции в его упрощенном варианте.

Пример 9.14. Получить хеш-код для сообщения «HASHING» при помощи хеш-функции X.509 с параметрами $p = 17$, $q = 19$.

Порядок вычисления хеш-кода:

а) получить значения модуля: $n = pq = 323$;

б) представить сообщение «HASHING» в виде символов ASCII:

H	A	S	H	I	N	G
72	65	83	72	73	78	71

в) представить коды ASCII битовой строкой:

72	65	83	72	73	78	71
01001000	01000001	01010011	01001000	01001001	01001110	01000111

г) разбить байт пополам (разбиение октета на полуоктеты), добавить в начало полубайта единицы и получить хешируемые блоки M_i:

M_1	M_2	M_3	M_4	M_5	M_6	M_7
11110100	11111000	11110100	11110001	11110101	11110011	11110100

M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}
11111000	11110100	11111001	11110100	11111110	11110100	11110111

д) выполнить итеративные шаги:

первая итерация:

M_1 = 11110100

$\oplus$

$H_0 = I$ = $\underline{00000000}$

$H_0 \oplus M_1$ = $11110100_2 = 244_{10}$

$[(H_0 \oplus M_1)]^2 (\mathrm{mod}\ 323)$ = $244^2 (\mathrm{mod} 323) = 104$

H_1 = $104_{10} = 01101000_2$

вторая итерация:

M_2 = 11111000

$\oplus$

H_1 = $\underline{01101000}$

$H_1 \oplus M_2$ = $10010000_2 = 144_{10}$

$[(H_1 \oplus M_2)]^2 (\mathrm{mod}\ 323) = I44^2 (\mathrm{mod} 323) = 64$

H_2 $= 64_{10} = 01000000_2$

…… … …….

i-я итерация...

Пример легко продолжить самостоятельно.

Пример 9.15 (упрощенный вариант). Хешируемое слово «ДВА». Коэффициенты $p = 7$, $q = 3$. Вектор инициализации $H_0 = 1$ выберем равным 6 (выбирается случайным образом). Определим $n = pq = 7 \cdot 3 = 21$. Слово «ДВА» в числовом эквиваленте можно представить как 531 (по номеру буквы в алфавите). Тогда хеш-код сообщения 531 получается следующим образом:

первая итерация:

$M_1 + H_0 = 5 + 6 = 11$; $[M_1 + H_0]^2 \pmod n = 11^2 \pmod{21} = 16 = H_1$;

вторая итерация:

$M_2 + H_1 = 3 + 16 = 19$; $[M_2 + H_1]^2 \pmod n = 19^2 \pmod{21} = 4 = H_2$;

третья итерация:

$M_3 + H_2 = 1 + 4 = 5$; $[M_3 + H_2]^2 \pmod n = 5^2 \pmod{21} = 4 = H_3$.

В итоге получаем хеш-код сообщения «ДВА», равный 4.

9.10. ГОСТ 28147-89 – стандарт на шифрование данных.

В Республике Беларусь установлен единый алгоритм криптографического преобразования данных для систем обработки информации в сетях ЭВМ, отдельных вычислительных комплексах и ЭВМ, который определяется ГОСТ 28147-89.

Алгоритм криптографического преобразования данных предназначен для аппаратной или программной реализации, удовлетворяет криптографическим требованиям и не накладывает ограничений на степень секретности защищаемых сообщений (информации). Из-за сложности этого алгоритма здесь будут приведены только основные его концепции. Чтобы подробно изучить алгоритм криптографического преобразования, следует обратиться к ГОСТ 28147-89 . Приведенный ниже материал должен использоваться лишь как ознакомительный.

При описании алгоритма приняты следующие обозначения. Если L и R – это последовательности бит, то LR будет обозначать конкатенацию последовательностей L и R. Под конкатенацией последовательностей L и R понимается последовательность бит, размерность которой равна сумме размерностей L и R. В этой последовательности биты последовательности R следуют за битами последовательности L. Конкатенация битовых строк является ассоциативной, т.е. запись $ABCDE$ обозначает, что за битами последовательности A следуют биты последовательности B, затем C и т.д.

Символом (+) обозначается операция побитового сложения по модулю 2, символом [+] – операция сложения по модулю 2^{32} двух 32-разрядных чисел. Числа суммируются по следующему правилу:

$$A[+]B = A + B, \text{ если } A + B < 2^{32};$$
$$A[+]B = A + B - 2^{32}, \text{ если } A + B \geq 2^{32}.$$

Символом {+} обозначается операция сложения по модулю $2^{32} - 1$ двух 32-разрядных чисел. Правила суммирования чисел следующие:

$$A\{+\}B = A + B, \text{ если } A + B < 2^{32} - 1;$$
$$A\{+\}B = A + B - (2^{32} - 1), \text{ если } A + B \geq 2^{32} - 1.$$

Алгоритм криптографического преобразования предусматривает несколько режимов работы. Но в любом случае для шифрования данных используется ключ, который имеет размерность 256 бит и представляется в виде восьми 32-разрядных чисел $X(i)$. Если обозначить ключ через W, то

$$W = X(7)X(6)X(5)X(4)X(3)X(2)X(1)X(0).$$

Расшифрование выполняется по тому же ключу, что и зашифрование, но этот процесс является инверсией процесса зашифрования данных.

Первый и самый простой режим –замена. Открытые данные, подлежащие зашифрованию, разбивают на блоки по 64 бит в каждом, которые можно обозначить $T(j)$. Очередная последовательность бит $T(j)$ разделяется на две последовательности $B(0)$ (левые или старшие биты) и $A(0)$ (правые или младшие биты), каждая из которых содержит 32 бита. Затем выполняется итеративный процесс шифрования, который описывается следующими формулами:

при $i = 1, 2, ..., 24; \quad j = (i - 1)(\text{mod } 8)$

$$A(i) = f(A(i-1)[+]X(j)(+)B(i-1));$$
$$B(i) = A(i-1);$$

при $i = 25, 26, ..., 31; \quad j = 32 - i$

$$A(i) = f(A(i-1)[+]X(j)(+)B(i-1));$$
$$B(i) = A(i-1);$$

при $i = 32$

$$A(32) = A(31);$$
$$B(32) = f(A(31)[+]X(0)(+)B(31)),$$

где i обозначает номер итерации (i = 1, 2,…, 32).

Функция f называется функцией шифрования. Ее аргументом является сумма по модулю 2^{32} числа $A(i)$, полученного на предыдущем шаге итерации, и числа $X(j)$ ключа (размерность каждого из этих чисел равна 32 знакам).

Функция шифрования включает две операции над полученной 32-разрядной суммой. Первая операция называется подстановкой K. Блок подстановки K состоит из восьми узлов замены $K(1)...K(8)$ с памятью 64 бит каждый. Поступающий на блок подстановки 32-разрядный вектор разбивается на восемь последовательно идущих 4-разрядных векторов, каждый из которых преобразуется в 4-разрядный вектор соответствующим узлом замены, представляющим собой таблицу из шестнадцати целых чисел в диапазоне 0...15.

Входной вектор определяет адрес строки в таблице, число из которой является выходным вектором. Затем 4-разрядные выходные векторы последовательно объединяются в 32-разрядный вектор. Таблицы блока подстановки K содержат ключевые элементы, общие для сети ЭВМ и редко изменяемые.

Вторая операция – циклический сдвиг влево 32-разрядного вектора, полученного в результате подстановки K 64-разрядный блок зашифрованных данных $T_{ш}$ представляется в виде:

$$T_{ш} = A(32)B(32).$$

Остальные блоки открытых данных в режиме простой замены зашифровываются аналогично.

Следует иметь в виду, что режим простой замены допустимо использовать для шифрования данных только в ограниченных случаях.

К этим случаям относится выработка ключа и зашифрование его с обеспечением имитозащиты для передачи по каналам связи или хранения в памяти ЭВМ.

Следующий режим шифрования называется режимом гаммирования. Открытые данные, разбитые на 64-разрядные блоки $T(i)$ (i =1, 2,..., m, где m определяется объемом шифруемых данных), зашифровываются в режиме гаммирования путем поразрядного сложения по модулю 2 с гаммой шифра $Г_{ш}$, которая вырабатывается блоками по 64 бит, т.е.

$$Г_{ш} = (Г(1), Г(2), ..., Г(i), ..., Г(m)).$$

Число двоичных разрядов в блоке $T(m)$ может быть меньше 64, при этом неиспользованная для шифрования часть гаммы шифра из блока $Г(m)$ отбрасывается.

Уравнение зашифрования данных в режиме гаммирования может быть представлено в следующем виде:

$$Ш(i) = A(Y(i-1)[+]C_2, Z(i-1)\{+\}C_1(+)T(i)) = Г(i)(+)T(i).$$

В этом уравнении $Ш(i)$ обозначает 64-разрядный блок зашифрованного текста; $A(-)$ – функцию шифрования в режиме простой замены (аргументами

этой функции являются два 32-разрядных числа); C_1 и C_2 – константы, заданные в обязательном приложении 2 к ГОСТ 28147-89. Величины $У(\text{-})$ и $Z(\bullet)$ определяются итерационно по мере формирования гаммы следующим образом $(Y(0), Z(0)) = A(S)$, где S – 64-разрядная двоичная последовательность (синхропосылка)

$$(Y(i), Z(i)) = Y(i-1)[+]C_2, Z(i-1)\{+\}C_1,$$

где $i = 1, 2, \ldots, m$.

Расшифрование данных возможно только при наличии синхропосылки, которая не является секретным элементом шифра и может храниться в памяти ЭВМ или передаваться по каналам связи вместе с зашифрованными данными.

Режим гаммирования с обратной связью очень похож на режим гаммирования Как и в режиме гаммирования, открытые данные, разбитые на 64-разрядные блоки $T(i)$ (i = 1, 2,..., m, где m определяется объемом шифруемых данных), зашифровываются путем поразрядного

$$Г_{ш} = (Г(1), Г(2), ..., Г(i), ..., Г(m)).$$

сложения по модулю 2 с гаммой шифра $Г_{ш}$, которая вырабатывается блоками по 64 бит.

Число двоичных разрядов в блоке $T(m)$ может быть меньше 64, при этом неиспользованная для шифрования часть гаммы шифра из блока $Г(m)$ отбрасывается.

Уравнение зашифрования данных в режиме гаммирования с обратной связью для г = 2,3,..., то может быть представлено в следующем виде:

$$Ш(1) = A(S)(+)T(1) = Г(1)(+)T(1);$$
$$Ш(i) = A(Ш(i-1)(+)T(i)) = Г(i)(+)T(i),$$

где $Ш(i)$ обозначает 64-разрядный блок зашифрованного текста; $A(\text{-})$ – функцию шифрования в режиме простой замены.

Аргументом функции на первом шаге итеративного алгоритма является 64-разрядная синхропосылка, а на всех последующих – предыдущий блок зашифрованных данных $Ш(i - 1)$.

В ГОСТ 28147-89 определяется процесс выработки имитовставки, который единообразен для любого из режимов шифрования данных. Имитовставка – это блок из p бит (имитоставка $И_{р}$), который вырабатывается либо перед шифрованием всего сообщения, либо параллельно с шифрованием по блокам. Первые блоки открытых данных, которые участвуют в выработке имитовставки, могут содержать служебную информацию (например адресную часть, время, синхропосылку) и не зашифровываться. Значение параметра p (число дво-

ичных разрядов в имитовставке) определяется криптографическими требованиями с учетом того, что вероятность навязывания ложных помех равна $1/2^p$.

Для получения имитовставки открытые данные представляются в виде 64-разрядных блоков $T(i)$ (i = 1, 2 m, где m определяется объемом шифруемых данных). Первый блок открытых данных $T(1)$ подвергается преобразованию, соответствующему первым 16 циклам алгоритма зашифрования в режиме простой замены, причем в качестве ключа для выработки имитовставки используется ключ, по которому шифруются данные

Полученное после 16 циклов работы 64-разрядное число суммируется по модулю 2 с вторым блоком открытых данных $T(2)$. Результат суммирования снова подвергается преобразованию, соответствующему первым 16 циклам алгоритма зашифрования в режиме простой замены.

Полученное 64-разрядное число суммируется по модулю 2 с третьим блоком открытых данных $T(3)$ и т д. Последний блок $T(m)$, при необходимости дополненный до полного 64-разрядного блока нулями, суммируется по модулю 2 с результатом работы на шаге m-1, после него зашифровывается в режиме простой замены по первым 16 циклам работы алгоритма. Из полученного 64-разрядного числа выбирается отрезок $И_р$ длиной p бит.

Имитовставка $И_р$ передается по каналу связи или в память ЭВМ после зашифрованных данных. Поступившие зашифрованные данные расшифровываются и из полученных блоков открытых данных $T(i)$ вырабатывается имитовставка $И_р$, которая затем сравнивается с имитовставкой $И_р$, полученной из канала связи или из памяти ЭВМ В случае несовпадения имитовставок все расшифрованные данные считают ложными.

9.11. Некоторая сравнительная оценка криптографических методов

Результаты сравнительных оценок криптографических методов приведены в [12].

Метод шифрования с использованием датчика ПСЧ наиболее часто используется в программной реализации системы криптографической защиты данных. Это объясняется тем, что, с одной стороны, он достаточно прост для программирования, а с другой стороны, позволяет создавать алгоритмы с очень высокой криптостойкостью. Кроме того, эффективность данного метода шифрования достаточна высока. Системы, основанные на методе шифрования с использованием датчика ПСЧ, позволяют зашифровать в секунду от нескольких десятков до сотен килобайт данных (здесь оценочные характеристики приведены для персональных компьютеров).

Основным преимуществом метода DES является то, что он является стандартом. Как утверждает Национальное Бюро Стандартов США, алгоритм обладает следующими свойствами:

–высоким уровнем защиты данных против дешифрования и возможной модификации данных;

– простотой в понимании;

–высокой степенью сложности, которая делает его раскрытие дороже получаемой при этом прибыли;

– метод защиты основывается на ключе и не зависит ни от какой «секретности» алгоритма;

–экономичен в реализации и эффективен в быстродействии.

Важной характеристикой этого алгоритма является его гибкость при реализации и использовании в различных приложениях обработки данных. Каждый блок данных шифруется независимо от других, что позволяет расшифровывать отдельный блок в зашифрованном сообщении и структуре данных. Поэтому можно осуществлять независимую передачу блоков данных и произвольный доступ к зашифрованным данным. Ни временная, ни позиционная синхронизация для операция шифрования не нужны.

Алгоритм вырабатывает зашифрованные данные, в которых каждый бит является функцией от всех битов открытых данных и всех битов ключа. Различие лишь в одном бите данных дает в результате равные вероятности изменения для каждого бита зашифрованных данных.

Конечно, эти свойства DES выгодно отличают его от метода шифрования с использованием датчика ПСЧ, поскольку большинство алгоритмов шифрования, построенных на основе датчиков ПСЧ, не характеризуются всеми преимуществами DES. Однако и DES обладает рядом недостатков

Самым существенным недостатком DES специалисты признают размер ключа, который считается слишком малым. Стандарт не является неуязвимым, хотя и очень труден для раскрытия. Для дешифрования сообщения методом подбора ключей достаточно выполнить 2^{56} операций расшифрования (т.е. всего около $7{,}6 \cdot 10^{16}$ операций). Хотя в настоящее время нет аппаратуры, которая могла бы выполнить в обозримый период времени подобные вычисления, никто не гарантирует, что она не появится в будущем. Некоторые специалисты предлагают простую модификацию для устранения этого недостатка исходный текст зашифровывается сначала по ключу K_1, а затем по ключу K_2 и, наконец по ключу K_3. В результате время, требующееся для дешифрования, возрастает до 2^{168} операций (приблизительно, до 10^{34} операций).

Еще один недостаток метода DES заключается в том, что отдельные блоки, содержащие одинаковые данные (например пробелы), будут одинаково выглядеть в зашифрованном тексте, что с точки зрения криптоанализа неправильно. Метод DES может быть реализован и программно. В зависимости от быстродействия и типа процессора персонального компьютера программная система, шифрующая данные с использованием метода DES, может обрабатывать от нескольких килобайт до десятков килобайт данных в секунду. В то же время необходимо отметить, что базовый алгоритм все же рассчитан на реализацию в электронных устройствах специального назначения.

Алгоритм криптографического преобразования, определяемый ГОСТ 28147-89, свободен от недостатков стандарта DES и в то же время обладает всеми его преимуществами. Кроме того, в стандарт уже заложен метод, с помощью которого можно зафиксировать необнаруженную случайную или умышленную модификацию зашифрованной информации.

Однако у алгоритма есть очень существенный недостаток, который заключается в том, что его программная реализация очень сложна и практически лишена всякого смысла из-за крайне низкого быстродействия. По оценкам авторов, за 1 с на персональном компьютере может быть обработано всего лишь несколько десятков (максимально сотен) байт данных, а подобная производительность вряд ли удовлетворит кого-либо из пользователей. Хотя сейчас уже разработаны аппаратные средства, реализующие данный алгоритм криптографического преобразования данных, которые демонстрируют приемлемую производительность (около 70 Кбайт/с для IBM PC с тактовой частотой 12 МГц).

Теперь о методе RSA. Он является очень перспективным, поскольку для зашифрования информации не требуется передачи ключа другим пользователям. Это выгодно отличает его от всех вышеописанных методов криптографической защиты данных. Но в настоящее время к этому методу относятся вероятностно-сомнительно, поскольку в ходе дальнейшего развития может быть найден эффективный алгоритм определения делителей целых чисел, в результате чего метод шифрования станет абсолютно незащищенным.

В остальном метод RSA обладает только достоинствами. К числу этих достоинств следует отнести очень высокую криптостойкость, довольно простую программную и аппаратную реализации. Правда, использование этого метода для криптографической защиты данных неразрывно связано с высоким уровнем развития вычислительной техники.

Кроме метода RSA есть еще несколько криптосистем с открытым ключом, в той или иной мере распространенных в теоретическом или практическом плане, например система Эль-Гамаля, основанная на трудности вычисления дискретных логарифмов в конечных полях. Мак-Элис предложил криптосистему, основанную на кодах, исправляющих ошибки. Вычисления в этой системе реализуются в несколько раз быстрее, чем в системе RSA.

9.12. Закрытие речевых сигналов в телефонных каналах

Главной целью при разработке систем передачи речи является сохранение тех ее характеристик, которые наиболее важны для восприятия слушателем.

Безопасность связи при передаче речевых сообщений основывается на использовании большого числа различных методов закрытия сообщений, меняющих характеристики речи таким образом, что она становится неразборчивой и неузнаваемой для подслушивающего лица, перехватившего закрытое речевое сообщение [16].

9.12.1. Основные методы и типы систем закрытия речевых сообщений.

В речевых системах связи известны два основных метода закрытия речевых сигналов, разделяющиеся по способу передачи по каналам связи: аналоговое скремблирование и дискретизация речи с последующим шифрованием. Под ***скремблированием*** понимается изменение характеристик речевого сигнала таким образом, чтобы полученный модулированный сигнал, обладая свойствами неразборчивости и неузнаваемости, занимал такую же полосу частот спектра, что и исходный открытый.

Каждый из этих двух методов имеет свои достоинства и недостатки. Так в первых двух системах, представленных на рисунке 9.14, а и б, в канале связи при передаче присутствуют кусочки исходного, открытого речевого сообщения, преобразованные в частотной и (или) временной областях. Это означает, что такие системы могут быть атакованы криптоаналитиком противника на уровне анализа звуковых сигналов.

Системы на рисунке 9.14, в и г не передают никакой части исходного речевого сигнала. Речевые компоненты кодируются в цифровой поток данных, который смешивается с псевдослучайной последовательностью, вырабатываемой ключевым генератором по одному из криптографических алгоритмов, и полученное таким образом закрытое речевое сообщение передается с помощью модема в канал связи, на приемном конце которого производятся обратные преобразования с целью получения открытого речевого сигнала [16].

Технология изготовления широкополосных систем закрытия речи по схеме, приведенной на рисунке 9.15, в, хорошо известна. Не представляет собой трудностей техническая реализация используемых для этих целей способов кодирования речи типа АДИКМ (адаптивной дифференциальной импульсно-кодовой модуляции), ДМ (дельта-модуляции) и т.п. Но представленная такими способами дискретизированная речь может передаваться лишь по специально выделенным широкополосным каналам связи с полосой пропускания 4,8...19,2 кГц и не пригодна для передачи по каналам телефонной сети общего пользования, полоса частот которых 3,1 кГц. В таких случаях используются узкополосные системы закрытия по схеме на рисунке 9.14, г, главной трудностью при реализации которых является высокая сложность алгоритмов сжатия речевых сигналов.

Посредством дискретного кодирования речи с последующим шифрованием достигается высокая степень закрытия, но в прошлом этот метод не находил широкого распространения в стандартных телефонных каналах вследствие низкого качества восстановления передаваемой речи. Последние достижения в развитии низкоскоростных дискретных кодеков позволили значительно улучшить качество восстановленной речи без снижения надежности закрытия.

Следует отметить, что уровень или степень секретности систем закрытия речи понятие весьма условное. Однако основные уровни защиты принято разделять на тактический (или низкий) и стратегический (или высокий), что в некотором смысле перекликается с понятиями практической и теоретической стойкости криптографических систем закрытия данных. Практический уровень

обеспечивает защиту информации от подслушивания посторонними лицами на период времени, измеряемый минутами или днями (большое число простых методов способны обеспечить такой уровень защиты при приемлемой стоимости). Стратегический уровень защиты информации от перехвата подразумевает, что высоковалифицированному, технически хорошо оснащенному специалисту, для дешифрования перехваченного сообщения потребуется период времени от нескольких месяцев до многих лет.

Часто используется и понятие средней степени защиты, занимающее промежуточное положение между тактическим и стратегическим уровнями закрытия.

По литературным данным составлена сравнительная диаграмма (рисунок 9.16), показывающая связь между различными методами закрытия речевых сигналов, степенью секретности и качеством восстановленной речи.

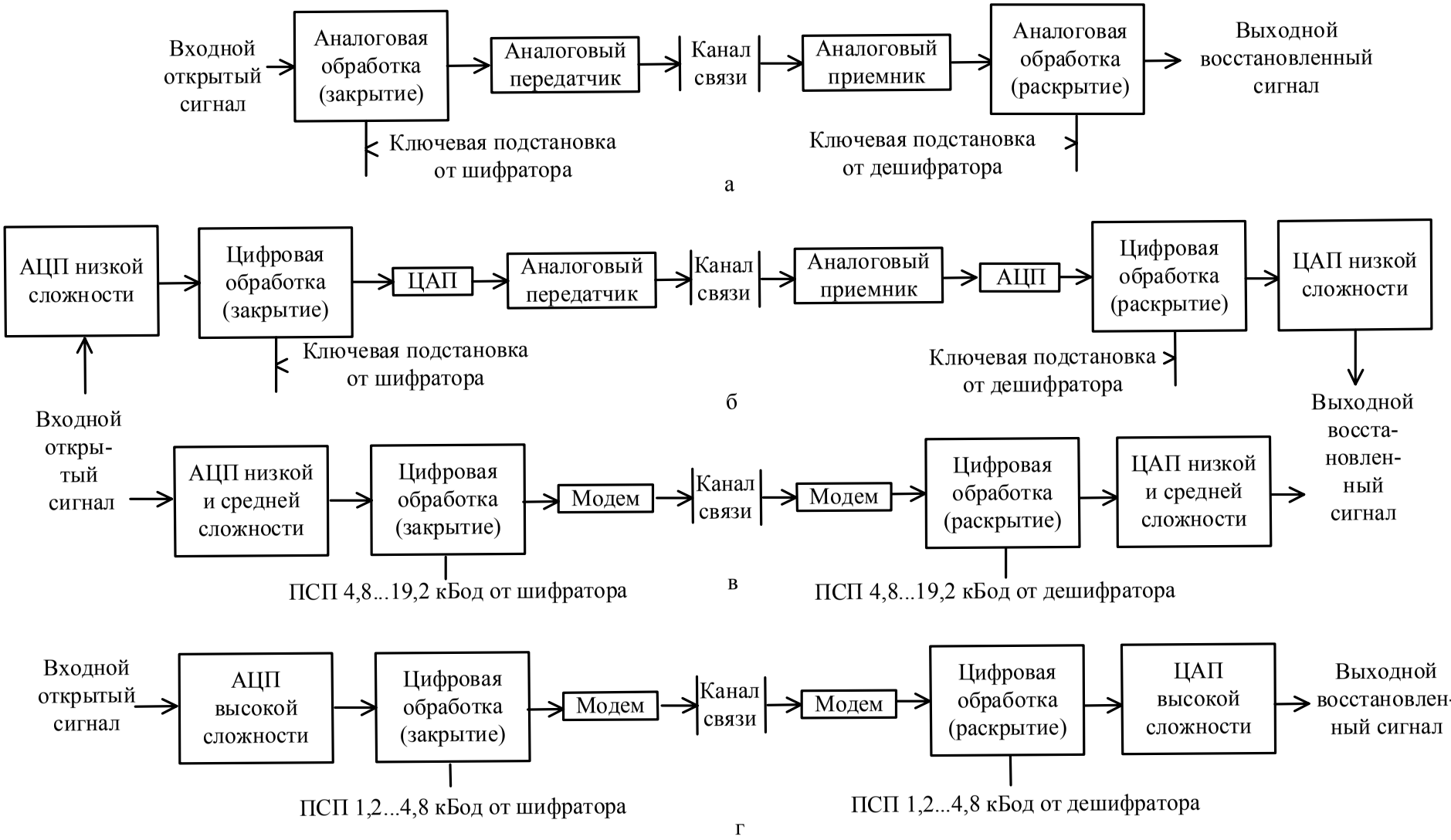


Рисунок 9.4 - . Виды систем закрытия речи:

а - аналоговые скремблеры на базе простейших временных и/или частотных перестановок отрезков речи; б - аналоговые комбинированные речевые скремблеры на основе частотно-временных перестановок отрезков речи, представленных дискретными отсчетами с применением цифровой обработки сигналов; в, г - широкополосные и узкополосные цифровые системы закрытия речи; АЦП - аналого-цифровое преобразование; ЦАП - цифро-аналоговое преобразование; ПСП - псевдослучайная последовательность

Понятие «качество восстановленного сигнала (речи)», используемое на диаграмме, весьма условно. Под ним, как правило, понимают узнаваемость абонента и разборчивость принимаемого речевого сигнала [16].

Рисунок 9.15 - Основные характеристики систем закрытия речи

9.12.2. Аналоговое скремблированиие

Наибольшая часть аппаратуры засекречивания речевых сигналов использует в настоящее время метод аналогового скремблирования, поскольку:

–это дешево;

–необходимая для этого аппаратура применяется в большинстве случаев в стандартных телефонных каналах с полосой 3,1 кГц;

–обеспечивается коммерческое качество дешифрованной речи;

–гарантируется достаточно высокая стойкость закрытия.

Аналоговые скремблеры преобразуют исходный речевой сигнал посредством изменения его амплитудных, частотных и временных параметров в различных комбинациях. Скремблированный сигнал затем может быть передан по каналу связи в той же полосе частот, что и исходный, открытый. В аппаратах такого типа используется один или несколько способов аналогового скремблирования из числа следующих:

–скремблирование в частотной области – частотная инверсия (пре образование спектра сигнала с помощью гетеродина и фильтра), частотная инверсия

и смещение (частотная инверсия с меняющимся скачкообразно смещением несущей частоты), разделение полосы частот речевого сигнала на ряд поддиапазонов с последующей их перестановкой и инверсией;

–скремблирование по временной области – разбиение блоков или частей речи на сегменты с перемешиванием их во времени с последующим их прямым и (или) реверсивным считыванием;

–комбинация временного и частотного; скремблирования.

Как правило, все перестановки каким-либо образом выделенных сегментов или участков речи во временной и (или) в частотной областях осуществляются по закону псевдослучайной последовательности, вырабатываемой шифратором по ключу, меняющемуся от одного сообщения к другому.

На стороне приемника выполняется дешифрование цифровых кодов, полученных из канала связи, и преобразование в аналоговую форму. Системы, работа которых основана на таком методе, являются достаточно сложными, поскольку для обеспечения высокого качества передаваемой речи требуется высокая частота дискретизации входного аналогового сигнала и соответственно высокая скорость передачи данных по каналу связи. Каналы связи, которые обеспечивают скорость передачи данных только 2400 Бод, называются узкополосными, в то время, как другие, обеспечивающие скорость передачи свыше 2400 Бод, относят к широкополосным. По этому же принципу можно разделять и устройства дискретизации речи с последующим шифрованием.

Несмотря на всю свою сложность, аппаратура данного типа представлена на коммерческом рынке рядом моделей, большинство из которых передает данные по каналу связи со скоростями модуляции от 2,4 до 19,2 кбит/с, обеспечивая при этом несколько худшее качество воспроизведения речи по сравнению с обычным телефоном. Основным же преимуществом таких цифровых систем кодирования и шифрования остается высокая степень закрытия речи, получаемая посредством использования широкого набора криптографических методов, применяемых для защиты передачи данных по каналам связи.

Методы речевого скремблирования впервые появились во время второй мировой войны. Среди последних достижений в этой области следует отметить широкое использование интегральных микросхем, микро процессоров и цифровых процессоров обработки сигналов (ЦПОС). Все это обеспечило высокую надежность устройств закрытия речи с уменьшением их размера и стоимости.

Аналоговым скремблерам удалось достичь определенного уровня развития, обеспечивающего среднюю и даже высокую степень защиты речевых сообщений. Поскольку скремблированные речевые сигналы в аналоговой форме лежат в той же полосе частот, что и исходные от крытые, это означает, что их можно передавать по обычным коммерческим каналам связи, используемым для передачи речи, без затребования какого-либо специального оборудования, например модемов. Поэтому устройства речевого скремблирования не так дороги и значительно менее сложны, чем устройства дискретизации с последующим цифровым шифрованием.

Аналоговые скремблеры по режиму работы можно разделить на два следующих класса:

–статические, схема кодирования которых остается неизменной в течение всей передачи речевого сообщения;

–динамические, постоянно генерирующие кодовые подстановки в ходе передачи (код может быть изменен в процессе передачи несколько раз в течение каждой секунды).

Очевидно, что динамические скремблеры обеспечивают более высокую степень защиты, поскольку резко ограничивают возможность лег кого прослушивания переговоров посторонними лицами.

Преобразование речевого сигнала возможно по трем параметрам: амплитуде, частоте и времени. Считается, что использовать амплитуду нецелесообразно, так как изменяющиеся во времени затухание канала и отношение сигнал/шум делают сложным точное восстановление амплитуды переданного сигнала. Поэтому практическое применение получило только частотное и временное скремблирование и их комбинации.

Существуют два основных вида частотных скремблеров: инверсный и полосовой. Оба основаны на преобразованиях спектра исходного речевого сигнала для скрытия передаваемой информации и восстановления полученного речевого сообщения путем обратных преобразований. Инверсный скремблер осуществляет преобразование речевого спектра, равносильное повороту частотной полосы речевого сигнала вокруг не которой средней точки (рисунок 9.16). Однако данный способ обеспечивает невысокий уровень закрытия, так как при перехвате легко устанавливается значение частоты, соответствующее средней точке инверсии в полосе речевого сигнала.

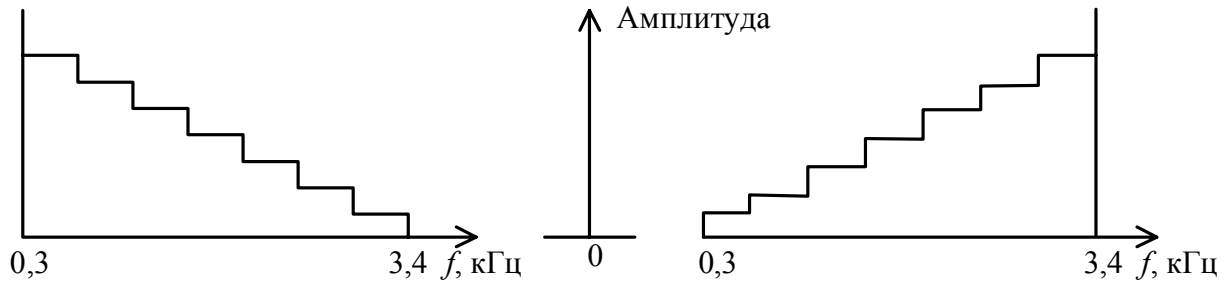


Рисунок 9.16 - Принцип работы инвертора спектра речи

Речевой спектр можно также разделить на несколько частотных полос и произвести их перемешивание и инверсию по некоторому правилу (ключу системы). Так функционирует полосовой скремблер (рисунок 9.17).

Изменение ключа системы позволяет повысить степень закрытия, но требует введения синхронизации на приемной стороне системы. Основная часть

энергии речевого сигнала сосредоточена в небольшой области низкочастотного спектра, поэтому выбор вариантов перемешивания ограничен.

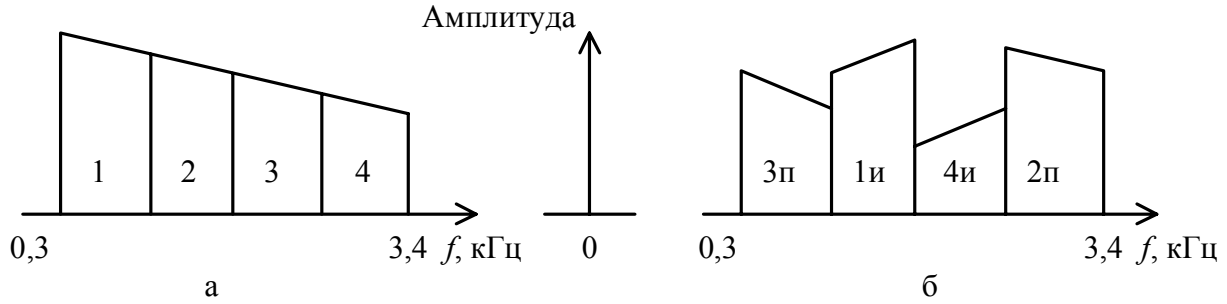


Рисунок 9.17 - Принцип работы четырехполосового скремблера речи:
а – исходный речевой спектр; б – измененный с помощью скремблера

Существенное повышение степени закрытия речи может быть достигнуто путем реализации в полосовом скремблере быстрого преобразования Фурье (БПФ). При этом число допустимых перемешиваний частотных полос значительно увеличивается, что обеспечивает высокую степень закрытия без ухудшения качества речи. Можно дополнительно повысить степень закрытия задержкой различных частотных компонент сигнала на различное время. Пример реализации такой системы показан на рисунке 9.18.

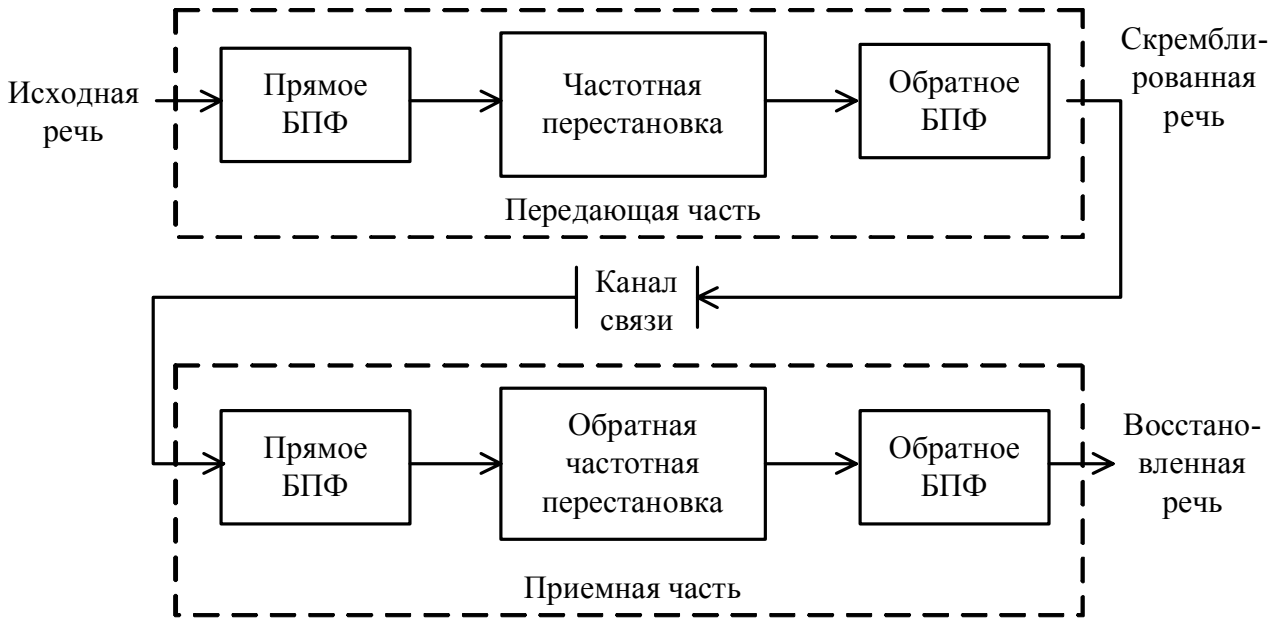


Рисунок 9.18 - Основная форма реализации аналогового скремблера речи на основе БПФ

Главным недостатком использования БПФ является возникновение в системе большой задержки сигнала (до 300 мс), обусловленной необходимостью использования весовых функций. Это приводит к затруднениям в работе дуплексных систем связи.

Временные скремблеры основаны на двух основных способах закрытия: инверсии по времени сегментов речи и их временной перестановке. По сравнению с частотными скремблерами задержка у временных скремблеров намного больше, но существуют различные методы ее уменьшения.

В скремблерах с временной инверсией речевой сигнал делится на последовательность временных сегментов и каждый из них передается инверсно во времени (с конца). Такие скремблеры обеспечивают ограниченный уровень закрытия, зависящий от длительности сегментов. Для достижения неразборчивости медленной речи необходимо, чтобы длина сегмента составляла около 250 мс. Это означает, что задержка си темы будет равна примерно 500 мс, что может оказаться неприемлемым в некоторых случаях.

Для повышения уровня закрытия прибегают к способу перестановки временных отрезков речевого сигнала в пределах фиксированного кадра (рисунок 9.19). Правило перестановок является ключом системы, изменением которого можно существенно повысить степень закрытия речи. Остаточная разборчивость зависит от длительностей отрезков сигнала и кадра и с увеличением последнего уменьшается.

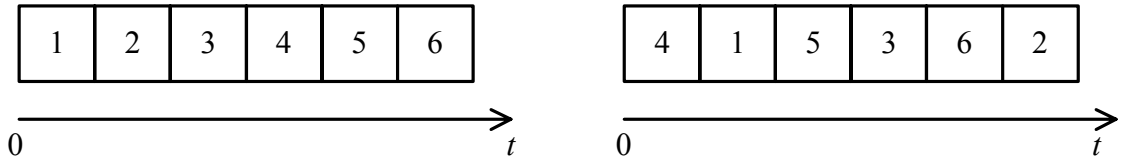


Рисунок 9.19 - Схема работы временного скремблера с перестановками в фиксированном кадре

Главным недостатком скремблера с фиксированным кадром является большое время задержки системы, равное удвоенной длительности кадра. Этот недостаток устраняется в скремблере с перестановкой временных отрезков речевого сигнала со скользящим окном. В нем число комбинаций возможных перестановок ограничено таким образом, что задержка любого отрезка не превосходит установленного максимального значения. Каждый отрезок исходного речевого сигнала как бы имеет временное окно, внутри которого он может занимать произвольное место при скремблировании. Это окно скользит во времени по мере поступления в него каждого нового отрезка сигнала. Задержка при этом снижается до длительности окна.

Используя комбинацию временного и частотного скремблирования, можно значительно повысить степень закрытия речи. Комбинированный скремблер намного сложнее обычного и требует компромиссного решения по выбору уровня закрытия, остаточной разборчивости, времени задержки, сложности системы и степени искажений в восстановленном сигнале. В качестве примера такой системы рассмотрим скремблер, схема которого представлена на рисунке 9.20, где операция частотно временных перестановок дискретизированных отрезков речевого сигнала осуществляется при помощи четырех процессоров

цифровой обработки сигналов, один из которых может реализовывать функцию генератора случайной последовательности (ключа системы закрытия).

В таком скремблере спектр оцифрованного аналого-цифровым преобразователем АЦП речевого сигнала разбивается посредством использования алгоритмов цифровой обработки сигналов на частотно-временные элементы, которые затем перемешиваются на частотно-временной плоскости в соответствии с одним из криптографических алгоритмов (рисунок 9.21) и суммируются, не выходя за пределы частотного диапазона исходного сигнала.

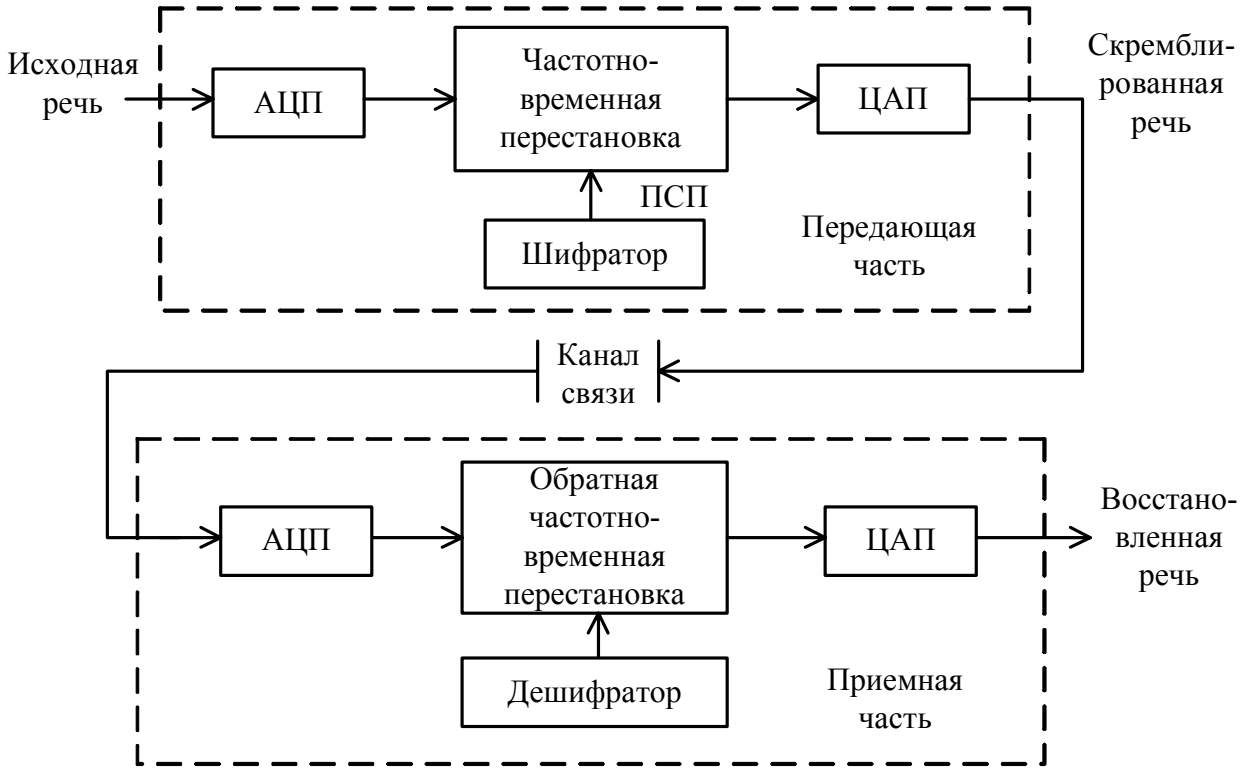


Рисунок 9.20 - Структурная схема комбинированного скремблера:
ПСП – псевдослучайная последовательность

Число частотных полос спектра, в которых производятся перестановки с возможной инверсией спектра, равно четырем Максимальная задержка частотно-временного элемента во времени равна пяти. Полученный таким образом закрытый сигнал при помощи ЦАП переводится в аналоговую форму и подается в канал связи. На приемном конце производятся обратные операции по восстановлению полученного закрытого речевого сообщения. Стойкость представленного алгоритма сравнима со стойкостью систем цифрового закрытия речи.

Скремблеры всех типов, за исключением простейшего (с частотной инверсией), вносят искажения в восстановленный речевой сигнал. Границы временных сегментов нарушают целостность сигнала, что неизбежно приводит к появлению внеполосных составляющих. Нежелательное влияние оказывают и

групповые задержки составляющих речевого сигнала в канале связи. Результатом искажения является увеличение минимально допустимого отношения сигнал/шум, при котором может осуществляться надежная связь.

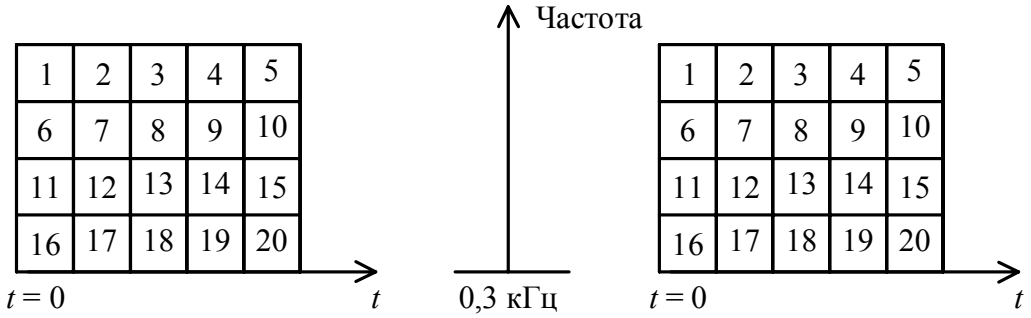


Рисунок 9.21 - Принцип работы комбинированного скремблера

Однако, несмотря на указанные проблемы, методы временного и частотного скремблирования, а также комбинированные методы успешно используются в коммерческих каналах связи для защиты конфиденциальной информации [16].

9.12.3. Дискретизация речи с последующим шифрованием

Альтернативным аналоговому скремблированию методом передачи речи в закрытом виде является шифрование речевых сигналов, преобразованных в цифровую форму, перед их передачей (см рисунок 9.14, в и г). Этот метод обеспечивает более высокий уровень закрытия по сравнению с описанными выше аналоговыми методами. В основе устройств, работающих по такому принципу, лежит представление речевого сигнала в виде цифровой последовательности, закрываемой по одному из криптографических алгоритмов. Передача данных, представляющих дискретизированные отсчеты речевого сигнала и его параметры, по телефонным сетям, как и в случае устройств шифрования алфавитно-цифровой и графической информации, осуществляется через устройства, называемые модемами. Основной целью при разработке устройств цифрового закрытия речи является сохранение тех ее характеристик, которые наиболее важны для восприятия слушателем.

Сохранение формы сигнала требует высокой скорости передачи и, соответственно, использования широкополосных каналов связи. Например, при импульсно-кодовой модуляции (ИКМ), используемой в большинстве телефонных сетей, необходима скорость передачи, равная 64 кбит/с. В случае применения адаптивной дифференциальной ИКМ она понижается до 32 кбит/с и ниже. Для узкополосных каналов, не обеспечивающих такие скорости передачи, требуются устройства, исключающие избыточность речи до ее передачи. Снижение ин формационной избыточности речи достигается параметризацией речевого сигнала, при которой характеристики речи, существенные для восприятия, сохраняются.

Основной особенностью использования систем цифрового закрытия речевых сигналов является необходимость использования модемов. В принципе возможны следующие подходы при проектировании систем цифрового закрытия речевых сигналов:

–цифровая последовательность параметров речи с выхода вокодерного устройства подается на вход шифратора, где подвергается преобразованию по одному из криптографических алгоритмов, затем поступает через модем в канал связи, на приемной стороне которого осуществляются обратные операции по восстановлению речевого сигнала, в которых задействованы модем и дешифратор (см. рисунок 9.14,г). Модем представляет собой отдельное устройство, обеспечивающее передачу данных по одному из протоколов, рекомендованных МККТТ. Шифрующие (дешифрующие) функции обеспечиваются либо в отдельных устройствах, либо в программно-аппаратной реализации самого вокодера;

–шифрующие (дешифрующие) функции обеспечиваются самим модемом (так называемый засекречивающий модем) обычно по известным. Криптографическим алгоритмам типа DES и др. Цифровой поток, несущий информацию о параметрах речи, с выхода вокодера непосредственно поступает на такой модем. Организация связи по каналу аналогична вышеприведенной.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Зачем необходимо криптографическое закрытие информации?
2. Что понимается под криптограммой?
3. Поясните метод замены.
4. Зашифруйте методом Важинера свою фамилию ключом ЗАЧЕТ.
5. Поясните методом шифрования с автоключом.
6. В чём состоит принцип гомофонической замены?
7. Зашифруйте своё имя шифром Плэйфера.
8. Поясните принцип шифрования перестановкой.
9. В чём сущность шифрования гоммированием.
10. Поясните стандарт шифрования данных DES.
11. Поясните стандарт шифрования данных ГОСТ 28147-89.
12. Поясните принцип работы криптографических систем с открытым ключом.
13. Укажите порядок подстановки цифровой подписи «Нотариус».
14. Получите хеш-код для сообщения MINSK при помощи хеш-функции с параметрами $p = 13$ и $q = 11$.
15. Назовите основные методы и типы систем закрытия речевых сообщений.
16. Поясните принцип аналогового и цифрового скремблирования, речи.

10. ИДЕНТИФИКАЦИЯ И АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ

Идентификацию и аутентификацию пользователей можно считать основой программно-технических средств безопасности. Идентификация позволяет субъекту (пользователю, процессу, действующему от имени определенных пользователей, или иному аппаратно-программному компоненту) назвать себя (сообщить свое имя). Посредством аутентификации вторая сторона убеждается, что субъект – действительно тот, за кого он себя выдает. Совокупность выполнений процедур идентификации и аутентификацию называют процедурой авторизации.

Реализация процедур авторизации пользователей является общей проблемой для любых технических систем, в которых требуется обеспечивать разграничение доступа к обрабатываемой информации. Так как функционирование всех механизмов разграничения доступа, использующих аппаратные или программные средства, основано на предположении, что любой пользователь системы представляет собой конкретное лицо, то должен существовать некоторый механизм его опознания, обеспечивающий установление подлинности данного пользователя, обращающегося к системе.

Существуют три класса опознания (рисунок 10.1, [20]), которые базируются:

- на условных, заранее присваиваемых признаках (сведениях), известных субъекту (что знает субъект);
- на физических средствах, действующих аналогично физическому ключу (что имеет субъект);
- на индивидуальных характеристиках субъекта, его физических данных, позволяющих выделить его среди других лиц (что присуще субъекту).

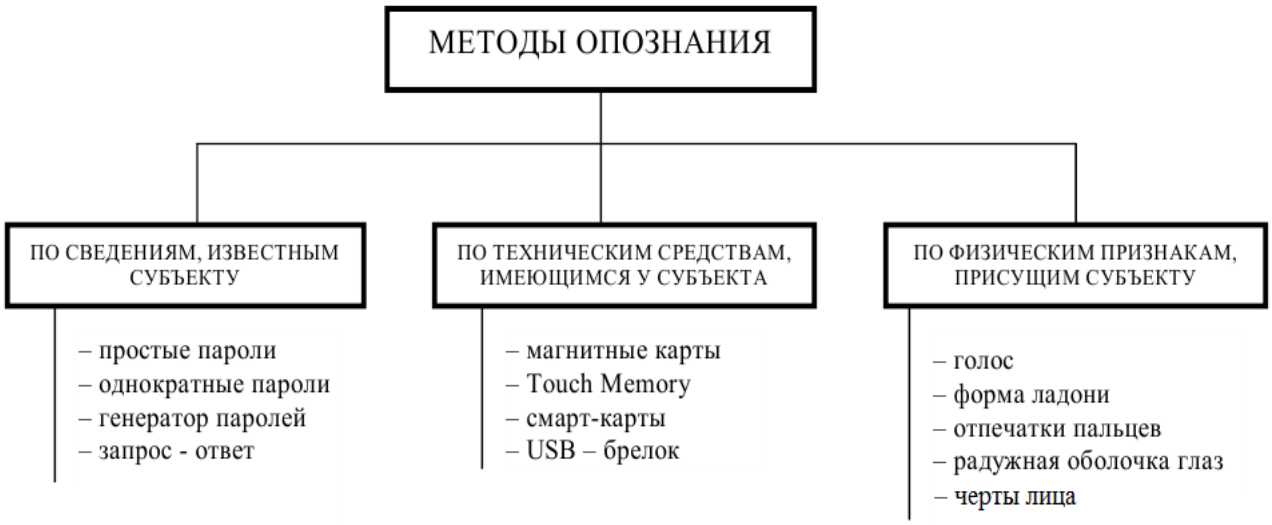


Рисунок 10.1 – Классификация методов опознания

10.1. Опознание на основе принципа «что знает субъект»

10.1.1. Метод паролей

Данный метод заключается в том, что пользователь на клавиатуре компьютера или специально имеющемся наборном поле набирает только ему известную комбинацию букв и цифр, которая собственно и является паролем. Введенный пароль сравнивается с эталонным, хранящимся в системе, и при положительном результате проверки пользователь получает доступ к системе. Приведенная схема опознания является простой с точки зрения реализации, так как не требует никакой специальной аппаратуры и реализуется посредством небольшого объема программного обеспечения.

Рассмотрим алгоритм функционирования парольного средства аутентификации пользователей в операционной системе Microsoft Windows XP. Аутентификация пользователей в операционной системе Microsoft Windows XP основана на использовании паролей и реализуется следующими компонентами: Winlogon, GINA, LSASS, MSV1_0, SAM.

Winlogon – системный процесс, который отвечает за проведение операций входа и выхода пользователя в ОС.

GINA (Graphical Identification and Authentication) – файл динамической библиотеки, который предназначен для ввода имени пользователя и его пароля.

LSASS (Local Security Authentification SunSystem) – подсистема локальной аутентификации, которая управляет процессом аутентификации.

MSV1_0 – пакет аутентификации, который используется ОС при интерактивном входе пользователя. Предназначен для идентификации и аутентификации пользователя.

SAM (Security Account Manager) – объект, который ведет базу данных имен пользователей и паролей.

Схема взаимодействия компонентов ОС Microsoft Windows XP в процессе интерактивного входа пользователя представлена на рисунке 10.2.

Алгоритм функционирования средства аутентификации с использованием паролей в операционной системе Microsoft Windows XP при интерактивном входе представлен на рисунке 10.3. Процесс аутентификации включает в себя следующие этапы.

- запрос на вход в систему;
- ввод имени и пароля;
- идентификация пользователя;
- аутентификация пользователя;
- создание маркера доступа.

Запрос на вход в систему. Пользователь нажимает комбинацию клавиш Ctrl+Alt+Del. В результате этого Winlogon вызывает GINA, который выводит на экран поля, необходимые для ввода имени и пароля пользователя.

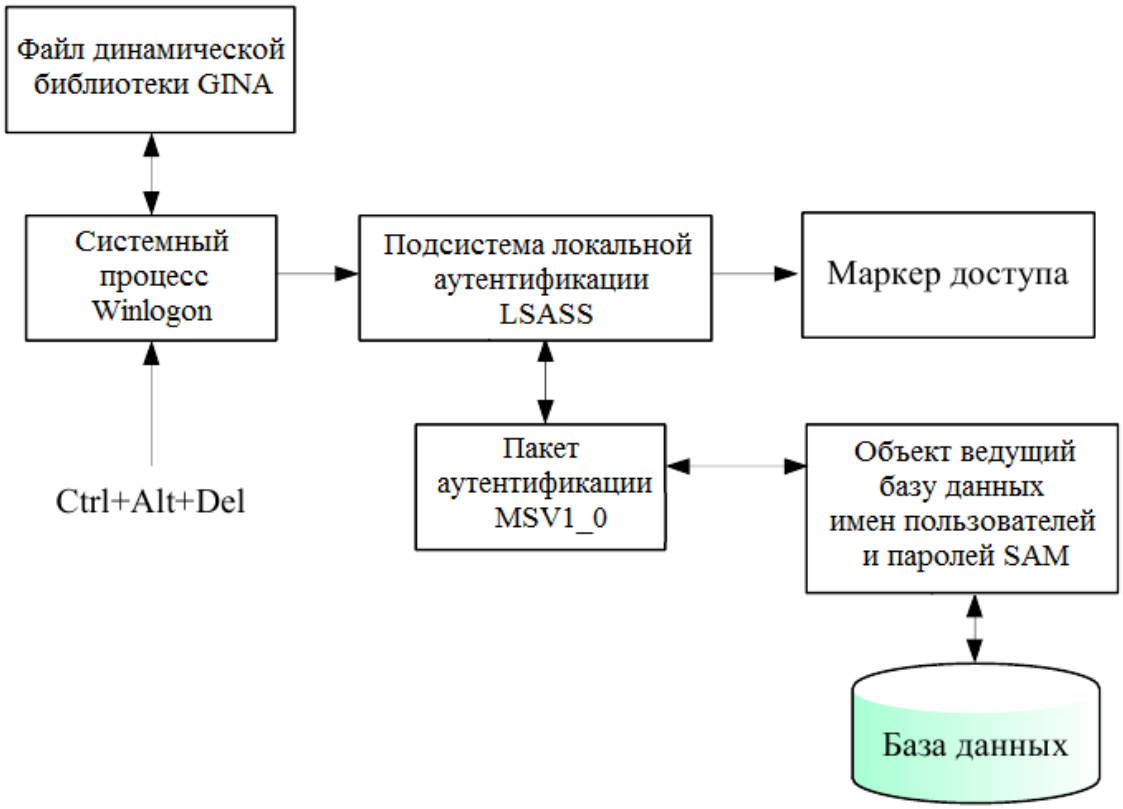


Рисунок 10.2 - Компоненты, участвующие в процессе интерактивного входа пользователя в операционную систему Microsoft Windows XP

Ввод имени и пароля. После набора пользователем имени и пароля GINA передает эти данные в Winlogon, который производит хеширование пароля, создает уникальный локальный идентификатор защиты (SID – Security IDentifer) для этого пользователя и вызывает LSASS.

Идентификация пользователя. LSASS подключает пакет аутентификации MSV1_0, который принимает от Winlogon имя пользователя и хешированную версию пароля и посылает в SAM запрос на получение из учетной записи пользователя, которая хранится в базе данных SAM, хешированного пароля. Идентификация заключается в нахождении введенного пользователем имени в базе данных SAM. Если введенное пользователем имя не содержится в базе данных SAM, то MSV1_0 возвращает в LSASS статус отказа.

Аутентификация пользователя. В случае нахождения имени пользователя в базе данных MSV1_0 сравнивает хешированный пароль пользователя с тем, который хранится в базе данных SAM и соответствует учетной записи пользователя. Если эти данные совпадают, MSV1_0 генерирует локально-уникальный идентификатор сеанса входа (LUID – Locally Unique IDentifer) и передает его вместе с SID в LSASS. Если данные не совпадают, то MSV1_0 возвращает в LSASS статус отказа.

Создание маркера доступа. Собрав необходимую информацию, LSASS вызывает исполнительную систему для создания маркера доступа. Исполнительная система создает маркер доступа для интерактивного сеанса, который

включает в себя SID пользователя. После успешного создания маркера доступа LSASS дублирует его, создавая описатель, который передается Winlogon, а свой описатель закрывает. На этом этапе LSASS сообщает Winlogon об успешном входе. При наличии в LSASS статуса отказа Winlogon сообщает пользователю о неправильно введенном имени или пароле. Программа Winlogon дает пользователю несколько попыток ввода правильных идентификатора и пароля. После превышения числа допустимых попыток программа прекращает свое выполнение.

Схема с использованием простого пароля имеет два недостатка:

– большинству пользователей сложно запомнить произвольное число, используемое в качестве пароля;

– пароль может быть использован другим лицом, так как его легко подсмотреть.

Модернизацией схемы с использованием простого пароля является пароль однократного использования. В этой схеме пользователю выдается список из N паролей, такие же N паролей хранятся в системе. Данная схема обеспечивает большую степень безопасности, но она является и более сложной.

Здесь при каждом обращении к системе синхронно используется пароль с текущим номером, а все пароли с предыдущими номерами вычеркиваются. В случае если старый пароль из предыдущего сеанса стал известен другому пользователю, система его не воспринимает, так как действующим будет следующий по списку пароль.

Схема паролей однократного использования имеет следующие недостатки:

– пользователь должен помнить или иметь при себе весь список паролей и следить за текущим паролем;

– в случае если встречается ошибка в процессе передачи, трудно определить, следует ли передавать тот же самый пароль или послать следующий;

– необходимо иметь разные таблицы паролей для каждого пользователя, так как может произойти рассинхронизация работы.

Последний недостаток можно устранить, используя генератор паролей. В этом случае в ЭВМ реализуется алгоритм, осуществляющий преобразование

$$F(x,k) = y,$$

где x, k, y – двоичные векторы соответственно характеристического номера, ключа и пароля.

Реализация процедуры опознания пользователя сводится к двум задачам: заготовке паролей и установлению подлинности.

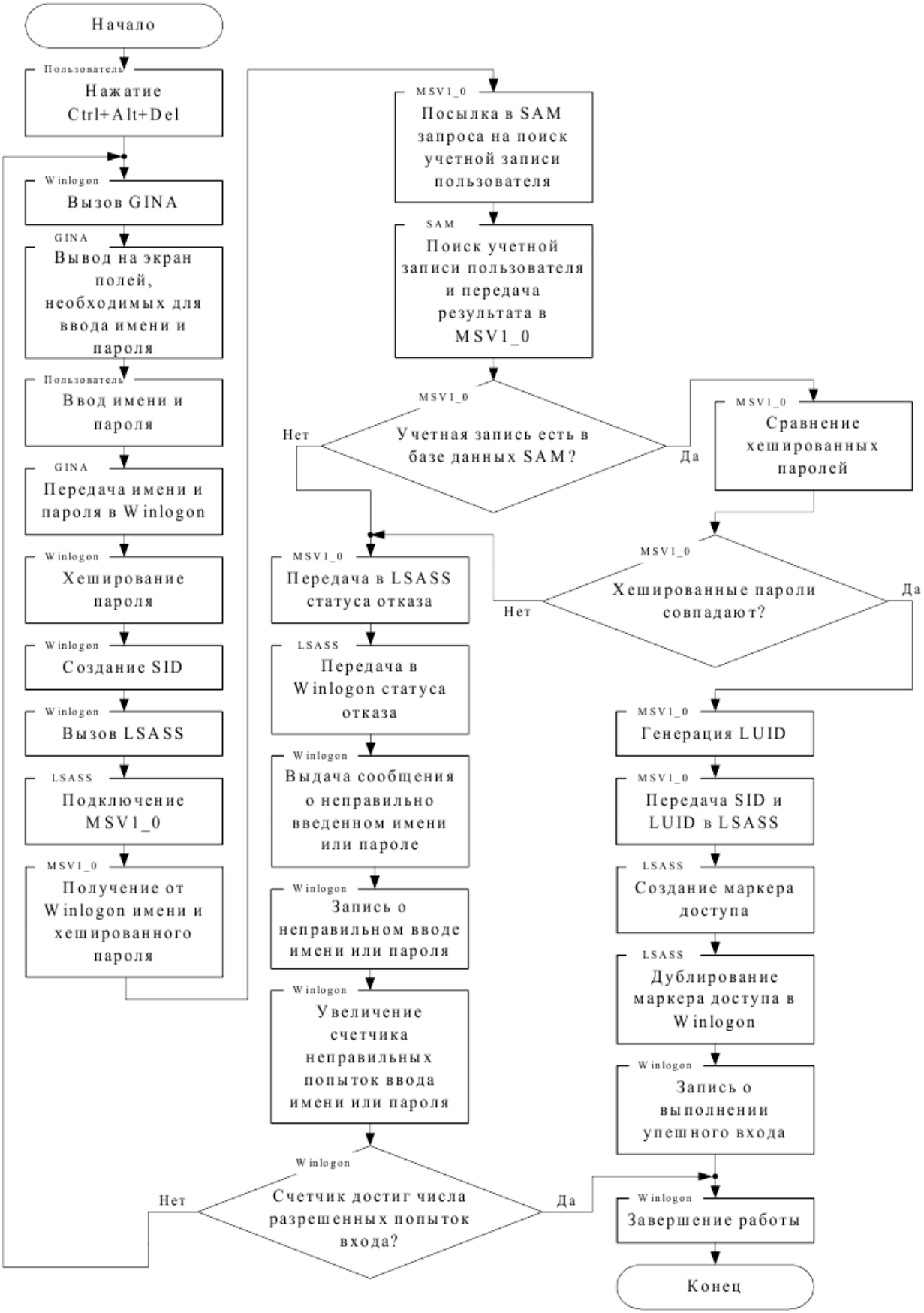


Рисунок 10.3. Схема алгоритма функционирования средства аутентификации с использованием паролей в операционной системе Microsoft Windows XP

При заготовке паролей с помощью преобразования $F(x,k)=y$ получают набор чисел

$$X_i^j, П_i^j,$$

где i – номер пользователя; j – номер обращения данного пользователя;

П – текущее значение пароля, сформированное на ключе k.

Сгенерированный набор чисел выдается соответствующим пользователям.

Опознание системой пользователя I происходит следующим образом: пользователь с номером i вводит парольный набор $X_i^j, П_i^j$ в ЭВМ. Программа опознания выделяет номер пользователя Xij, а также запоминает пароль $П_i^j$. Для каждого i-го пользователя существует свой счетчик обращений S_i. В случае, если $j \le S_i$, программа выдает сообщение о несанкционированном доступе (НСД). В противном случае включается генератор паролей. Преобразование $F(x_i^j, k)$ на действующем ключе k выдает число y, которое сравнивается с паролем $П_i^j$. В случае совпадения y и $П_i^j$ пользователь считается опознанным, а в случае несовпадения выдается сигнал о несанкционированном доступе.

Использование генератора паролей избавляет от необходимости хранить таблицы паролей для каждого пользователя, однако первые два недостатка при его использовании сохраняются.

10.1.2. Метод «запрос-ответ»

В методе «запрос-ответ» набор ответов на m стандартных и n ориентированных на пользователя вопросов хранится в ЭВМ и управляет программой опознания. Когда пользователь делает попытку включиться в работу, программа опознания случайным образом выбирает и задает ему некоторые (или все) из этих вопросов. Пользователь должен дать правильный ответ на все вопросы, чтобы получить разрешение на доступ к системе. Вопросы могут быть выбраны таким образом, чтобы пользователь запомнил ответы и не записывал их.

Модификация этого метода предполагает изменение каждый раз одного или более вопросов, на которые пользователь давал ответ до этого.

Существует два варианта использования метода «запрос-ответ», вытекающих из условий m = 0 или n = 0. Вариант с m = 0 предполагает, что вопросы составлены на основе различных фактов биографии индивидуального пользователя, представляют собой имена его друзей, дальних родственников, старые адреса и т.д. Пользователь, который сам предложил опознавательный вопрос, всегда даст на него правильный ответ, чего не сможет сделать злоумышленник. Иногда предпочтительнее вариант с n = 0, т.е. пользователям задается большее количество стандартных вопросов и от них требуются ответы на те, которые они сами выберут. Достоинство рассмотренной схемы в том, что пользователь может выбирать вопросы, а это дает весьма высокую степень безопасности в процессе включения в работу. В то же время нет необходимости хранить в системе тексты вопросов для каждого пользователя, достаточно хранить указате-

ли на вопросы, выбранные данным пользователем, вместе с информацией, устанавливающей его подлинность. Текст каждого стандартного вопроса необходимо ввести для хранения только один раз, поэтому в системе с большим числом пользователей это может дать экономию памяти.

Наряду с достоинствами метод «запрос-ответ» все же имеет и недостатки, ограничивающие возможность его использования, а именно:

– метод требует проявления изобретательности от самих пользователей, что для них является дополнительной нагрузкой;

– большинство людей, как правило, предлагают стереотипные вопросы и ответы в качестве опознавательных, поэтому весьма вероятно, что настойчивый нарушитель может, собрав статистику, предугадать многие вопросы и ответы;

– процедура обмена множеством опознавательных запросов и соответствующих им ответов может быть сложной и утомительной для пользователей;

– метод «запрос-ответ» может использоваться только для небольших организованных групп пользователей, он неприменим для массового использования в силу некоторой громоздкости.

10.2. Опознание на основе принципа «что имеет субъект»

К данному классу опознания относятся методы, основывающиеся на физических средствах, которые имеет при себе данный пользователь, обращающийся к системе. К ним относятся магнитные карточки, смарт-карты, USB-ключи, таблетки Touch Memory и прочие подобные средства, которые можно объединить общим названием – электронный ключ. Электронным ключом в самом общем смысле являются физические носители идентификатора субъекта и его пароля. Кроме того, на носителях содержится дополнительная информация, необходимая в процессе опознания субъекта.

Для восприятия смарт-карта должна иметь ридер. В процессе обмена информацией с ридером происходит опознание смарт-карты. Опознание субъекта происходит после подтверждения им того, что именно он является владельцем смарт-карты в результате ввода с клавиатуры PIN-кода. Аналогом ридера для USB-ключей выступает стандартный USB-порт, а для электронного ключа Touch Memory – считывающее устройство.

10.2.1. Идентификационные магнитные карты

В магнитных картах информация записывается на нескольких дорожках магнитного слоя и представляет собой данные, используемые для идентификации. К этим данным относятся: номер пользователя или его имя, пароль, количество допустимых использований карты и т.д. Наряду с очевидной простотой использования магнитные карты обладают низкой защищенностью от копирования содержимого. Для защиты от копирования магнитные карты снабжаются различными защитными средствами. Один из методов состоит в нанесении магнитного слоя обычного типа поверх второго слоя с более высокой коэрци-

тивной силой, т.е. для изменения состояния первого слоя требуется более сильное магнитное поле. В этом случае обычными методами невозможно считать или изменить запись нижнего слоя. Считывающее устройство, читая карту, содержащую идентификатор, вначале создает поле, стирающее любую запись, сделанную обычным способом, а затем уже считывает лежащую ниже «твердую» запись, в которой находится идентификационная информация.

В другом методе используется постоянная магнитная разметка ленты, которая наносится в процессе ее производства. Метод, известный под названием «влажной разметки», состоит в определенной ориентации осей ферромагнитных кристаллов до момента, пока наполнитель еще не высох, причем селективная ориентация осей кристаллов в различных частях ленты создает магнитную запись, которую никак нельзя изменить. Чтобы прочесть эту запись, кристаллы необходимо подвергнуть воздействию постоянного магнитного поля с определенной ориентацией. Изменение положения кристаллов вдоль ленты будет наводить внешнее поле, которое можно прочитать с помощью обычных, удобно расположенных головок. Изготовленные таким образом идентификационные карточки могут обеспечить «уникальную» идентичность, которую трудно подделать, поскольку для этого требуется овладеть технологией производства магнитных покрытий и влажной разметки.

10.2.2. Электронные ключи

Электронный ключ в самом общем смысле представляет собой физический носитель секретного кода, являющегося аутентификатором пользователя. В отличие от парольных систем использование электронного ключа (ЭК) имеет ряд преимуществ:

– пользователю не надо запоминать значение пароля, так как пароль записан в ключе;

– пользователь освобожден от проблемы защиты пароля от компрометации при его вводе, так как пароль считывается из ключа;

– все функции по защите от подделки пароля или его несанкционированного использования (метод разовых паролей, метод «рукопожатия») возлагаются на электронный ключ;

– секретный код можно сделать сколь угодно большим, так как пользователь с ним непосредственно не работает.

Рассмотрим алгоритм функционирования средства аутентификации с использованием смарт-карт. Средство аутентификации с использованием смарт-карты реализуется следующими модулями: центральный процессор (ЦП), ПЗУ, ОЗУ, ЭСППЗУ, программное обеспечение (ПО), дисплей, клавиатура, приемо-передатчик.

ЦП предназначен для реализации криптографических алгоритмов и разграничения доступа к хранящейся в памяти смарт-карты информации. В ПЗУ хранится исполняемый код ЦП, а ОЗУ используется в качестве рабочей памяти. Энергонезависимая память для хранения информации пользователя смарт-карты (ЭСППЗУ) необходима для хранения изменяемых данных владельца кар-

ты. ПО предназначено для осуществления взаимодействия смарт-карты с рабочей станцией. Приемопередатчик предназначен для приема и передачи информации как от смарт-карты к рабочей станции, так и наоборот.

Схема взаимодействия компонентов, участвующих в процессе аутентификации пользователя с использованием смарт-карты, представлена на рисунке 10.4.

Алгоритм функционирования средства аутентификации с использованием смарт-карты представлен на рисунке 10.5 и включает в себя следующие этапы:

- ввод смарт-карты в специальное устройство для чтения.
- идентификация смарт-карты.
- аутентификация пользователя.
- формирование записи о результате входа в систему.

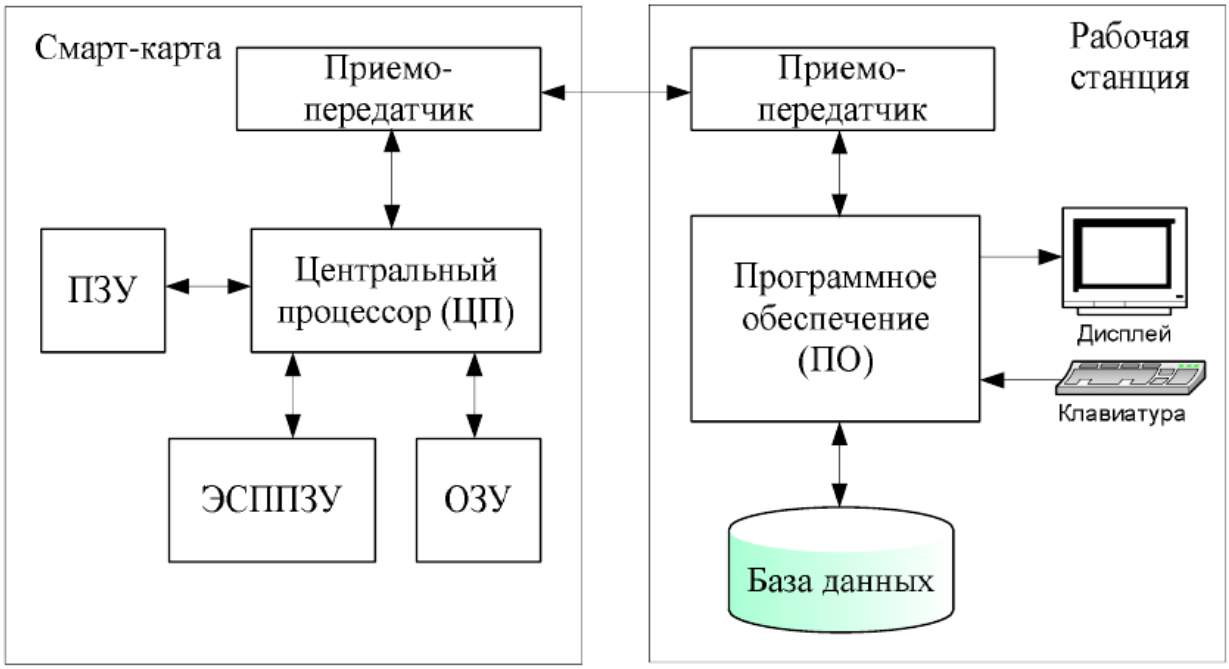


Рисунок 10.4 - Компоненты, участвующие в процессе аутентификации пользователя с использованием смарт-карты.

Ввод смарт-карты в специальное устройство для чтения и ввод пользователем своего PIN-кода. Пользователь вставляет смарт-карту в специальное устройство для чтения смарт-карт (ридер, терминал), которое подключено к рабочей станции. ПО рабочей станции посылает в смарт-карту управляющий сигнал и в ЦП смарт-карты загружается исполняемый код из ПЗУ смарт-карты.

Затем ПО выдает на монитор запрос на ввод пользователем своего PIN-кода. Пользователь вводит с клавиатуры свой PIN-код, который поступает в ПО рабочей станции. Эталонный PIN-код владельца смарт-карты в зашифрованном виде хранится в ЭСППЗУ смарт-карты.

Идентификация смарт-карты. ПО посылает запрос ЦП смарт-карты на выдачу персональной информации, которая содержит срок окончания работы смарт-карты и ее серийный номер.

Если срок окончания работы смарт-карты подошел к концу, то ПО выдает на монитор сообщение о том, что смарт-карта устарела и пользователю необходимо изъять ее.

Если срок работы смарт-карты еще не истек, то ПО ищет в базе данных учетную запись с полученным серийным номером смарт-карты. Идентификация смарт-карты считается успешной, если ПО находит в базе данных учетную запись с таким серийным номером.

Если на предъявленный серийный номер учетной записи нет, то это означает, что смарт-карта не является зарегистрированной в данной системе и,следовательно, не проходит идентификацию. В таком случае ПО выводит на монитор сообщение о том, что формат смарт-карты является неверным и предлагает пользователю изъять ее.

Аутентификация пользователя:

Выработка ПО случайного числа. ПО вырабатывает случайное число и посылает его в ЦП смарт-карты. Случайное число записывается в ОЗУ смарт-карты.

Вычисление смарт-картой хеш-кода. ЦП смарт-карты вычисляет хеш-код от зашифрованного на общем для смарт-карты и ЭВМ ключе PIN-кода, сцепленного со случайным числом и ключом приложения. Полученный хеш-код ЦП отправляет в ЭВМ.

Вычисление устройством доступа хеш-кода. ПО вычисляет хеш-код от зашифрованного на общем для смарт-карты и ЭВМ ключе введенного пользователем PIN-кода, сцепленного со случайным числом и ключом приложения.

Сравнение результатов устройством доступа и принятие решения о подлинности пользователя. ПО сравнивает вычисленные хеш-коды. Если хеш-коды не совпадают, то ПО выдает на монитор сообщение о том, что введен неверный PIN-код и предлагает повторить попытку ввода PIN-кода. Так как число попыток ввода ограничено, то если PIN-код введен неверно установленное количество раз, устройство доступа блокирует смарт-карту. Если хеш-коды совпадают, то ПО переходит к записи результата входа в систему.

Формирование записи о результате входа в систему. При совпадении хеш-кодов ПО делает запись об успешном входе в систему и выводит на монитор соответствующее сообщение. Если хеш-коды не совпадают, то ПО выдает запись об отказе в доступе.

Рисунок 10.5 - Схема алгоритма функционирования средства аутентификации с использованием смарт-карт

10.3. Опознание на основе принципа «что присуще субъекту»

Данный принцип опознания базируется на определении индивидуальных характеристик, присущих каждому пользователю и позволяющих выделить его среди других лиц. К наиболее широко используемым персональным характеристикам относятся голос, личная подпись, форма ладони и отпечатки пальцев, форма лица. В последнее время появилось еще несколько методов физического опознания – по структуре сетчатки глаз, сопротивлению определенных участков кожи, запаху тела и др. В каждом случае способ опознания состоит в измерении индивидуальных характеристик и вычислении индексов, аналогичных характеристическим параметрам распознавания образов, которые можно передать в центральную ЭВМ для сопоставления с набором индексов, хранящихся в памяти ЭВМ и взятых непосредственно у интересующего лица.

10.3.1. Параметры идентификации физиологических признаков

Механизм опознания личной подписи может измерять число касаний и отрывов пера от бумаги, среднюю вертикальную скорость движения пера, число вертикальных отклонений и множество других подобных параметров. Эти характеристики могут быть самыми разнообразными, однако не все из них являются независимыми, и задача состоит в том, чтобы выбрать хороший набор характеристик с достаточно малой взаимной корреляцией. Проверка подлинности подписи зависит от движения пера, которое нельзя воспроизвести по виду подписи, зафиксированной на бумаге. Это практически полностью исключает возможность подлога, так как умение профессионально подделывать подписи, основано на внешнем виде почерка. Набор измеряемых характеристик должен сохраняться в тайне, так как их знание может привести к подделке подписи посредством тренировки в копировании измеряемых характеристик. Как показала практика, обеспечение секретности – это сложная задача.

Аналогичные особенности характерны и для других методов опознания этого класса. Например, некоторые устройства, определяющие форму ладони, измеряют прозрачность тканей кожи между пальцами для защиты от подлогов с помощью картонных шаблонов. Механизмы, построенные на анализе отпечатков пальцев, используют мельчайшие детали в виде разветвлений, окончаний и пробелов в линиях на кончиках пальцев. Так как в каждом отпечатке содержится множество таких отличий, измеряемые характеристики могут базироваться на выбранном наборе деталей. Существуют два основополагающих алгоритма распознавания отпечатков пальцев:

− по отдельным деталям (характерным точкам);

− по рельефу всей поверхности пальца.

В первом случае устройство регистрирует только некоторые участки, уникальные для конкретного отпечатка, и определяет их взаимное расположение. Во втором случае обрабатывается изображение всего отпечатка.

Метод опознания субъекта по лицу основан на уникальности черт лица. Метод заключается в преобразовании черт конкретного лица в алгоритмиче-

скую модель, которая сравнивается или с фотографией на пропуске, или с содержимым базы фотографических данных.

Метод опознания субъекта по радужной оболочке глаза основан на уникальности рисунка радужной оболочки каждого субъекта. Радужная оболочка субъекта сканируется, разворачивается и преобразуется в цифровую последовательность. Подтверждение подлинности субъекта происходит на основании сравнения полученной цифровой последовательности с эталонной.

Метод опознания по образцу голоса основан на том, что у каждого субъекта неповторимый голосовой рисунок, который определяется полом, физическими особенностями субъекта, в частности его речевым аппаратом: типом строения голосовых связок, полостью носа, формой рта, таких характеристик голоса, как частота и амплитуда. Этот метод построен на выделении различных сочетаний частотных и статистических характеристик голоса.

10.3.2. Средство аутентификации с устройством сканирования отпечатка пальца

Данное устройство использует отпечаток пальца в качестве биометрического признака личности и реализуется такими компонентами, как датчик изображения папиллярных линий кожи пальца, USB-разъем, USB-порт, программное обеспечение (ПО), монитор, клавиатура.

Датчик изображения папиллярных линий кожи пальца (ДИПЛКП) предназначен для сканирования отпечатка пальца, преобразования полученного изображения в цифровую форму и передачу его на USB-разъем. USB-разъем и USB-порт служат для передачи цифровой информации от датчика изображения папиллярных линий кожи пальца в ЭВМ. ПО предназначено для работы с изображением отпечатка пальца, сравнения отпечатка пальца с эталонным, хранящимся в базе данных, и для управления диалогом с пользователем.

Схема взаимодействия компонентов, участвующих в процессе аутентификации пользователя по отпечатку пальца, представлена на рисунке 10.6.

Алгоритм функционирования средства аутентификации по отпечатку пальца представлен на рисунке 10.7 и включает в себя следующие этапы:

- ввод имени пользователя;
- сканирование отпечатка пальца;
- работа с файлом отпечатка пальца;
- идентификация пользователя;
- аутентификация пользователя;
- принятие окончательного решения.

Ввод имени пользователя. Для входа в систему пользователь запускает на ЭВМ процесс аутентификации. ПО средства аутентификации выдает на монитор запрос на ввод пользователем своего имени. Пользователь вводит имя с клавиатуры.

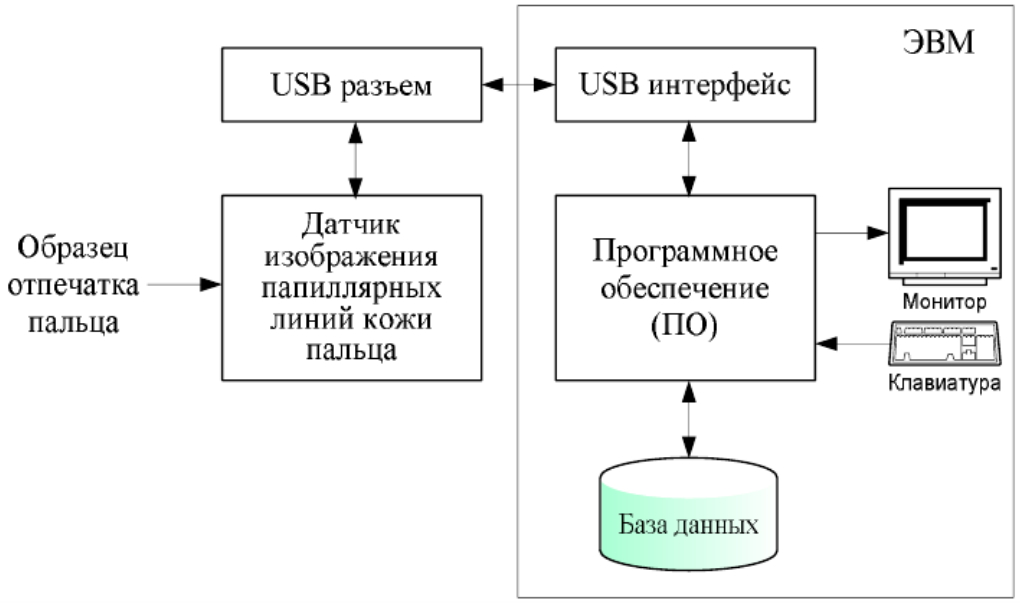


Рисунок 10.6 - Компоненты, участвующие в процессе аутентификации пользователя по отпечатку пальца

Сканирование отпечатка пальца. После ввода пользователем своего имени ПО активирует датчик изображения папиллярных линий кожи пальца. После этого пользователь прикладывает свой палец к сканирующей области датчика, который сканирует отпечаток пальца пользователя и преобразует его в цифровую форму.

Работа с файлом отпечатка пальца.

Формирование файла картинки отпечатка пальца. От датчика изображения папиллярных линий кожи пальца через USB-разъем в USB-порт цифровая форма отпечатка пальца попадает в ЭВМ. ПО сохраняет полученную от датчика информацию в файл-образ отпечатка пальца с заданными разрешением, числом пикселов на дюйм, количеством уровней яркости. Далее ПО создает образ папиллярных линий пальца, где темным участкам соответствуют выступы папиллярного рисунка, а светлым – впадины.

Улучшение качества исходного изображения отпечатка. Для улучшения структуры гребней папиллярных линий и резкости их границ ПО производит низкочастотную фильтрацию изображения отпечатка пальца.

Бинаризация изображения отпечатка. ПО производит пороговую обработку изображения отпечатка пальца, в результате которой пиксели изображения, цвета которых меньше заданного порога делаются чёрными, а те, цвета которых выше, – белыми.

Утончение линий изображения отпечатка. ПО производит утончение линий изображения отпечатка пальца до тех пор, пока эти линии не станут равными одному пикселу.

Идентификация пользователя. ПО ищет в базе данных учетную запись с введенным именем. Идентификация считается успешной, если ПО находит в базе данных учетную запись с введенным именем пользователя.

Аутентификация пользователя.

Выделение минуций. ПО производит локальную обработку всего изображения отпечатка пальца с помощью маски 9×9 пикселов и подсчета числа пикселов, находящихся вокруг центра маски и имеющих ненулевые значения. Пиксел в центре маски принимается за минуцию, если он сам имеет ненулевое значение и если число «соседей» также ненулевое и равно 1 или 2.

Координаты обнаруженных минуций, а также углы их ориентации ПО записывает в вектор минуций.

Регистрация данных. При положительном результате идентификации ПО выбирает эталонный вектор минуций, соответствующий данному пользователю и определяет параметры аффинных преобразований, при которых некоторая минуция сформированного вектора будет согласована с некоторой минуцией эталонного вектора.

При отрицательном результате идентификации ПО выводит на монитор сообщение об отказе в доступе. Количество попыток ограниченно. После исчерпания всех попыток ПО закрывается, а его запуск блокируется.

Поиск пар соответствующих друг другу минуций. На каждом шаге ПО подвергает аффинным преобразованиям координаты минуций из полученного вектора и полученные новые координаты сопоставляет с каждой из координат минуций эталонного вектора.

Оценка меры согласования двух сопоставляемых отпечатков. ПО осуществляет количественную оценку согласования двух сопоставляемых отпечатков, как отношение квадрата количества найденных пар минуций к произведению количества минуций в полученном векторе минуций на количество минуций в эталонном векторе минуций, умноженное на сто процентов.

Принятие окончательного решения. В базе данных хранятся несколько эталонных векторов минуций одного и того же отпечатка пальца, полученных при разных условиях его сканирования. ПО сравнивает полученный вектор минуций с каждым из эталонных векторов. После сравнений ПО выбирает тот эталонный вектор минуций, количественная оценка согласования которого максимальна. Если эта количественная оценка согласования превышает некоторое пороговое значение, то ПО вырабатывает положительный результат аутентификации пользователя и выдает на монитор сообщение об успешной аутентификации. В противном случае ПО вырабатывает отрицательный результат и выдает на монитор сообщение об отказе в доступе.

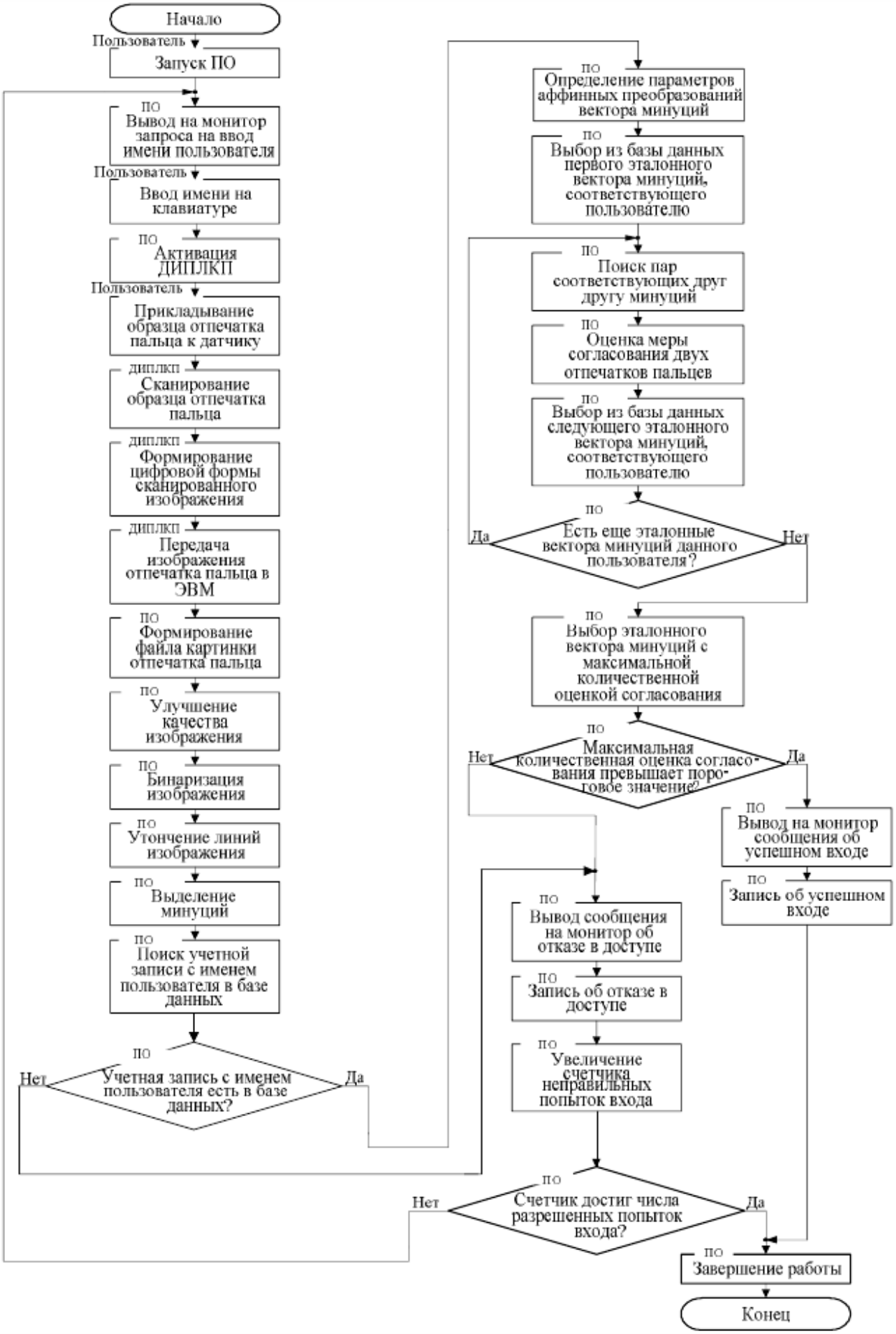


Рисунок 10.7 - Схема алгоритма функционирования средства аутентификации с устройством сканирования отпечатка пальца

10.3.3. Алгоритм функционирования средства аутентификации с устройством распознавания голоса

Аутентификация основана на использовании образца голоса в качестве биометрического признака и реализуется следующими компонентами: микрофоном, программным обеспечением (ПО), монитором, клавиатурой.

Микрофон служит для ввода образца голоса, а клавиатура – для ввода имени пользователя. ПО предназначено для работы с образцом голоса пользователя, для выделения и сравнения векторов речевых признаков и для управления диалогом с пользователем. На монитор выводятся необходимые пользователю сообщения.

Схема взаимодействия компонентов, участвующих в процессе аутентификации пользователя по образцу голоса, представлена на рисунке 10.8.

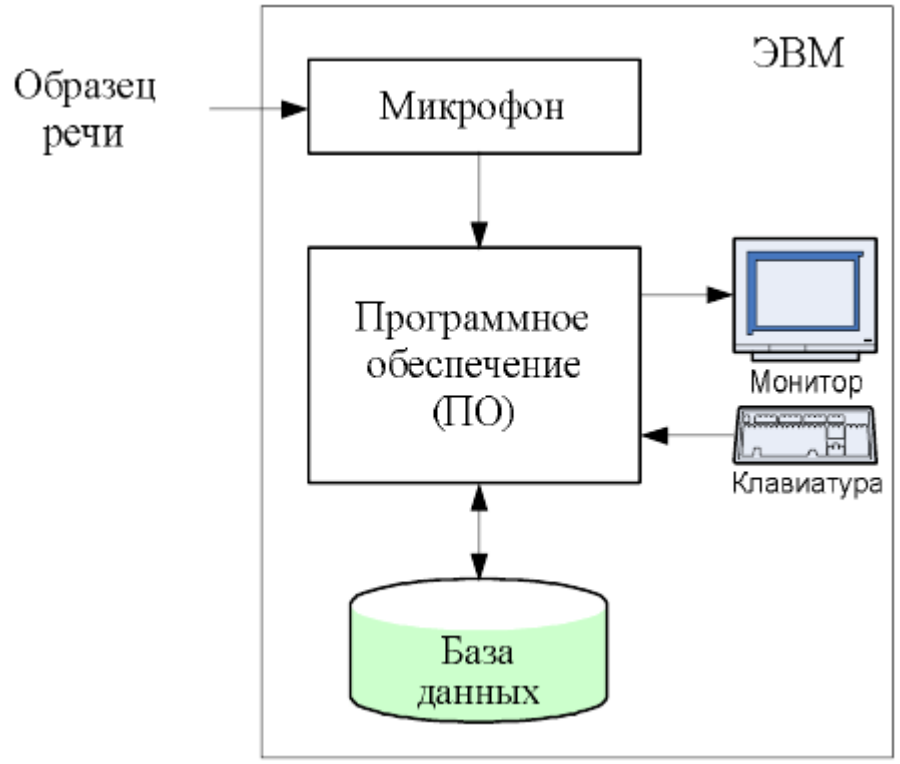


Рисунок 10.8 - Компоненты, участвующие в процессе аутентификации пользователя по образцу голоса

Алгоритм функционирования средства аутентификации по образцу голоса представлен на рисунке 10.9 и включает в себя следующие этапы.

- ввод имени пользователя и образца голоса.
- идентификация пользователя.
- выделение векторов речевых признаков.
- принятие окончательного решения об аутентификации.

Ввод имени пользователя и образца голоса. Пользователь запускает пользовательский интерфейс, который предлагает ввести свое имя и образец голоса. Затем пользователь набирает на клавиатуре свое имя и подает на микрофон фрагмент речи, который представляет собой голосовой пароль.

Идентификация пользователя. ПО производит поиск в базе данных учетной записи с введенным именем. Идентификация считается успешной, если ПО находит в базе данных учетную запись с введенным именем пользователя.

Выделение векторов речевых признаков. Поданный на микрофон образец голоса записывается в память ЭВМ и обрабатывается ПО, которое определяет векторы речевых признаков, представляющие характерные параметры входного речевого сигнала. Векторы речевых признаков определяются с помощью линейного предсказания для нахождения его кепстральных коэффициентов. ПО генерирует векторы речевых признаков в виде кепстральных коэффициентов методом векторного квантования и формирует из них матрицу.

Далее производится вычисление мер близости между сгенерированной матрицей кепстральных коэффициентов и каждой из трех эталонных матриц, хранящихся в базе данных и соответствующих имени пользователя. В результате ПО принимает решение о том, превышает мера близости пороговое значение или нет.

Принятие окончательного решения об аутентификации. Если меры близости между сгенерированной матрицей и хотя бы двумя из трех эталонных матриц не превышают порогового значения, то ПО принимает положительное решение об аутентификации пользователя и выводит соответствующие сообщение на монитор. В противном случае ПО принимает отрицательное решение об аутентификации пользователя и выводит на монитор сообщение об отказе в доступе.

10.4. Функциональная структура средства аутентификации

Анализ реализаций средств аутентификации, приведенных в предыдущих разделах, показывает, что они имеют общие закономерности функционирования. Каждый из рассмотренных алгоритмов работы средств аутентификации содержит этапы:

− обработки входных воздействий и преобразования их в необходимый вид;

− идентификации и аутентификации;

− принятия решения о разрешении доступа к защищаемой системе или его запрете;

− контроля исполнения управляющего воздействия.

Таким образом, в процессе работы алгоритма каждое из средств аутентификации субъекта выполняет следующие функции:

− обнаружения;

− опознания;

− управления;

− контроля.

Рисунок 10.9 - Схема алгоритма функционирования средства аутентификации по образцу голоса

К функции *обнаружения* относятся те элементы алгоритма работы средства аутентификации, которые обеспечивают выявление подлежащих анализу входных воздействий и их преобразование в форму, необходимую для работы

средства аутентификации. К функции обнаружения, например, можно отнести процессы сканирования отпечатка пальца и обработки файла рисунка отпечатка пальца.

К функции *опознания* относятся те элементы алгоритма работы средства аутентификации, которые обеспечивают проверку законности субъекта и устанавливают, является ли он тем, за кого себя выдает. К функции опознания, например, можно отнести процессы сравнения уникального серийного номера смарт-карты с номерами, имеющимися в базе данных рабочей станции, сравнение хеш-кода пароля с эталонным хеш-кодом при парольной аутентификации в ОС Windows.

К функции *управления* относятся те элементы алгоритма работы средства аутентификации, которые обеспечивают формирование разрешающего или запрещающего управляющего воздействия. К функции управления, например, можно отнести процесс передачи LUID (разрешающее управляющее воздействие) или статуса отказа (запрещающее управляющее воздействие) в LSASS.

К функции *контроля* относятся те элементы алгоритма работы средства аутентификации, которые обеспечивают проверку соответствия управляющего воздействия, выработанного функцией управления, результатам аутентификации.

Таким образом, схема обобщенного алгоритма работы средства аутентификации должна иметь вид, представленный на рисунке 10.11.

Сформулируем ряд утверждений, определяющих полноту и достаточность полученной блок-схемы для представления алгоритма функционирования средства аутентификации.

Утверждение 1.1. Средства аутентификации относятся к классу средств защиты каналов доступа.

Утверждение 1.2. Необходимым и достаточным условием реализации средства аутентификации является наличие в его функциональной структуре совокупности функций обнаружения, опознания, управления и контроля.

Утверждение 1.3. Алгоритм работы средства аутентификации заключается в строгой последовательности выполнения функций обнаружения, опознания, управления и контроля.

Утверждение 1.4. Средство аутентификации должно обеспечивать формирование выходного воздействия только при выполнении полного цикла работы.

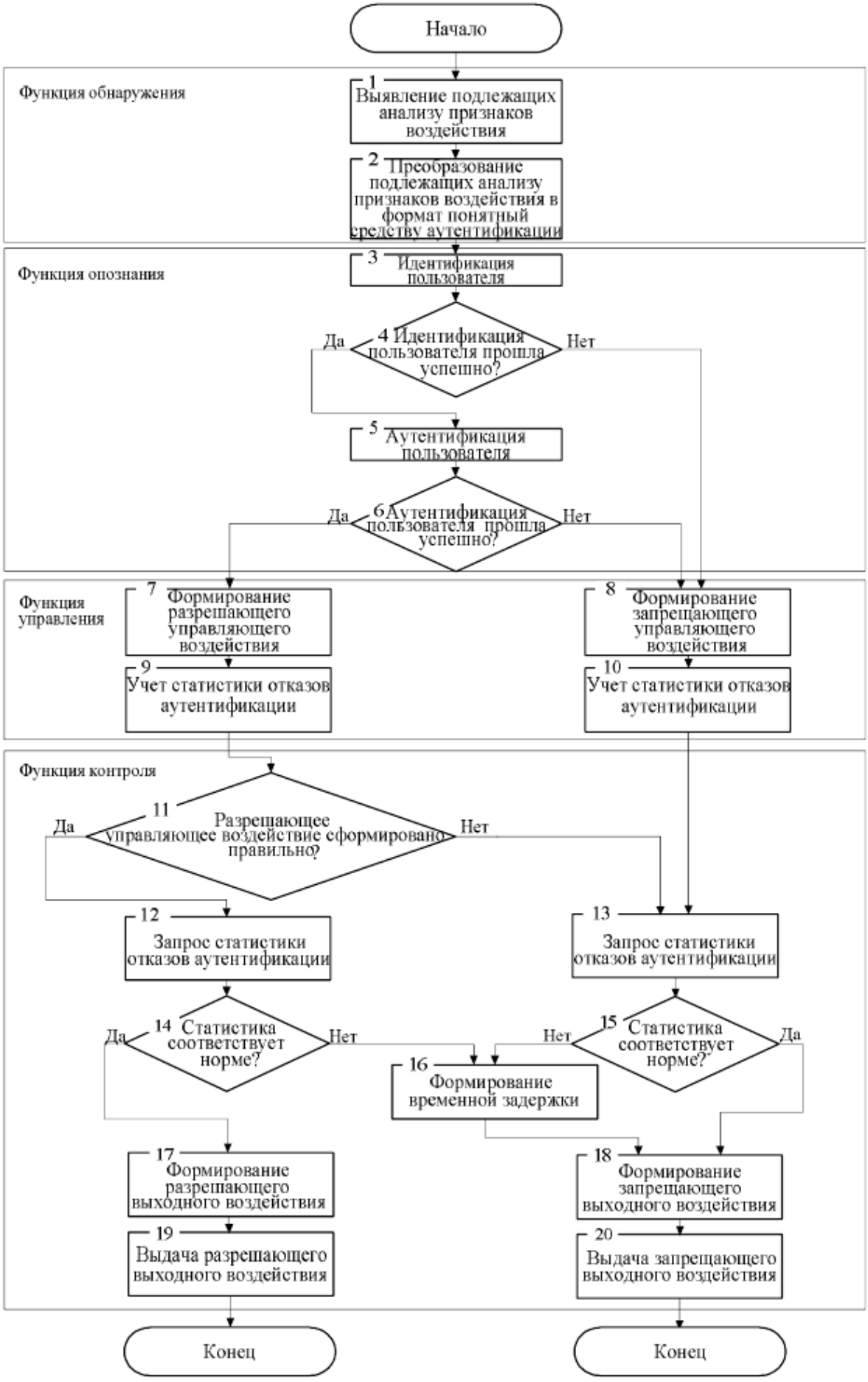


Рисунок 10.11 - Схема алгоритма работы средства аутентификации

10.5. Эффективность средства аутентификации

Любая техническая система (средство) создается для выполнения определенного набора задач (функций). Выполнение системой (средством) заданного набора задач (функций) назовем операцией. Определим эффективность операции как степень соответствия реального (фактического или ожидаемого) результата операции требуемому, или, иными словами, как степень достижения цели операции. Тогда эффективность технического средства можно определить как степень выполнения заданного набора функций. Как и всякое свойство, эффективность обладает определенной интенсивностью своего проявления. Меру интенсивности проявления эффективности называют *показателем эффективности E.*

Как известно, главной (основной) задачей средства аутентификации является надежное опознание конкретного субъекта. В соответствии с этим показатель эффективности средства аутентификации можно определить как меру приближения вероятности правильного опознания субъекта данным средством в реальных условиях функционирования $P_{ПО}$ требуемой $P_{ТР}$. Тогда функцию соответствия ρ а следовательно, и эффективность Е средства аутентификации можно определить в виде

$$E = F(P_{ТР} - P_{ПО}). \qquad (10.1)$$

Функция F должна обладать определёнными свойствами. При равенстве $P_{ПО}$ и $P_{ТР}$ эффективность средства аутентификации является максимальной и должна быть равна единице, а если $P_{ПО}$ стремится к нулю, то и эффективность снижается и стремится к нулю.

В общем случае выражение для вычисления эффективности средства аутентификации примет вид [20]

$$E = e^{-2\pi\cdot[2\cdot 10^{\delta+1}\cdot(P_{ТР}-P_{ПО})]^2}, \qquad (10.2)$$

Вероятность правильного опознания субъекта средством аутентификации в реальных условиях функционирования можно определить как

$$P_{ПО} = 1 - P_{ПЧ}, \qquad (10.3)$$

где $P_{ПЧ}$ – вероятность пропуска «чужого» субъекта средством аутентификации.

Средство аутентификации может пропустить «чужого» субъекта в том случае, если произойдет хотя бы одно из следующих событий:

− подбор аутентификатора нарушителем;

− выдача разрешающего выходного сообщения в результате отказа (сбоя) оборудования;

− выдача разрешающего выходного сообщения в результате действий нарушителя.

Таким образом, вероятность правильного опознания субъекта средством аутентификации будет иметь вид

$$P_{ПО} = (1 - P_{ПА}) \cdot (1 - P_{ОТ}) \cdot (1 - P_{ДН}), \qquad (10.4)$$

где $P_{ПА}$ – вероятность подбора аутентификатора;

$P_{ОТ}$ – вероятность пропуска «чужого» в результате отказов (сбоев) оборудования;

$P_{ДН}$ – вероятность пропуска «чужого» в результате действий нарушителя.

Тогда формула для вычисления эффективности средства аутентификации будет равна

$$E = F(P_{ТР} - (1 - P_{ПА}) \cdot (1 - P_{ОТ}) \cdot (1 - P_{ДН})). \qquad (10.5)$$

Рассмотрим способы определения указанных в выражении (10.5) вероятностей.

Вероятность $P_{ПА}$ зависит от объёма алфавита, длины аутентификатора и является функцией числа попыток подбора

$$P_{ПА} = 1 - \prod_{i=1}^{k} (1 - P_{\prod_i}), \qquad (10.6)$$

где k – число попыток подбора,

$P_{\prod_i}$ – вероятность подбора аутентификатора с первой попытки.

Вероятность подбора аутентификатора с первой попытки определяется известной формулой

$$P_{ПА1} = \frac{1}{A^n}, \qquad (10.7)$$

где A – объём алфавита,

n – длина аутентификатора.

Отсюда вероятность подбора аутентификатора с k-й попытки равна

$$P_{ПАk} = \frac{1}{A^n - k + 1}, \qquad (10.8)$$

а подбора за k попыток –

$$P_{ПА1} = \frac{k}{A^n}. \qquad (10.9)$$

Вероятность $P_{ОТ}$ определяется надёжностью элементов средства аутентификации и является функцией интенсивности их отказов

$$P_{ОТ}(\lambda) = 1 - e^{\sum_{j=1}^{n} \lambda_{ij} t}, \qquad (10.10)$$

где λ_{ij} – интенсивность отказов элементов, выполняющих i-ю функцию,

n – количество элементов, реализующих i-ю функцию.

Вероятность пропуска «чужого» в результате действия нарушителя РДН можно определить как произведение вероятностей того, что действие нарушителя было реализовано, и что эта реализация привела к пропуску «чужого»:

$$P_{ДН} = P_{РДН} \cdot P_{ПДН}, \qquad (10.11)$$

где $P_{РДН}$ – вероятность того, что действие нарушителя было реализовано,

$P_{ПДН}$ – вероятность того, что реализованное действие нарушителя привело к пропуску «чужого».

В качестве требуемой (расчётной) вероятности правильного опознания выберем вероятность того, что аутентификатор не будет подобран с первой попытки:

$$P_{ТР} = 1 - P_{ПА1}. \qquad (10.12)$$

Вероятность $P_{ПА1}$ (а следовательно, и вероятность $P_{ТР}$) определяется только конструктивными особенностями средства аутентификации, не зависит от внешних и внутренних негативных факторов. Поэтому $P_{ТР}$ может служить верхней границей вероятности $P_{ПО}$.

Таким образом, для оценки эффективности средств аутентификации согласно формуле (10.5), необходимо знать механизмы определения $P_{ОТ}(\lambda)$, не зависящие от класса средства аутентификации, и аналитические выражения для расчёта $P_{ТР}$ применительно к биометрическим средствам.

Контрольные вопросы и задачи

1. Пояснить сущность и особенности классов опознания пользователей в вычислительных сетях.

2. Поясните принцип работы устройств аутентификации по схемам алгоритмов.

3. Какие требования предъявляются к элементам электронных ключей при их реализации?

4. Почему методы опознания по физическим признакам малопригодны для распределенных систем с большим количеством пользователей?

5. Докажите утверждения 1.1, 1.2, 1.3 и 1.4, определяющие полноту и достоверность функциональной структуры средства аутентификации.

6. Укажите недостатки схемы паролей однократного использования.

7. Дайте определение эффективности технической системы и поясните физический смысл показателя эффективности для средства аутентификации.

8. Если число попыток подбора аутентификатора ограничено числом 10, то как изменится показатель эффективности устройства аутентификации с восьмизначным цифровым паролем?

11. ЦИФРОВАЯ СТЕГАНОГРАФИЯ

11.1. Общие сведения. Категории информационной безопасности.

Задача надёжной защиты авторских прав, прав интеллектуальной собственности или конфеденциальных данных (которые в большинстве случаев имеют цифровой формат) от несанкционированного доступа является одной из старейших и нерешённых на сегодня проблем. Поэтому во всём мире назрел

вопрос разработки методов (мер) по защите информации организационного, методологичекского и технического характера, среди них - методы криптографии и стеганографии.

Как показано в разделе 9 данного конспекта лекций *криптографическая* (с греческого χρυπτός – "тайный", *γράφω* – "пишу") защита информации (система изменения последней с целью сделать ее непонятной для непосвященных, сокрытие содержания сообщений за счет их шифрования) не снимает упомянутую выше проблему полностью, поскольку наличие шифрованного сообщения само по себе привлекает внимание, и злоумышленник, завладев криптографически защищенным файлом, сразу понимает о размещении в нем секретной информации и переводит всю суммарную мощь своей компьютерной сети на дешифрование данных.

Скрытие же самого факта существования секретных данных при их передаче, хранении или обработке является задачей *стеганографии* (от греческого *στεγανύός* – "скрытый") – науки, которая изучает способы и методы скрытия конфиденциальных сведений. Задача извлечения информации при этом отступает на второй план и решается в большинстве случаев стандартными криптографическими методами.

Иначе говоря, под скрытием существования информации подразумевается не только невозможность обнаружения в перехваченном сообщении наличия иного (скрытого) сообщения, но и вообще сделать невозможным возникновение любых подозрений на этот счет, поскольку в последнем случае проблема информационной безопасности возвращается к стойкости криптографического кода. Таким образом, занимая свою нишу в обеспечении безопасности, стеганография не заменяет, а дополняет криптографию [21].

Стеганографирование осуществляется различными способами. Общей же чертой таких способов является то, что скрываемое сообщение встраивается в некий непривлекающий внимание объект, который затем открыто транспортируется (пересылается) адресату.

Таким образом, стеганографическая система или, сокращённо стеганосистема – это совокупность средств и методов, которые используются с целью формирования скрытого (незаметного) канала передачи информации.

Информация, с точки зрения информационной безопасности, характеризуется следующими категориями:

- *конфиденциальность* – гарантия того, что конкретная информация является доступной только тому кругу лиц, для которого она предназначена; нарушение этой категории является *похищением* или *раскрытием информации;*

- *целостность* – гарантия того, что на данный момент информация существует в изначальном виде, то есть, при ее хранении или передаче не было сделано несанкционированных изменений; нарушение данной категории называется *фальсификацией сообщения;*

- *аутентичность* – гарантия того, что источником информации является именно тот субъект, который заявлен как ее автор; нарушение этой категории называется *фальсификацией автора сообщения;*

- *апеллированность* – гарантия того, что в случае необходимости можно доказать, что автором сообщения является именно заявленный субъект и никто другой; отличие данной категории от предыдущей в том, что при подмене автора, кто-то другой пытается заявить об авторстве сообщения, а при нарушении апеллированности сам автор пытается избежать ответственности за выданное им сообщение.

Относительно информационных систем используются другие категории:

- *надежность* – гарантия того, что система будет вести себя запланированно как в нормальном, так и во внештатном режимах;

- *точность* – гарантия точного и полного исполнения всех команд;

- *контроль доступа* - гарантия того, что разные группы лиц имеют разный доступ к информационным объектам, и эти ограничения доступа постоянно исполняются;

- *контролируемость* – гарантия того, что в любой момент может быть выполнена полноценная проверка любого компонента программного комплекса;

-*контроль идентефикации* – гарантия того, что клиент (адресат) является именно тем, за кого себя выдаёт.

Потенциально возможные сферы использования стеганографии указаны на рисунке 11.1.

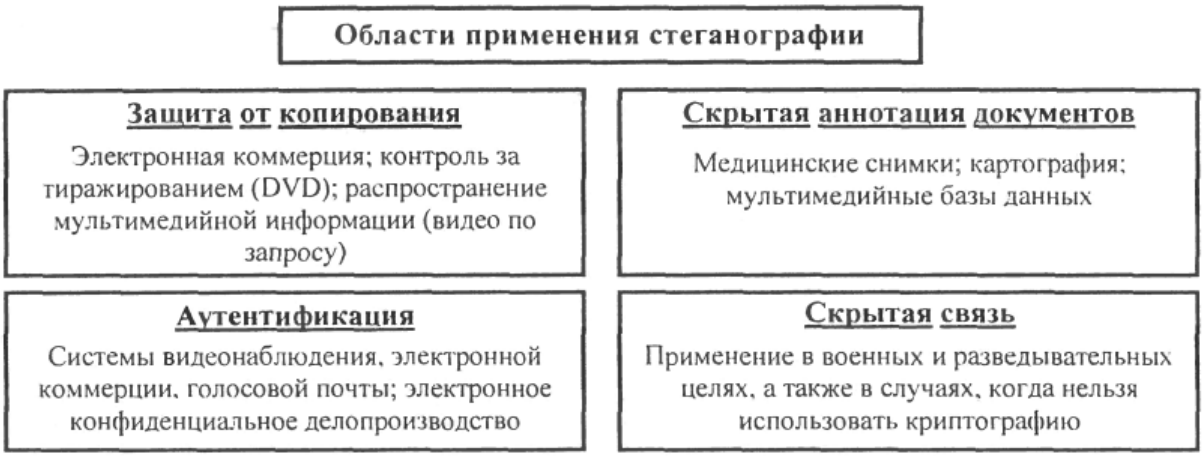


Рисунок 11.1 – Потенциальные области использования стеганографии

11.2. Структурная схема стеганосистемы

В общем случае стеганосистема может быть рассмотрена как система связи [21]. Обобщенная структурная схема стеганосистемы изображена на рисунке 11.2

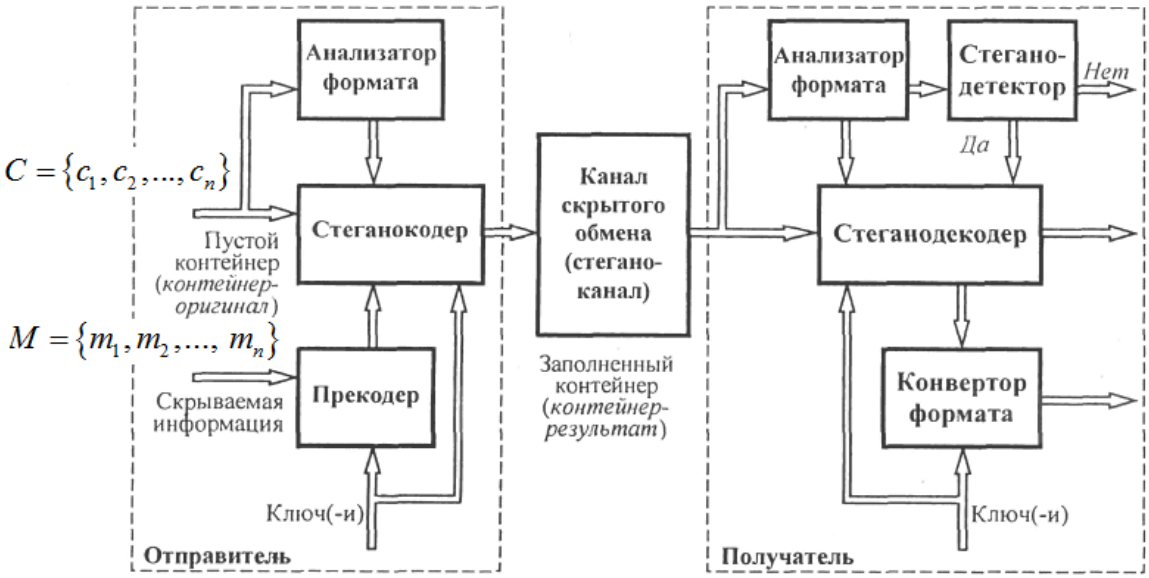


Рисунок 11.2 - Структурная схема стеганосистемы как системы связи.

Основными стеганографическими понятиями являются сообщение и контейнер. Сообщение $m \in M$ – это секретная информация, наличие которой необходимо скрыть, $M = \{m_1, m_2, ..., m_n\}$ – множество всех сообщений.

Контейнером $c \in C$ называется несекретная информация, которую можно использовать для скрытия сообщения, C = {c_1, c_2,..., c_q} – множество всех контейнеров, причем q » n. В качестве сообщения и контейнера могут выступать как обычный текст, так и файлы мультимедийного формата.

Пустой контейнер (или так называемый контейнер-оригинал)– это контейнер c, который не содержит скрытой информации. Заполненный контейнер {контейнер-результат)– контейнер c, который содержит скрытую информацию m (c_m). Одно из требований, которое при это ставится: контейнер-результат не должен быть визуально отличим от контейнера-оригинала. Выделяют два основных типа контейнера: *потоковый* и *фиксированный.*

Потоковый контейнер представляет собой последовательность битов, которая непрерывно изменяется. Сообщение встраивается в него в реальном масштабе времени, поэтому в кодере заранее неизвестно, хватит ли размеров контейнера для передачи всего сообщения. В один контейнер большого размера может быть встроено несколько сообщений. Интервалы между встроенными битами определяются генератором ПСП с равномерным распределением интервалов между отсчетами.

Основная проблема заключается в выполнении синхронизации, определении начала и конца последовательности. Если в данных контейнера существуют биты синхронизации, заголовки пакетов и т.д., то скрытая информация может следовать сразу же после них. Сложность организации синхронизации является преимуществом с точки зрения обеспечения скрытости передачи. К сожалению, на сегодняшний день практически отсутствуют работы, посвященные разработке стеганосистем с потоковым контейнером.

В качестве примера перспективной реализации потокового контейнера можно привести стеганоприставку к обычному телефону. При этом под прикрытием заурядного, несущественного телефонного разговора можно передавать другой разговор, данные и т.д. Не зная секретного ключа, нельзя не только узнать о содержании скрытой передачи, но и о самом факте ее существования.

В *фиксированном контейнере* размеры и характеристики последнего заранее известны. Это позволяет выполнять вложение данных оптимальным (в определенном смысле) образом. Далее будут рассматриваться преимущественно фиксированные контейнеры (в дальнейшем – просто "контейнеры").

Контейнер может быть избранным, случайным или навязанным. Избранный контейнер зависит от встроенного сообщения, а в предельном случае является его функцией. Такой тип контейнера больше характерен именно для стеганографии. Навязанный контейнер появляется, когда лицо, которое предоставляет контейнер, подозревает о возможной скрытой переписке и желает предотвратить ее. На практике же чаще всего имеют дело со случайным контейнером [5].

Скрытие информации, которая преимущественно имеет большой объем, выдвигает существенные требования к контейнеру, размер которого должен по меньшей мере в несколько раз превышать размер встраиваемых данных. Понятно, что для увеличения скрытости указанное соотношение должно быть как можно большим.

Перед тем как выполнить вложение сообщения в контейнер, его необходимо преобразовать в определенный удобный для упаковки вид. Кроме того, перед упаковкой в контейнер, для повышения защищенности секретной информации последнюю можно зашифровать достаточно устойчивым криптографическим кодом [14].

Во многих случаях также желательна устойчивость полученного стеганосообщения к искажениям (в том числе и злоумышленным) [21].

В процессе передачи звук, изображение или какая-либо другая информация, используемая в качестве контейнера, может подвергаться разным трансформациям (в том числе с использованием алгоритмов с потерей данных): изменение объема, преобразование в другой формат и т.п.. – поэтому для сохранения целостности встроенного сообщения может понадобиться использование кода с исправлением ошибок (помехоустойчивое кодирование).

Начальную обработку скрываемой информации выполняет изображенный на рисунке 11.2 прекодер. В качестве одной из важнейших предварительных обработок сообщения (а также и контейнера) можно назвать вычисление его обобщенного преобразования Фурье. Это позволяет осуществить встраивание данных в спектральной области, что значительно повышает их устойчивость к искажениям.

Следует отметить, что для увеличения секретности встраивания, предварительная обработка довольно часто выполняется с использованием ключа.

Упаковка сообщения в контейнер (с учетом формата данных, представляющих контейнер) выполняется с помощью стеганокодера. Вложение проис-

ходит, например, путем модификации наименьших значащих битов контейнера. Вообще, именно алгоритм (стратегия) внесения элементов сообщения в контейнер определяет методы стеганографии, которые в свою очередь делятся на определенные группы, например, в зависимости от того, файл какого формата был выбран в качестве контейнера.

В большинстве стеганосистем для упаковки и извлечения сообщений используется ключ, который предопределяет секретный алгоритм, определяющий порядок внесения сообщения в контейнер. По аналогии с криптографией, тип ключа обуславливает существование двух типов стеганосистем:

-с секретным ключом – используется один ключ, который определяется до начала обмена стеганограммой или передается защищенным каналом;

-с открытым ключом – для упаковки и распаковки сообщения используются разные ключи, которые отличаются таким образом, что с помощью вычислений невозможно получить один ключ из другого, поэтому один из ключей (открытый) может свободно передаваться по незащищенному каналу.

В качестве секретного алгоритма может быть использован генератор псевдослучайной последовательности (ПСП) битов.

Скрываемая информация заносится в соответствии с ключом в те биты, модификация которых не приводит к существенным искажениям контейнера. Эти биты образуют так называемый *стеганопуть.* Под "существенным" подразумевается искажение, которое приводит к росту вероятности выявления факта наличия скрытого сообщения после проведения стеганоанализа.

Стеганографический канал – канал передачи контейнера-результата (вообще, существование канала как, собственно говоря, и получателя – наиболее обобщенный случай, поскольку заполненный контейнер может, например, храниться у "отправителя", который поставил перед собой цель ограничить неавторизованный доступ к определенной информации. В данном случае отправитель выступает в роли получателя). Во время пребывания в стегано графическом канале контейнер, содержащий скрытое сообщении, может подвергаться умышленным атакам или случайным помехам.

В *стеганодетекторе* определяется наличие в контейнере (возможно уже измененном) скрытых данных. Это изменение может быть обусловлено влиянием ошибок в канале связи, операций обработки сигнала, намеренных атак нарушителей.

Различают стеганодетекторы, предназначенные только для обнаружения факта наличия встроенного сообщения, и устройства, предназначенные для выделения этого сообщения из контейнера, – *стеганодекодеры.*

11.3. Классификация методов скрытых данных

Подавляющее большинство методов цифровой (компьютерной) стеганографии (КС) базируется на двух ключевых принципах:

- файлы, которые не требуют абсолютной точности (например, файлы с изображением, звуковой информацией и т.д.), могут быть видоизменены (конечно, до определенной степени) без потери своей функциональности;

- органы чувств человека неспособны надежно различать незначительные изменения в модифицированных таким образом файлах и/или отсутствует специальный инструментарий, который был бы способен выполнять данную задачу.

Для существующих методов компьютерной стеганографии вводят следующую классификацию (рисунок 11.3).

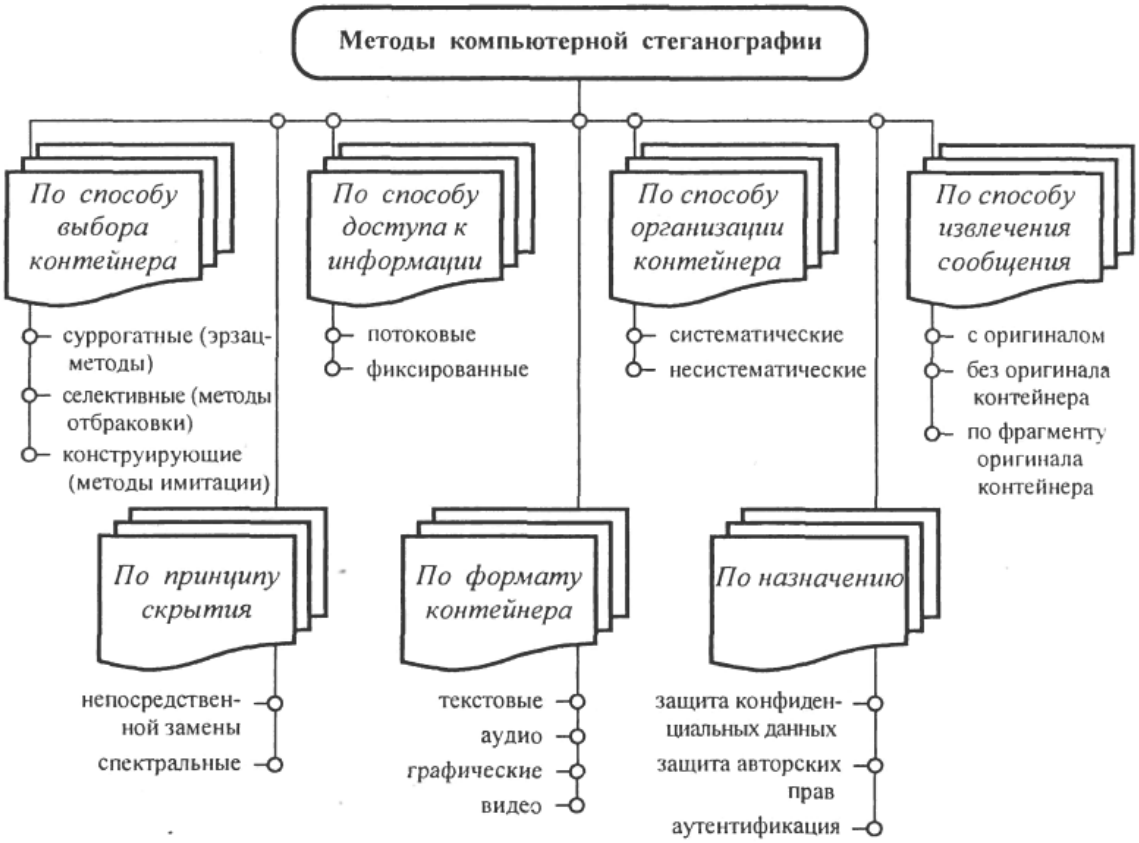


Рисунок 11.3 - Классификация методов компьютерной стеганографии.

Как видно из рисунка 11.3, по способу выбора контейнера различают суррогатные (или так называемые эрзац-методы), селективные и конструирующие методы стеганографии.

В **суррогатных (безальтернативных)** методах стеганографии полностью отсутствует возможность выбора контейнера, и для скрытия сообщения избирается первый попавшийся контейнер, – эрзац-контейнер, – который в большинстве случаев не оптимален для скрытия сообщения заданного формата.

В **селективных** методах КС предусматривается, что скрытое сообщение должно воспроизводить специальные статистические характеристики шума контейнера. Для этого генерируют большое количество альтернативных контейнеров с последующим выбором (путем отбраковки) наиболее оптимального из них для конкретного сообщения. Особым случаем такого подхода является

вычисление некоторой хэш-функции для каждого контейнера. При этом для скрытия сообщения избирается тот контейнер, хэш-функция которого совпадает со значением хэш-функции сообщения (то есть стеганограммой является избранный контейнер).

В **конструирующих** методах стеганографии контейнер генерируется самой стеганосистемой. При этом существует несколько вариантов реализации. Так, например, шум контейнера может имитироваться скрытым сообщением. Это реализуется с помощью процедур, которые не только кодируют скрываемое сообщение под шум. но и сохраняют модель изначального шума. В предельном случае по модели шума может строиться целое сообщение.

По способу доступа к скрываемой информации различают методы для *потоковых (беспрерывных)* контейнеров и методы для фиксированных (ограниченной длины) контейнеров (более подробно см. подраздел 11.2).

По способу организации контейнеры, подобно помехоустойчивым кодам, могут быть систематическими и несистематическими. В первых можно указать конкретные места стеганограммы, где находятся информационные биты собственно контейнера, а где– шумовые биты, предназначенные для скрытия информации (как, например, в широко распространенном методе наименее значащего бита). В случае несистематической организации контейнера такое разделение невозможно. В этом случае для выделения скрытой информации необходимо обрабатывать содержимое всей стеганограммы.

По используемому принципу скрытия методы компьютерной стеганографии делятся на два основных класса: *методы непосредственной* замены и *спектральные* методы. Если первые, используя избыток информационной среды в пространственной (для изображения) или временной (для звука) области, заключаются в замене малозначительной части контейнера битами секретного сообщения, то другие для скрытия данных используют спектральные представления элементов среды, в которую встраиваются скрываемые данные (например, в разные коэффициенты массивов дискретно-косинусных преобразований, преобразований Фурье, Каруне- на-Лоева, Адамара, Хаара и т.д.).

Основным направлением компьютерной стеганографии является использование свойств именно избыточности контейнера-оригинала, но при этом следует принимать во внимание то, что в результате скрытия информации происходит искажение некоторых статистических свойств контейнера или же нарушение его структуры. Это необходимо учитывать для уменьшения демаскирующих признаков.

В особую группу можно также выделить методы, которые **используют специальные свойства форматов представления файлов**:

- зарезервированные для расширения поля файлов, которые зачастую заполняются нулями и не учитываются программой;

- специальное форматирование данных (сдвиг слов, предложений, абзацев или выбор определенных позиций символов);

- использование незадействованных участков на магнитных и оптических носителях;

удаление файловых заголовков-идентификаторов и т.д.

В основном, для таких методов характерны низкая степень скрытости, низкая пропускная способность и слабая производительность.

По назначению различают стеганометоды собственно для скрытой передачи (или скрытого хранения) данных и методы для скрытия данных в цифровых объектах с целью защиты авторских прав на них.

По типам контейнера выделяют стеганографические методы с контейнерами в виде текста, аудиофайла, изображения и видео.

Рассмотрим подробнее стеганографические методы скрытия данных в неподвижных изображениях, в аудиосигналах и в текстовых файлах.

11.4. Скрытие данных в неподвижных изображениях

Большинство исследований посвящено использованию в качестве стеганоконтейнеров именно изображений. Это обусловлено следующими причинами:

- существованием практической необходимости защиты цифровых фотографий, изображений, видео от противозаконного тиражирования и распространения;

- относительно большим объемом цифрового представления изображений, что позволяет встраивать цифровые водяные знаки (ЦВЗ) значительного объема или же повышать устойчивость этого встраивания;

- заранее известным (фиксированным) размером контейнера, отсутствием ограничений, которые накладываются требованиями скрытия в реальном времени:

- наличием в большинстве реальных изображений текстурных областей, имеющих шумовую структуру и наилучшим образом подходящих для встраивания информации;

- слабой чувствительностью человеческого глаза к незначительным изменениям цветов изображения, его яркости, контрастности, содержания в нем шума, искажений вблизи контуров;

- наконец, хорошо разработанными в последнее время методами цифровой обработки изображений.

Однако, как указывается в [22], последняя причина вызывает и значительные трудности в обеспечении стойкости ЦВЗ: чем более совершенными становятся методы компрессии, тем меньше остается возможностей для встраивания посторонней информации.

Развитие теории и практики алгоритмов компрессии изображений привело к изменению представлений о технике встраивания ЦВЗ. Если сначала предлагалось встраивать информацию в незначащие биты для уменьшения визуальной заметно- сти, то современный подход, наоборот, заключается во встраивании ЦВЗ в наиболее существенные области изображений, разрушение которых будет приводить к полной деградации самого изображения. Поэтому абсолютно

понятна необходимость учета стеганоалгоритмами не только алгоритмов компрессии изображений, но и свойств зрительной системы человека (ЗСЧ).

В последнее время создано достаточное количество методов скрытия данных в цифровых изображениях, что позволяет провести их классификацию и выделить следующие обобщенные группы [22]:

методы замены в пространственной области;

- методы скрытия в частотной области изображения;
- широкополосные методы;
- статистические (стохастические) методы;
- методы искажения;
- структурные методы.

11.4.1. Скрытие данных в пространственной области

Алгоритмы, описанные в данном подразделе, встраивают скрываемые данные в области первичного изображения. Их преимущество заключается в том, что для встраивания нет необходимости выполнять вычислительно сложные и длительные преобразования изображений.

Цветное изображение С будем представлять через дискретную функцию, которая определяет вектор цвета $c(x, y)$ для каждого пикселя изображения (x, y), где значение цвета задает трехкомпонентный вектор в цветовом пространстве. Наиболее распространенный способ передачи цвета – это модель RGB, в которой основные цвета – красный, зеленый и синий, а любой другой цвет может быть представлен в виде взвешенной суммы основных цветов.

Вектор цвета $c(x,y)$ в RGB-пространстве представляет интенсивность основных цветов. Сообщения встраиваются за счет манипуляций цветовыми составляющими {R(x,y), G(x,y), B(x,y)} или непосредственно яркостью $\lambda(x,y) \in \{0, 1, 2,..., L_c\}$.

Общий принцип этих методов заключается в замене избыточной, малозначимой части изображения битами секретного сообщения. Для извлечения сообщения необходимо знать алгоритм, по которому размещалась по контейнеру скрытая информация.

11.4.1.1. Метод замены наименее значащего бита. Метод замены наименее значащего бита (НЗБ, LSB– Least Significant Bit) наиболее распространен среди методов замены в пространственной области [22,23].

Младший значащий бит изображения несет в себе меньше всего информации. Известно, что человек в большинстве случаев не способен заметить изменений в этом бите. Фактически, НЗБ – это шум, поэтому его можно использовать для встраивания информации путем замены менее значащих битов пикселей изображения битами секретного сообщения. При этом, для изображения в градациях серого (каждый пиксель изображения кодируется одним байтом) объем встроенных данных может составлять 1/8 от общего объема контейнера. Например, в изображение размером 512x512 можно встроить ~32 кБайт информации. Если же модифицировать два младших бита (что также практически незаметно), то данную пропускную способность можно увеличить еще вдвое.

Популярность данного метода обусловлена его простотой и тем, что он позволяет скрывать в относительно небольших файлах достаточно большие объемы информации (пропускная способность создаваемого скрытого канала связи составляет при этом от 12,5 до 30%). Метод зачастую работает с растровыми изображениями, представленными в формате без компрессии (например, GIF и BMP).

Метод НЗБ имеет низкую стеганографическую стойкость к атакам пассивного и активного нарушителей. Основной его недостаток – высокая чувствительность к малейшим искажениям контейнера. Для ослабления этой чувствительности часто дополнительно применяют помехоустойчивое кодирование.

11.4.1.2 Метод псевдослучайного интервала. В рассмотренном выше простейшем случае выполняется замена НЗБ всех последовательно размещенных пикселей изображения. Другой подход – *метод случайного интервала* [21], заключается в случайном распределении битов секретного сообщения по контейнеру, в результате чего расстояние между двумя встроенными битами определяется псевдослучайно. Эта методика особенно эффективна в случае, когда битовая длина секретного сообщения существенно меньше количества пикселей изображения.

11.4.1.3 Метод псевдослучайной перестановки. Недостатком метода псевдослучайного интервала является то, что биты сообщения в контейнере размещены в той же последовательности, что и в самом сообщении, и только интервал между ними изменяется псевдослучайно. Поэтому для контейнеров фиксированного размера более целесообразным является использование метода псевдослучайной перестановки (выбора) [21], смысл которого заключается в том, что генератор ПСЧ образует последовательность индексов $j_1, j_2, ..., j_{l_M}$ и сохраняет k-й бит сообщения в пикселе с индексом j_k.

Пусть N – общее количество бит (самых младших) в имеющемся контейнере; P^N – перестановка чисел $\{1,2, ..., N\}$. Тогда, если у нас имеется для скрытия конфиденциальное сообщение длиной n бит, то эти биты можно просто встроить вместо бит контейнера $P^N(1), P^N(2), ..., P^N(n)$.

Функция перестановки должна быть псевдослучайной, иными словами, она должна обеспечивать выбор бит контейнера приблизительно случайным образом. Таким образом, секретные биты будут равномерно распределены по всему битовому пространству контейнера.

11.4.1.4 Метод блочного скрытия. ***Метод блочного скрытия*** – это еще один подход к реализации метода замены и заключается в следующем [21]. Изображение-оригинал разбивается на l_M непересекающихся блоков $\Delta_i (1 \le i \le l_M)$ произвольной конфигурации, для каждого из которых вычисляется бит четности $b(\Delta_i)$:

$$b(\Delta_i) = \sum_{j \in \Delta_j}^{\mathrm{mod}\,2} LSB(C_j). \qquad (5.4)$$

В каждом блоке выполняется скрытие одного секретного бита M_i. Если бит четности $b(\Delta_i) \neq M_i$, то происходит инвертирование одного из НЗБ блока Δ_i, в результате чего $b(\Delta_i) = M_i$. Выбор блока может происходить псевдослучайно с использованием стеганоключа.

Хотя этот метод имеет такую же низкую устойчивость к искажениям, как и все предыдущие, у него есть ряд преимуществ. Во-первых, существует возможность модифицировать значение такого пикселя в блоке, изменение которого приведет к минимальному изменению статистики контейнера. Во-вторых, влияние последствий встраивания секретных данных в контейнер можно уменьшить за счет увеличения размера блока.

11.4.1.5 Методы замены палитры. Для скрытия данных можно также воспользоваться палитрой цветов, присутствующих в формате изображения [21]. Палитра из N цветов определяется как список пар индексов (i, Λ_i), который определяет соответствие между индексом i и его вектором цветности Λ_i, (так называемая таблица цветов). Каждому пикселю изображения ставится в соответствие определенный индекс в таблице. Поскольку порядок цветов в палитре не важен для восстановления общего изображения, конфиденциальная информация может быть скрыта путем перестановки цветов в палитре.

Существует $N!$ различных способов перестановки N-цветной палитры, чего вполне достаточно для скрытия небольшого сообщения. Однако методы скрытия, в основе которых лежит порядок формирования палитры, также являются неустойчивыми: любая атака, связанная со сменой палитры, уничтожает встроенное сообщение.

11.4.1.6 Метод квантования изображения. К методам скрытия в пространственной области можно также отнести метод ***квантования изображения*** [21], основанный на межпиксельной зависимости, которую можно описать некоторой функцией Θ. В простейшем случае можно вычислить разницу ε_i между смежными пикселями c_{i-} и $ci+1$ (или c_{i-1} и c_i) и задать ее как параметр функции Θ: $\Delta_i = \Theta (c_i - c_{i+1})$, где Δ,– дискретная аппроксимация разницы сигналов $c_i - c_{i+1}$.

Поскольку Δ, – целое число, а реальная разница $c_i - c_{i+1}$ – действительное число, то возникают ошибки квантования $\delta_i = \Delta_i - \varepsilon_i$. Для сильно коррелированных сигналов эта ошибка близка к нулю: $\delta_i \approx 0$.

При данном методе скрытие информации производится путем корректировки разностного сигнала Δ_i. Стеганоключ представляет собой таблицу, которая каждому возможному значению Δ_i ставит в соответствие определенный бит, например:

Δ_i	-4	-3	-2	-1	0	1	2	3	4
b_i	1	0	1	1	0	0	1	0	1

Для скрытия **i**-го бита сообщения вычисляется разница Δ_i. Если при этом b_i не соответствует секретному биту, который необходимо скрыть, то значение Δ_i заменяется ближайшим Δ_j, для которого такое условие выполняется. При

этом соответствующим образом корректируются значения интенсивностей пикселей, между которыми вычислялась разница Δ_i. Извлечение секретного сообщения осуществляется согласно значению b^*_i, соответствующему разнице Δ^*_i.

11.4.2. Скрытие данных в частотной области изображения

Как уже было отмечено выше, стеганографические методы скрытия данных в пространственной области изображения являются нестойкими к большинству из известных видов искажений. Так, например, использование операции компрессии с потерями (относительно изображения, это может быть JPEG-компрессия) приводит к частичному или, что более вероятно, полному уничтожению встроенной в контейнер информации. Более стойкими к разнообразным искажениям, в том числе и компрессии, являются методы, использующие для скрытия данных не пространственную область контейнера, а частотную.

Существует несколько способов представления изображения в частотной области.

Наибольшее распространение среди всех ортогональных преобразований в стеганографии получили вейвлет-преобразования и ДКП [22], что определенной мерой объясняется значительным распространением их использования при компрессии изображений. Кроме того, для скрытия данных целесообразно применять именно то преобразование изображения, которому последнее будет подвергаться со временем при возможной компрессии. Например, известно, что алгоритм ДКП является базовым в стандарте JPEG, а вейвлет-преобразования – в стандарте JPEG2000.

Структурная схема стеганосистемы приведена на рисунке 11.4.

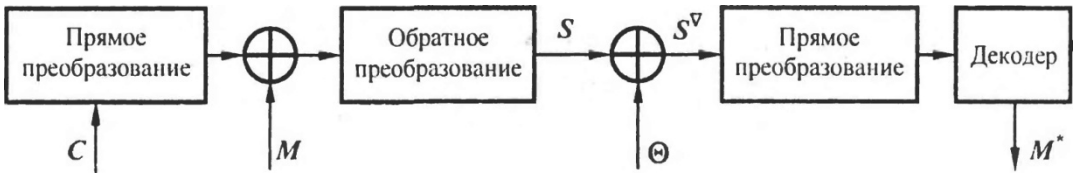


Рисунок 11.4 - Структурная схема стеганосистемы при наличии в стеганоканале атаки компрессии

Первичное зображение ***C*** раскладывается на ***D*** субполос (прямое преобразование), в каждую из которых встраивается скрываемая информация **M**. После обратного преобразования получается модифицированное изображение ***S.*** После компрессии/декомпрессии Θ в канале связи получается изображение S∇, которое на принимающей стороне вновь подвергается прямому преобразованию и из каждой субполосы ***D*** независимо извлекается скрытое сообщение – оценка **M** .

Известно, и данный факт используется в алгоритмах компрессии, что большая часть энергии изображений сосредоточена в низкочастотной (НЧ) области спектра. Отсюда и возникает необходимость в осуществлении декомпо-

зиции изображения на субполосы, к которым прибавляется стеганособщение. НЧ субполосы содержат основную часть энергии изображения и, таким образом, носят шумовой характер. Высокочастотные (ВЧ) субполосы спектра изображения наибольшим образом поддаются влиянию со стороны разнообразных алгоритмов обработки, таких как, например, компрессия или НЧ-фильтрация. Таким образом, можно сделать вывод, что для встраивания сообщения самыми оптимальными являются среднечастотные (СЧ) субполосы спектра изображения.

Во время цифровой обработки изображения часто применяется двумерная версия дискретного косинусного преобразования:

$$\Omega(\upsilon,\nu)=\frac{\zeta(\upsilon)\cdot\zeta(\nu)}{\sqrt{2N}}\cdot\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}C(x,y)\cdot\cos\left[\frac{\pi\cdot\upsilon\cdot(2x+1)}{2N}\right]\cdot\cos\left[\frac{\pi\cdot\nu\cdot(2y+1)}{2N}\right]; \qquad (11.1)$$

$$S(x,y)=\frac{1}{\sqrt{2N}}\cdot\sum_{\upsilon=0}^{N-1}\sum_{\nu=0}^{N-1}\zeta(\upsilon)\cdot\zeta(\nu)\,\Omega(\upsilon,\nu)\cdot\cos\left[\frac{\pi\cdot\upsilon\cdot(2x+1)}{2N}\right]\cdot\cos\left[\frac{\pi\cdot\nu\cdot(2y+1)}{2N}\right], \qquad (11.2)$$

где **C(x**, ***y)*** и ***S{x, y)*** – соответственно, элементы оригинального и восстановленного по коэффициентам ДКП изображения размерностью ***NxN; x,y*** – пространственные координаты пикселей изображения; **Ω(υ,ν)** – массив коэффициентов ДКП; **υ,ν** – координаты в частотной области; **ζ(ν) =1/**$\sqrt{2}$, если **ν** = 0, и **ζ(ν)** = 1, если **ν** > 0.

Рассмотрим существующие методы, которые базируются на алгоритме ДКП.

11.4.2.1 Метод относительной замены величин коэффициентов ДКП (метод Коха и Жао). Один из наиболее распространенных на сегодня методов скрытия конфиденциальной информации в частотной области изображения заключается в ***относительной замене величин коэффициентов ДКП***[21].

На начальном этапе первичное изображение разбивается на блоки размерностью 8x8 пикселей. ДКП применяется к каждому блоку – формула (11.1), в результате чего получают матрицы 8x8 коэффициентов ДКП, которые зачастую обозначают $\Omega_b(\upsilon,\nu)$, где b – номер блока контейнера **C**, а (**υ,ν**)– позиция коэффициента в этом блоке. Каждый блок при этом предназначен для скрытия одного бита данных.

Было предложено две реализации алгоритма: псевдослучайно могут выбираться два или три коэффициента ДКП. Рассмотрим первый вариант.

Во время организации секретного канала абоненты должны предварительно договориться о двух конкретных коэффициентах ДКП из каждого блока, которые будут использоваться для скрытия данных. Зададим данные коэффициенты их координатами в массивах коэффициентов ДКП: (υ_1,ν_1) и (υ_2, ν_2). Кроме этого, указанные коэффициенты должны отвечать косинус-функциям со средними частотами, что обеспечит скрытость информации в существенных для ЗСЧ областях сигнала, к тому же информация не будет искажаться при JPEG-компрессии с малым коэффициентом сжатия.

Непосредственно процесс скрытия начинается со случайного выбора блока C_b изображения, предназначенного для кодирования **b**-го бита сообщения. Встраивание информации осуществляется таким образом: для передачи бита "0"стремятся, чтобы разница абсолютных значений коэффициентов ДКП превышала некоторую положительную величину, а для передачи бита "1" эта разница делается меньшей по сравнению с некоторой отрицательной величиной:

$$\begin{cases} \left|\Omega_b(\upsilon_1,\nu_1)\right| - \left|\Omega_b(\upsilon_2,\nu_2)\right| > P, при \;\; m_b = 0; \\ \left|\Omega_b(\upsilon_1,\nu_1)\right| - \left|\Omega_b(\upsilon_2,\nu_2)\right| < -P, при \;\; m_b = 1. \end{cases} \qquad (11.3)$$

Таким образом, первичное изображение искажается за счет внесения изменений в коэффициенты ДКП, если их относительная величина не отвечает скрываемому биту. Чем больше значение **P** тем стеганосистема, созданная на основе данного метода. является более стойкой к компрессии, однако качество изображения при этом значительно ухудшается.

После соответствующего внесения коррекции в значения коэффициентов, которые должны удовлетворять неравенству (11.3), проводится обратное ДКП.

Для извлечения данных, в декодере выполняется аналогичная процедура выбора коэффициентов, а решение о переданном бите принимается в соответствии со следующим правилом:

$$\begin{cases} m_b^* = 0, \; при \;\; \left|\Omega_b^*(\upsilon_1,\nu_1)\right| > \left|\Omega_b^*(\upsilon_2,\nu_2)\right|; \\ m_b^* = 1, \; при \;\; \left|\Omega_b^*(\upsilon_1,\nu_1)\right| < \left|\Omega_b^*(\upsilon_2,\nu_2)\right|. \end{cases} \qquad (11.4)$$

11.4.2.2 Метод Бенгама-Мемона-Эо-Юнг. Бенгам (D. Benham), Мемон (N. Метоп), Эо (B.-L. Yeo) и Юнг (Minerva Yeung) [29] предложили оптимизированную версию вышерассмотренного метода. Причем оптимизация была проведена ими по двум направлениям: во-первых, было предложено для встраивания использовать не все блоки, а только наиболее подходящие для этого, во-вторых, в частотной области блока для встраивания выбираются не два, а три коэффициента ДКП, что, как будет показано в дальнейшем, существенно уменьшает визуальные искажения контейнера. Рассмотрим отмеченные усовершенствования более подробно.

Пригодными для встраивания информации считаются такие блоки изображения, которые одновременно удовлетворяют следующим двум требованиям:

-блоки не должны иметь резких переходов яркости;

-блоки не должны быть слишком монотонными.

Блоки, которые не отвечают первому требованию, характеризуются наличием слишком больших значений низкочастотных коэффициентов ДКП, сопоставимых по своей величине с DC-коэффициентом. Для блоков, которые не удовлетворяют второму требованию, характерно равенство нулю большинства

высокочастотных коэффициентов. Указанные особенности являются критерием отбраковки непригодных блоков.

Отмеченные требования отбраковки учитываются использованием двух пороговых коэффициентов: $\mathbf{P_L}$ (для первого требования) и $\mathbf{P_H}$ (для второго требования), превышение ($\mathbf{P_L}$) или недостижение ($\mathbf{P_H}$) которых будет указывать на то, что рассматриваемый блок не пригоден для модификации в частотной области.

Встраивание в блок бита сообщения совершается следующим образом. Выбираются (для большей стойкости стеганосистемы – псевдослучайно) три коэффициента ДКП блока из среднечастотной области с координатами ($\mathbf{\upsilon_1,\nu_1}$) ,($\mathbf{\upsilon}_2$, $\mathbf{\nu}_2$) и ($\mathbf{\upsilon}_3$, $\mathbf{\nu}_3$). Если необходимо провести встраивание "0", эти коэффициенты изменяются таким образом (если, конечно, это необходимо), чтобы третий коэффициент стал меньше любого из первых двух; если необходимо скрыть “1”, он делается большим по сравнению с первым и вторым коэффициентами:

$$\begin{cases} \left.\begin{aligned} &|\Omega_b(\upsilon_3,\nu_3)| < |\Omega_b(\upsilon_1,\nu_1)|; \\ &|\Omega_b(\upsilon_3,\nu_3)| < |\Omega_b(\upsilon_2,\nu_2)|. \end{aligned}\right\}, при \;\; m_b = 0; \\ \left.\begin{aligned} &|\Omega_b(\upsilon_3,\nu_3)| > |\Omega_b(\upsilon_1,\nu_1)|; \\ &|\Omega_b(\upsilon_3,\nu_3)| > |\Omega_b(\upsilon_2,\nu_2)|. \end{aligned}\right\}, при \;\; m_b = 1. \end{cases} \qquad (11.5)$$

Как и в предыдущем методе, для принятия решения о достаточности различения указанных коэффициентов ДКП, в выражение (11.5) вводится значение порога различения **P**:

$$\begin{cases} |\Omega_b(\upsilon_3,\nu_3)| < \min\left(|\Omega_b(\upsilon_1,\nu_1)|, |\Omega_b(\upsilon_2,\nu_2)|\right) - P, при \;\; m_b = 0; \\ |\Omega_b(\upsilon_3,\nu_3)| > \max\left(|\Omega_b(\upsilon_1,\nu_1)|, |\Omega_b(\upsilon_2,\nu_2)|\right) - P, при \;\; m_b = 1. \end{cases} \qquad (11.6)$$

В том случае, если такая модификация приводит к слишком большой деградации изображения, коэффициенты не изменяют, и блок в качестве контейнера не используется.

Использование трех коэффициентов вместо двух и, что самое главное, отказ от модификации блоков изображения в случае неприемлемых их искажений, уменьшает погрешности, которые вносятся сообщением. Получатель всегда может определить блоки, в которые не проводилось встраивание, просто повторив анализ, аналогичный выполненному на передающей стороне.

11.4.2.3 Метод Хсу и Ву. Хсу (Chiou-Ting Hsu) и Ву (Ja-Ling Wu) [21] был предложен алгоритм встраивания цифрового водяного знака в массив коэффициентов ДКП блоков изображения-контейнера. Приведем основные положения, заложенные авторами в основ) алгоритма.

Пусть ***C*** – полутоновое изображение размером ***X x Y,*** а ***W*** – ЦВЗ, который представляет собой двоичное изображение размером ***A x Z. B*** ЦВЗ пиксель может принимать значение или «1», или «0». Разумеется, что непосредственное

наблюдение такого изображения невозможно, поскольку интенсивности 0 и 1 отвечают черному цвету (последняя – в некотором приближении). Изображение ЦВЗ можно создать черно-белым, а перед скрытием заменить интенсивность белых пикселей (255) на единицу, например, путем деления всего массива ЦВЗ на 255. При извлечении, наоборот, для визуального наблюдения массив ЦВЗ необходимо умножить на 255.

11.4.2.4 Метод Фридрих. Алгоритм, предложенный Джессикой Фридрих (J. Fridrich) [106]. по сути является комбинацией двух алгоритмов: в соответствии с одним из них скрываемые данные встраиваются в низкочастотные, а с другим – в среднечастотные коэффициенты ДКП. Как было показано автором, каскадное использование двух разных алгоритмов позволяет получить хорошие результаты относительно стойкости стеганографической системы к атакам.

11.4.3. Методы расширения спектра

Изначально методы расширения спектра (РС или SS – *Spread-Spectrum)* использовались при разработке военных систем управления и связи. Во время Второй мировой войны расширение спектра использовалось в радиолокации для борьбы с намеренными помехами. В последние годы развитие данной технологии объясняется желанием создать эффективные системы радиосвязи для обеспечения высокой помехоустойчивости при передаче узкополосных сигналов по каналам с шумами и осложнения их перехвата. Система связи является системой с расширенным спектром в следующих случаях [25]

- Полоса частот, которая используется при передаче, значительно шире минимально необходимой для передачи текущей информации. При этом энергия информационного сигнала расширяется на всю ширину полосы частот при низком соотношении сигнал/шум, в результате чего сигнал трудно обнаружить, перехватить или воспрепятствовать его передаче путем внесения помех. Хотя суммарная мощность сигнала может быть большой, соотношение сигнал/шум в любом диапазоне частот является малым, что делает сигнал с расширенным спектром трудно определяемым при радиосвязи и, в контексте скрытия информации стеганографическими методами, трудно различимым человеком.

-Расширение спектра выполняется с помощью так называемого расширяющего (или кодового) сигнала, который не зависит от передаваемой информации. Присутствие энергии сигнала во всех частотных диапазонах делает радиосигнал с расширенным спектром стойким к внесению помех, а информацию, встроенную в контейнер методом расширения спектра, стойкой к ее устранению или извлечению из контейнера. Компрессия и другие виды атак на систему связи могут устранить энергию сигнала из некоторых участков спектра, но поскольку последняя была распространена по всему диапазону, в других полосах остается достаточное количество данных для восстановления информации. В результате, если, разумеется, не разглашать ключ, который использовался для генерации кодового сигнала, вероятность извлечения информации неавторизованными лицами существенно снижается.

-Восстановление первичной информации (то есть "сужение спектра") осуществляется путем сопоставления полученного сигнала и синхронизированной копии кодового сигнала.

-В радиосвязи применяют три основных способа расширения спектра:

-с помощью прямой ПСП (РСПП);

-с помощью скачкообразного перестраивания частот;

-с помощью компрессии с использованием линейной частотной модуляции (ЛЧМ).

При расширении спектра прямой последовательностью информационный сигнал модулируется функцией, которая принимает псевдослучайные значения в установленных пределах, и умножается на временную константу– частоту (скорость) следования элементарных посылок (элементов сигнала). Данный псевдослучайный сигнал содержит составляющие на всех частотах, которые, при их расширении, модулируют энергию сигнала в широком диапазоне.

В методе расширения спектра с помощью скачкообразного перестраивания частот передатчик мгновенно изменяет одну частоту несущего сигнала на другую. Секретным ключом при этом является псевдослучайный закон изменения частот.

При компрессии с использованием ЛЧМ сигнал модулируется функцией, частота которой изменяется во времени.

Очевидно, что любой из указанных методов может быть распространен на использование в пространственной области при построении стеганографических систем.

Рассмотрим один из вариантов реализации метода РСПП, авторами которого являются Смит (J.R. Smith) и Комиски (B.O. Comiskey). Алгоритм модуляции следующий: каждый бит сообщения m_i, представляется некоторой базисной функцией φ_i, размерностью $X \times Y$, умноженной, в зависимости от значения бита (1 или 0), на +1 или -1:

$$E(x,y) = \sum_i m_i \cdot \varphi_i(x,y). \qquad (11.7)$$

Модулированное сообщение $E(x,y)$**,**полученное при этом, попиксельно суммируется с изображением-контейнером $C(x,y)$**,** в качестве которого используется полутоновое изображение размером $X \times Y$**.** Результатом является стеганоизображение $S(x,y) = C(x,y) + E(x,y)$, при $x \in 1...X$, $y \in 1...Y$.

Основное преимущество стеганографических методов, основанных на расширении спектра – сравнительно высокая стойкость к различного рода атакам на изображение, поскольку скрываемая информация распределена в широкой полосе частот и ее трудно удалить без полного разрушения контейнера.

11.5. Скрытие данных в аудиосигналах

Особое развитие получили цифровые методы стеганографии в аудиосреде. Скрытие данных в звуковых (аудио-) сигналах является особенно перспек-

тивным, поскольку слуховая система человека (ССЧ) работает в сверхшироком динамическом диапазоне. ССЧ воспринимает более чем миллиард к одному в диапазоне мощности и более чем тысяча к одному в частотном диапазоне [21]. Кроме этого, высокой является и чувствительность к аддитивному флуктуационному (белому) шуму. Отклонения в звуковом файле могут быть выявлены вплоть до одной десятимиллионной (на 70 дБ ниже уровня внешних шумов).

Несмотря на это, существуют определенные возможности для скрытия информации и в аудиосреде. Хотя ССЧ и имеет широкий динамический диапазон, она характеризуется достаточно малым разностным диапазоном. Как следствие, громкие звуки содействуют маскировке тихих звуков. Кроме того, ССЧ не способна различать абсолютную фазу, распознавая только относительную. Наконец, существуют некоторые виды искажений, вызванных окружающей средой, которые настолько обычны для слушателя, что в большинстве случаев им игнорируются.

Подобные особенности слухового аппарата человека позволяют удачно использовать аудиосреду с целью стеганографической защиты конфиденциальной информации.

11.5.1. Кодирование наименее значащих бит (временная область)

Кодирование младших разрядов является простейшим способом внедрить конфиденциальные данные в иные структуры данных. Используя звуковой сигнал, путем замены НЗБ каждой точки осуществления выборки, представленной двоичной последовательностью, можно зашифровать значительный объем информации.

Главный недостаток метода кодирования НЗБ, как и в случае с графическим контейнером, – это его слабая стойкость к посторонним воздействиям. Встроенная информация может быть разрушена из-за наличия шумов в канале, в результате передискретизации выборки и т.п., за исключением случаев, когда информация встраивалась с внесением избыточности. Однако последнее, обеспечивая приемлемую стойкость к помехам, приводит к уменьшению скорости передачи данных, зачастую на один/два порядка. На практике метод полезен только в замкнутых, полностью цифровых средах, не требующих дополнительного преобразования.

11.5.2. Метод фазового кодирования (частотная область)

Основная идея метода фазового кодирования состоит в замене фазы исходного звукового сегмента на опорную фазу, характер изменения которой отражает собой данные, которые необходимо скрыть. Для того чтобы сохранить разностную фазу между сегментами, фазы последних соответствующим образом согласовываются.

Фазовое кодирование, когда оно может быть использовано, является одним из наиболее эффективных методов по критерию отношения сигнал/воспринимаемый шум. Существенное изменение соотношения фаз между каждыми частотными составляющими приводит к значительному рассеиванию фазы. Тем не менее, до тех пор. пока модификация фазы в достаточной мере

мала, может быть достигнуто скрытие, неощутимое на слух. Разумеется, модификация считается малой по отношению к конкретному наблюдателю, поскольку специалисты по спектральному анализу способны обнаружить те изменения, которые непрофессионалу могут показаться незначительными.

Процедура фазового кодирования заключается в следующем:

-Звуковая последовательность $S[i],(1 \le i \le I)$ разбивается на серию N коротких сегментов (блоков) $S_n[i],(1 \le n \le N)$ – рисунок 11.5, а, б.

-К n-му сегменту сигнала $S_n[i]$ применяется K-точечное ДПФ, где $K = I / N$, и создаются массивы фаз $\phi_n(\omega_k)$ и амплитуд $A_n(\omega_k)$ для $1 \le k \le K$ (рисунок 11.5, в).

-Запоминается разность фаз между каждыми соседними сегментами для $1 \le n \le N$ (рисунок 11.5,г):

$$\Delta\phi_n(\omega_k) = \phi_n(\omega_k) - \phi_{n-1}(\omega_k); \qquad \Delta\phi_1(\omega_k) = 0. \tag{11.8}$$

-Двоичная последовательность данных представляется как $\phi_{data} = \pi / 2$ или

$\phi_{data} = -\pi / 2$, отображая, соответственно, “1” или “0” (рисунок 11.5, д):

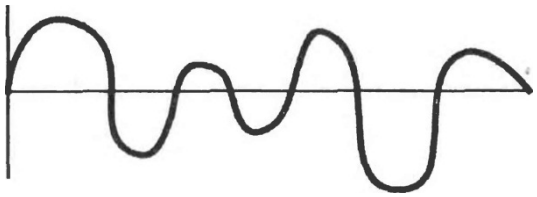
$\phi_1^{'}(\omega_k) = \phi_{data}$.

-С учетом разности фаз воссоздается новый массив фаз для $n > 1$ (рисунок 11.5, е).

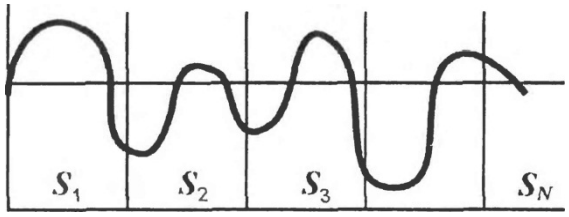
$$\left\| \begin{array}{l} \phi_1^{'}(\omega_k) = \phi_{data} \\ \phi_2^{'}(\omega_k) = \phi_1^{'}(\omega_k) + \Delta\phi_2(\omega_k) \\ \qquad\qquad ... \\ \phi_n^{'}(\omega_k) = \phi_{n-1}^{'}(\omega_k) + \Delta\phi_n(\omega_k) \\ \qquad\qquad ... \\ \phi_N^{'}(\omega_k) = \phi_{N-1}^{'}(\omega_k) + \Delta\phi_N(\omega_k) \end{array} \right\| \tag{11.9}$$

-Восстановление звукового сигнала осуществляется путем применения операции обратного ДПФ к исходной матрице амплитуд и модифицированной матрице фаз (рисунок 11.5, ж, з).

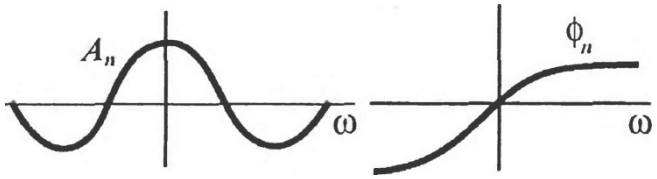
Перед процессом расшифровывания должна быть проведена синхронизация последовательности. Приемной стороне должны быть известны длина сегмента, точки ДПФ и интервал данных. Значение основной фазы первого сегмента определяется как "0" или "1", которые представляют закодированную двоичную последовательность.



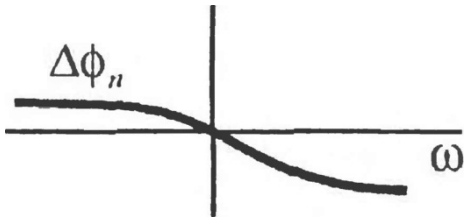
а) Исходный сигнал



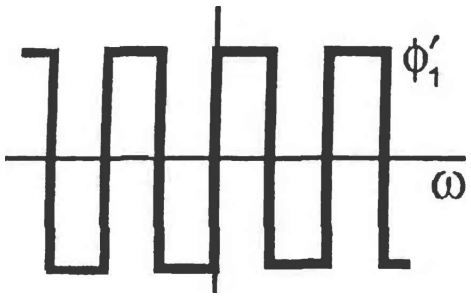
б) Разбитие S на N сегментов



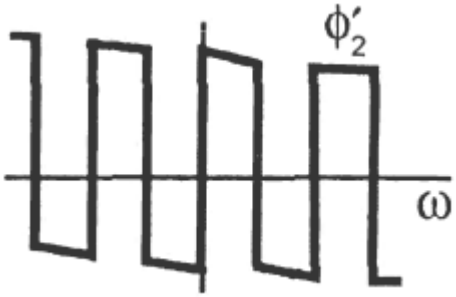
в) Выделение амплитуды и фазы каждого сегмента



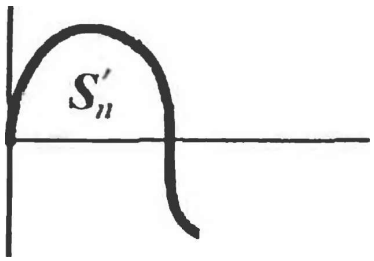
г) Вычисление разности фаз между соседними сегментами $\phi_n - \phi_{n-1}$



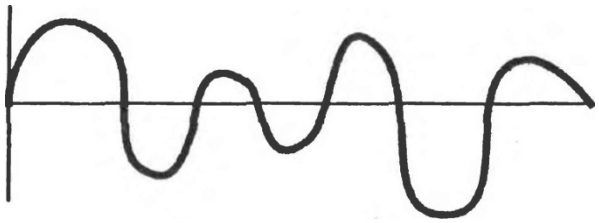
д) Для сегмента S_1 создается искусственная абсолютная фаза ϕ_1'



е) Для всех остальных сегментов создаются новые массивы фаз $(\phi_1^{'} + \Delta\phi_2)$



ж) Новая фаза и исходная амплитуда объединяются для получения нового сегмента $S_n^{'}$



з) Новые сегменты объединяют вместе для получения “заполненного” сигнала-контейнера

Рисунки 11.5 – Последовательность фазового кодирования

11.5.3. Метод расширения спектра (временная область)

В стандартном канале связи нередко бывает желательным сосредоточить информацию в как можно более узком диапазоне частотного спектра, например, для того чтобы сохранить имеющуюся полосу пропускания и уменьшить мощность сигнала. С другой стороны, основной метод расширения спектра предназначен для шифрования потока информации путем "рассеивания" кодированных данных по всему возможному частотному спектру. Последнее делает возможным прием сигнала даже при наличии помех на определенных частотах.

В [21] рассматривается технология расширения спектра сигнала прямой последовательностью (РСПП). Как уже указывалось выше (см. методы скрытия данных в изображении путем расширения спектра), методы РСПП расширяют

сигнал данных (сообщения), умножая его на элементарную посылку – ПСП максимальной длины, модулированную известной частотой.

Поскольку аудиосигналы, используемые в качестве контейнеров, имеют дискретный формат, то для кодирования в качестве частоты элементарной посылки можно использовать частоту дискретизации. Как следствие, дискретный характер сигнала устраняет наиболее сложную проблему, которая возникает при получении сигнала с расширенным прямой последовательностью спектром, – корректное определение начала и конца составляющих элементарной посылки с целью фазовой синхронизации. Следовательно, возникает возможность использования намного более высокой частоты следования элементарных посылок, и, таким образом, получения значительной связанной с ней скорости передачи данных.

В РСПП для шифрования и дешифрования информации необходим один и тот же ключ – псевдослучайный шум, который в идеальном случае имеет плоскую частотную характеристику во всем диапазоне частот (так называемый белый шум). Ключ применяется к скрываемой информации и трансформирует ее последовательность в последовательность с расширенным спектром.

Метод РСПП по отношению к аудиосигналам заключается в следующем. Сигнал данных умножается на сигнал несущей и псевдослучайную шумовую последовательность, характеризующуюся широким частотным спектром. В результате этого спектр данных расширяется на всю доступную полосу. В дальнейшем последовательность расширенных данных ослабляется и прибавляется к исходному сигналу как аддитивный случайный шум.

11.5.4. Скрытие данных с использованием эхо-сигнала

Данный метод подразумевает под собой встраивание данных в аудиосигнал – контейнер путем введения в него эхо–сигнала [21]. Данные скрываются изменением трех параметров эхо–сигнала: начальной амплитуды, скорости затухания [(начальная амплитуда – затухание)/ δ] и сдвига (рисунок 11.6).

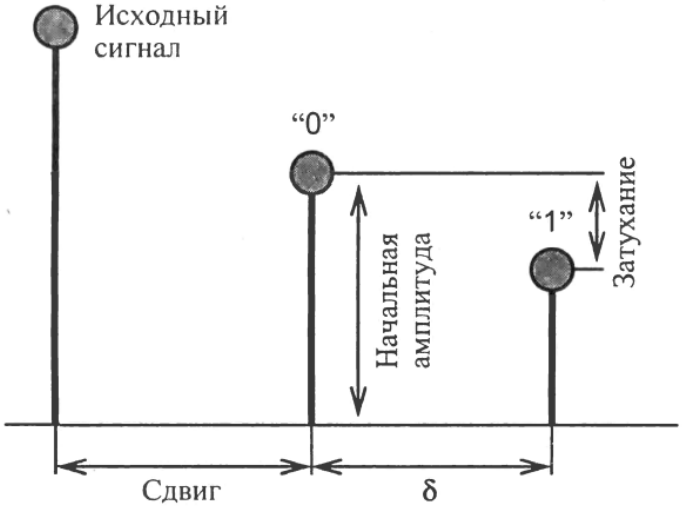


Рисунок 11.6 – Регулируемые параметры эхо–сигнала

Когда сдвиг (задержка) между первичным и эхо–сигналом уменьшается, начиная с некоторого значения задержки, ССЧ становится не способной обнаружить разницу между двумя сигналами, а эхо–сигнал воспринимается только как дополнительный резонанс. Упомянутое значение трудно определить точно, поскольку оно зависит от качества первичной звукозаписи, типа звука, для которого формируется эхо–сигнал, и$_г$ в конечном итоге, – от слушателя.

В общем случае согласно [21] для большинства звуков и большинства слушателей смешивание происходит при задержке, соответствующей приблизительно одной миллисекунде.

Стеганокодер использует два времени задержки: одно для представления двоичного нуля ("сдвиг" на рисунке 11.6), а другое – для представления двоичной единицы ("сдвиг + δ"). Оба времени задержки меньше того предельного времени, за которое ССЧ способна распознать эхо–сигнал. Кроме уменьшения времени задержки для обеспечения неощущаемости также можно установить уровни начальной амплитуды и времени затухания, которые бы не превышали порог чувствительности ССЧ.

Для того чтобы в первичный сигнал закодировать более одного бита, сигнал раскладывается на меньшие сегменты. Каждый сегмент при этом рассматривается как отдельный сигнал и в него может быть встроен (путем эхо–отображения) один бит информации. Результирующий закодированный сигнал (содержащий несколько бит) представляет собой новое объединение всех независимо закодированных сегментов исходного сигнала.

На рисунке 11.7 изображен пример, при котором сигнал был разделен на 7 равных сегментов, помеченных как a,b,c,d,e,f и g.

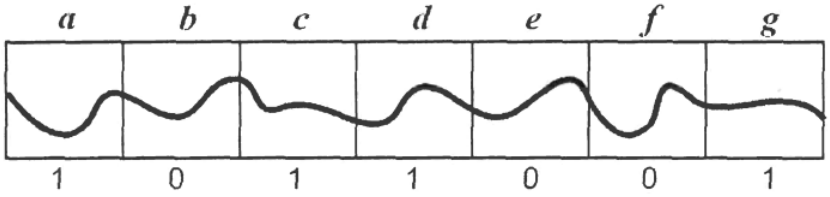


Рисунок 11.7 – Разбиение первичного сигнала на меньшие сегменты для встраивания информации, представляющей собой последовательность двоичных данных

Пусть необходимо, чтобы сегменты a,c,d и g содержали "1". Следовательно, для каждого из них нужно применить системную функцию представления единицы (рисунок 11.6). Каждый сегмент индивидуально сворачивается с системной функцией. Нули, помещенные в сегменты b,e и f, кодируются аналогично, используя способ представления нуля (рисунок 11.6).

Полученные после сворачивания с соответствующей функцией результаты повторно объединяются.

Для достижения минимальной заметности повторного объединения, в [21] предварительно предлагается создать отдельные "единичный" и "нулевой"

эхо– сигналы, повторяя первичный и используя соответствующие представления "1" и "0". Полученные в результате сигналы изображены на рисунок 11.8.

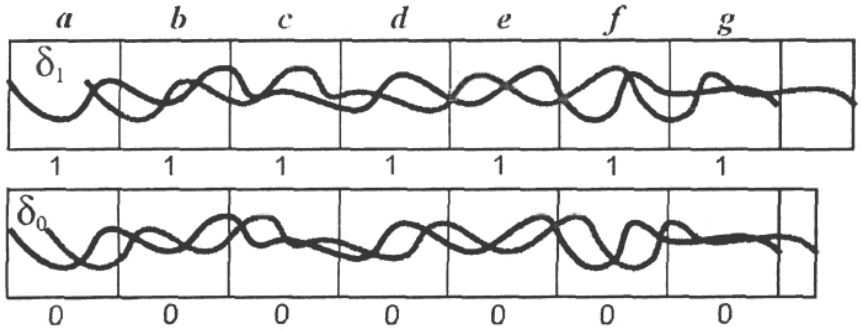


Рисунок 11.8 – Создание "единичного" и "нулевого" эхо–сигналов (более светлая линия)

"Единичный" и "нулевой" эхо–сигналы содержат, соответственно, только единицы и нули. Для того чтобы объединить эти два сигнала, также создаются два смешивающих сигнала (рисунок 11.9), которые представляют собой последовательность двоичных данных, состояние которой зависит оттого, какой бит необходимо скрыть в том или ином сегменте первичного сигнала.

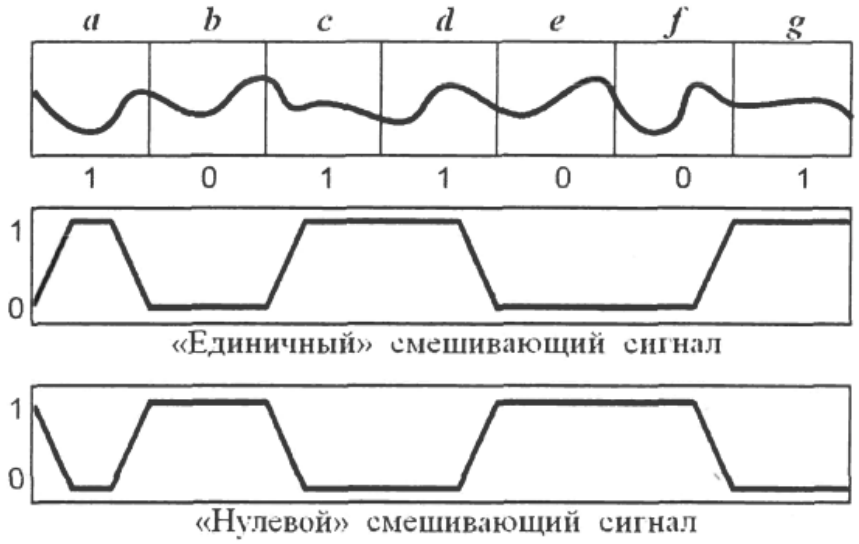


Рисунок 11.9 – Смешивающие сигналы

"Единичный" и "нулевой' смешивающие сигналы умножаются на соответствующие им эхо–сигналы. Иными словами, последние масштабируются единицей или нулем на протяжении всего времени действия сигнала в зависимости от того, какой бит предусматривается поместить в любой из его отдельных сегментов. В дальнейшем два результата складываются друг с другом.

Необходимо заметить, что "нулевой" смешивающий сигнал представляет собой инверсию "единичного". Кроме этого, фронты переходов каждого из сиг-

налов являются наклонными. Сумма обоих смешивающих сигналов всегда равняется единице. Все это позволяет получить плавный переход между сегментами, кодированными разными битами, а также предотвращает возникновение резких изменений в звучании результирующего (смешанного) сигнала.

Структурная схема, которая отображает полный процесс встраивания, показана на рисунке 11.10

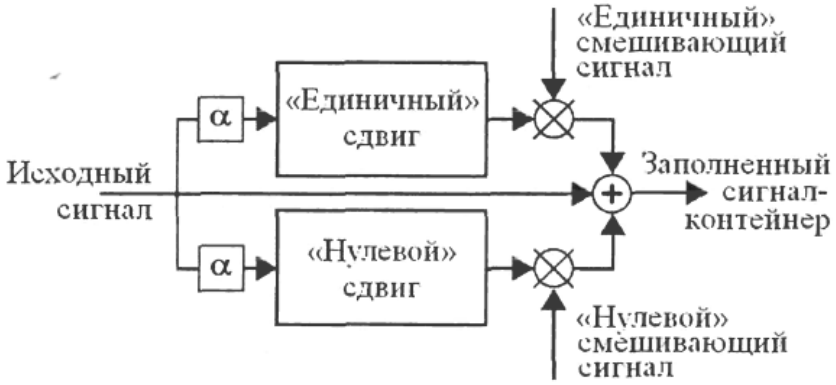


Рисунок 11.10 – Структурная схема встраивания информации методом эхо–сигнала

Извлечение вложенной информации подразумевает под собой выявление интервала между эхо–сигналами отдельных сегментов.

11.6. Скрытие данных в тексте

Для скрытия конфиденциальных сообщений в тексте (так называемая лингвистическая стеганография) используется или обычная избыточность письменной речи, или же форматы представления текста.

Наиболее сложным объектом для скрытия данных по многим причинам является электронная (файловая) версия текста. В отличие от текстового файла его "жесткая" копия (например, бумажная) может быть обработана как высокоструктурированное изображение и поэтому является относительно легко поддающейся разнообразным методам скрытия, таким как незначительные изменения формата текстовых шаблонов, регулирование расстояния между определенными парами символов (кернинг), расстояния между строками и т.п. В значительной степени такая ситуация вызвана относительным дефицитом в текстовом файле избыточной информации, особенно в сравнении с графическими или, например, звуковыми файлами. В то время как в большинстве случаев существует возможность внести незаметные глазу и неощутимые на слух модификации в изображение и звук, даже дополнительная буква или знак пунктуации в тексте могут быть легко распознаны случайным читателем.

Скрытие данных в тексте требует поиска таких модификаций, которые были бы незаметными подавляющему большинству читателей. Авторы [21]

рассматривают три группы методов, которые получили наибольшее распространение при встраивании скрываемых данных в текст:

–*методы произвольного интервала*, которые осуществляют встраивание путем манипуляции с пробельными символами (свободным местом на печатной полосе);

–синтаксические методы, которые работают с пунктуацией;

–семантические методы, в основу алгоритмов которых положено манипулирование словами, зависимое от скрываемых бит данных.

11.6.1. Методы произвольного интервала

Существует, по меньшей мере, две причины, по которым манипулирование свободным местом в определенных случаях показывает довольно неплохие результаты. Во–первых, изменение количества пробелов в конце текстовой строки не вызывает существенных изменений в значении фразы или предложения. Во–вторых, среднестатистический читатель вряд ли заметит незначительные модификации свободного места страницы текста.

В [21] предложено три метода, которые для скрытия данных используют свободное место в тексте. Указанные методы оперируют с интервалами между предложениями, пропусками в конце текстовых строк и интервалами между словами в тексте, выровненном по ширине.

11.6.1.1. Метод изменения интервала между предложениями. Метод изменения интервала между предложениями позволяет встраивать в текст сообщение, имеющее двоичный формат, путем размещения одного или двух пробелов после каждого символа завершения предложения. В качестве символов окончания предложения могут служить, к примеру, точки в обычном тексте, точки с запятой для кода программ на языке C++ и т.п. При этом единичным пробелом может кодироваться бит "1", двойным – бит "0".

Кроме несомненной простоты, данный метод имеет и ряд недостатков. Во–первых, он не эффективен, поскольку для встраивания незначительного количества бит требуется текст значительного объема. В частности, один бит, который возможно скрыть в одном предложении, эквивалентен скорости передачи данных, соответствующей приблизительно одному биту на 160 байт текстового контейнера, при условии, что в среднем предложение представляет собой две строки по 80 символов каждая.

Во–вторых, возможность скрытия весьма зависит от структуры текстового контейнера (некоторые тексты, как например, верлибры или свободные стихи характеризуются отсутствием стабильных согласованных или однозначных знаков завершения строки).

В-третьих, существуют текстовые редакторы, которые автоматически устанавливают после точки в конце предложения один–два пробела (так называемое автозавершение). И, наконец, как отмечается в [21], непоследовательное и противоречивое использование свободных мест может оказаться достаточно заметным для читателя.

11.6.1.2. Метод изменения количества пробелов в конце текстовых строк. Еще один метод использования свободных мест полосы текста для встраивания конфиденциальных данных заключается в добавлении пробелов в конец каждой текстовой строки. Количество добавляемых пробелов зависит от значения встраиваемого бита. Два пробела кодируют один бит на строку, четыре пробела – два бита, восемь – три и т.д. Такой подход позволяет существенно увеличить, по сравнению с предыдущим методом, количество информации, которую можно скрыть в тексте аналогичного объема.

Дополнительные преимущества указанного метода состоят в том, что он может быть применен к любому тексту. Изменения в формате последнего будут в достаточной степени незаметными, поскольку используемые при этом свободные места являются периферийными по отношению к основному тексту.

Недостатком данного (как, в конечном счете, и предыдущего) метода является то, что некоторые программы обработки текста могут непреднамеренно удалять дополнительно внесенные пробелы. Кроме того, характерным недостатком рассматриваемого метода является очевидная невозможность извлечения скрытых данных из бумажной копии текста (из–за невидимости пробелов).

11.6.1.3. Метод изменения количества пробелов между словами выровненного по ширине текста. Данный метод позволяет скрывать данные в свободных местах текста, выровненного по ширине. При этом биты данных встраиваются путем управляемого выбора позиций, в которых будут размещены дополнительные пробелы. Один пробел между словами интерпретируется как "0". Два пробела – как "1". В среднем метод позволяет встраивать по несколько бит в одну строку.

Рассмотренные методы произвольного интервала эффективны, при условии, что текст представлен в формате ASCII. Как было уже отмечено выше, некоторые данные могут оказаться утраченными после распечатывания текста. Печатные документы выдвигают к скрытию данных такие требования, которые далеко выходят за возможности текстового файла при кодировании ASCII. При этом скрытие данных в «жестких» копиях текста может выполняться путем незначительных изменений расстояния между словами и отдельными буквами, изменением позиций базовых линий (линий, на которых лежат наиболее низкие элементы букв или знаков пунктуации строки), изменением форм символов и т.п.

11.6.2. Синтаксические и семантические методы

Тот факт, что свободное место для встраивания выбирается произвольно, является одновременно как преимуществом, так и недостатком с точки зрения скрытости данных. Обычный читатель может и не заметить манипуляции с текстом, тогда как текстовый редактор способен автоматически изменить количество и размещение пробелов, таким образом разрушая скрытые данные.

Низкая стойкость к атакам, в свете возможного переформатирования документа, выступает одной из причин поиска других методов встраивания данных в текстовые контейнеры. Кроме этого, синтаксические и семантические методы вообще никоим образом не используют свободные места в тексте, кар-

динально отличаясь от рассмотренных выше алгоритмов. Однако, все они могут использоваться одновременно, дублируя или же дополняя друг друга.

К *синтаксическим методам* текстовой стеганографии относятся методы изменения пунктуации и методы изменения структуры и стиля текста [21]. Существует немало случаев, когда правила пунктуации являются неоднозначными и несоблюдение их не влияет существенно на общее содержание текста. Так, например, фразы "красный, зеленый, синий" и "красный, зеленый и синий" эквивалентны друг другу. Тот факт, что выбор подобных форм может быть произвольным (разумеется, с позиций используемого в качестве контейнера текста, поскольку очевидно, что стеганосистема, построенная на основе видоизменения текста, известного широкому кругу лиц (например, классики), вряд ли может считаться надежной), и используется при построении стеганосистем на основе синтаксических методов. Периодическое изменение форм при этом может быть поставлено в соответствие с двоичными данными. Например, появление в тексте формы перечисления с союзом "и" может подразумевать под собой встроенный бит "1", в то время как отсутствие союза при перечислении будет говорить о том, что в данном случае встроен бит "0". Другим примером может служить использование сокращений и аббревиатур. Средняя скорость передачи данных такими методами составляет несколько бит на один килобайт текста [21].

Однако, в то время как письменный язык предоставляет достаточно возможностей для синтаксического скрытия данных, эти возможности исчезают в известных классических произведениях. Кроме того, хотя некоторые из правил пунктуации и считаются неоднозначными, их противоречивое использование может стать объектом внимания для цензора. Также возможны случаи, когда изменение пунктуации приводит к снижению воспринимаемости текста или же к приобретению текстом диаметрально противоположного смысла. Поэтому синтаксические методы рекомендуется применять с осмотрительностью [21].

К синтаксическим методам также относятся методы изменения стиля и структуры текста без значительного изменения его смысловой нагрузки. Например, предложение "*Существует немало случаев, когда правила пунктуации являются неоднозначными*" можно сформулировать как "*Правила пунктуации являются неоднозначными во многих случаях*". Такие методы являются более незаметными для посторонних, по сравнению с методами изменения пунктуации, однако возможность их использования ограничена сложностью автоматизирования процесса стеганографического встраивания и извлечения бит сообщения.

Семантические методы подобны синтаксическим. Наряду с этим, вместо того чтобы встраивать двоичные данные, используя двусмысленность грамматической формы, семантические методы определяют два синонима, которые отвечают значениям скрываемых бит. К примеру, слово "но" может быть поставлено в соответствие к "0", а слово "*однако*" – к "1".

Для проведения скрытия с использованием семантических методов необходимо наличие таблицы синонимов. Кроме того, как отмечается в [21], если слову отвечает достаточно большое количество синонимов, возникает возмож-

ность одновременного кодирования большего количества бит. Скажем, выбор между синонимами "*секретный", "тайный", "скрытый", "конфиденциальный", "негласный", "неизвестный", "засекреченный", "закрытый*" дает возможность представить три бита данных за одно встраивание. Проблемы могут возникнуть, однако, когда желанию встроить бит информации препятствует нюанс значения слова.

11.7. Скрытие данных с использованием хаотических сигналов

Наиболее планомерные исследования по реализации и технической оптимизации систем передачи информации с применением хаотических сигналов в качестве носителя информации были проведены на основе схем хаотической синхронизации для скрытой передачи сообщений. Принцип работы таких систем заключается в следующем: сигнал с канала связи поступает на генератор (генераторы) хаотического сигнала и синхронизирует его при приеме бита “0” и не синхронизирует при приеме бита “1”, т.е. при приеме бита “0” вырабатывается из информационного, и на выходе будет получен восстановленный сигнал $m(t)$, представляющий собой последовательность участков с синхронным (бит “0”) и несинхронным (бит “1”) поведением (рисунок 11.11)

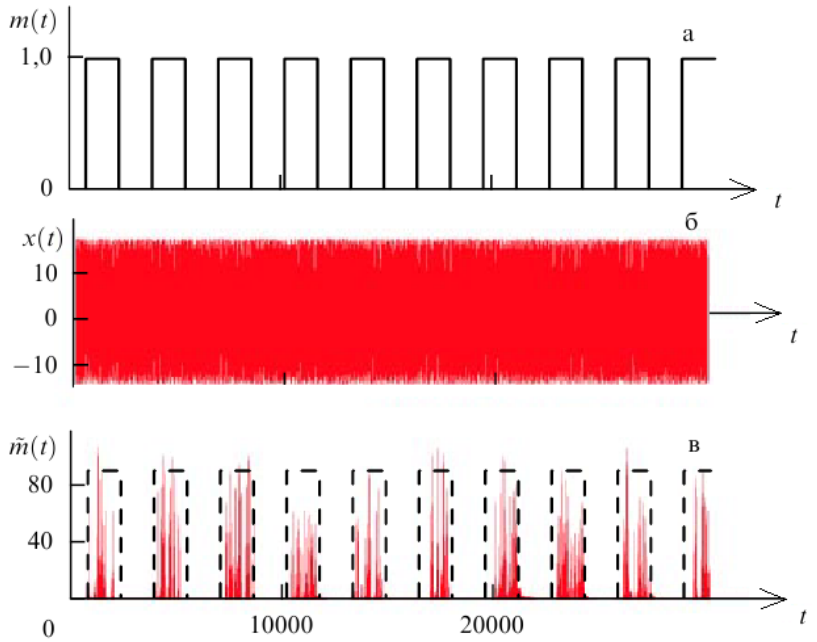


Рисунок 11.11 – Иллюстрация реализации способа скрытой передачи информации на основе хаотической синхронизации: а – информационный сигнал $m(t)$, представленный простой последовательностью бинарных битов 0/1, б – сигнал $x(t)$, производимый передающим генератором для последующей передачи по каналу связи, в – восстановленный сигнал $\tilde{m}(t)$ и детектированный информационный сигнал (штриховая линия).

Принципиальным достоинством методов на основе хаотической синхронизации по сравнению с традиционными методами (методом LSB (Least Significant Bit), эхо–методами, методами расширенного спектра и др.) являются значительное повышение устойчивости к шумам и искажениям в канале связи, а также увеличение скорости передачи информации. Кроме того, использование именно хаотической синхронизации чрезвычайно важно для повышения конфиденциальности передачи данных.

Основными типами хаотической синхронизации, лежащими в основе современных систем связи, являются режимы полной, фазовой и обобщённой синхронизации. Для создания целостной картины кратко остановимся на описании этих типов синхронного поведения.

Режим *полной синхронизации* означает точное совпадение векторов состояния взаимодействующих (однонаправленно или взаимно связанных) систем $x(t) \equiv u(t)$, и, следовательно, этот режим возможен лишь в случае их идентичности по управляющим параметрам. Если управляющие параметры слегка различаются, возможно возникновение режима *синхронизации с запаздыванием*, в котором взаимодействующие системы демонстрируют близкие к идентичным, но сдвинутые на некоторый временной интервал τ колебания, т.е. $x(t) \approx u(t+\tau)$.

Обобщённая синхронизация, которая вводится в рассмотрение для системы двух однонаправленно связанных хаотических осцилляторов – ведущего $x(t)$ и ведомого $u(t)$, означает, что после завершения переходного процесса устанавливается функциональная зависимость между их состояниями, т.е. $u(t) = F[x(t)]$. При этом вид зависимости $F[\bullet]$ может быть достаточно сложным, а процедура её нахождения весьма нетривиальной.

Фазовая синхронизация означает, что происходит захват фаз хаотических сигналов, в то время как амплитуды этих сигналов остаются несвязанными между собой и выглядят хаотическими.

11.7.1. Способы скрытой передачи информации, основанные на явлении полной хаотической синхронизации

Использование полной хаотической синхронизации для скрытой передачи информации подразумевает наличие, как минимум, двух однонаправленно связанных идентичных хаотических генераторов. Предложено достаточно много таких способов скрытой передачи данных. Это, в первую очередь, хаотическая маскировка, переключение хаотических режимов, нелинейное подмешивание информационного сигнала к хаотическому, модулирование управляющих параметров передающего генератора полезным цифровым сигналом и др. На основе этих методов было предложено множество способов скрытой передачи данных. Поэтому рассмотрение основных принципов работы таких схем является очень важным. Остановимся на них более подробно.

11.7.1.1. Хаотическая маскировка. Хаотическая маскировка – один из первых и наиболее простых способов скрытой передачи данных [26]. Принципиальная схема реализации этого способа приведена на рисунке 11.12. На передающей стороне информационный сигнал $m(t)$ подмешивается в сумматоре к несущему сигналу, генерируемому передающей хаотической системой $x(t)$, и далее передаётся по каналу связи. В приёмнике осуществляется полная хаотическая синхронизация находящегося в нём хаотического генератора $u(t)$ с помощью принимаемого сигнала, в результате чего динамика принимающего генератора становится идентичной динамике передающего. Детектированный сигнал $\tilde{m}(t)$ получается после прохождения через вычитающее устройство как разность между принимаемым сигналом и синхронным откликом генератора хаоса в приёмнике.

Такая схема скрытой передачи данных работает достаточно эффективно (т.е. позволяет качественно передавать информацию и детектировать её на выходе) в отсутствие шумов в канале связи в том случае, когда мощность сигнала, генерируемого передающей системой, превышает мощность информационного сигнала на 35–65 дБ. Добавление шума в канал связи приводит к резкому ухудшению качества передаваемой информации, а следовательно, к высоким отношениям сигнал/шум, при которых схема остаётся работоспособной. Кроме того, введение расстройки управляющих параметров между идентичными хаотическими генераторами (находящимися на различных сторонах канала связи) также приводит к появлению на выходе дополнительных шумов десинхронизации и делает передачу информации труднореализуемой. Более того, существует проблема конфиденциальности передачи информации (Здесь и далее под конфиденциальностью мы понимаем отсутствие возможности детектирования третьей стороной информационного сообщения по сигналу, передаваемому по каналу связи). Несмотря на низкий уровень информационного сигнала по сравнению с уровнем несущего, существуют методы и подходы, позволяющие восстановить исходный хаотический сигнал по сигналу, передаваемому по каналу связи, а следовательно, выделить полезную информацию.

Все вышеуказанные недостатки делают схемы скрытой передачи информации на основе хаотической маскировки малоприменимыми на практике.

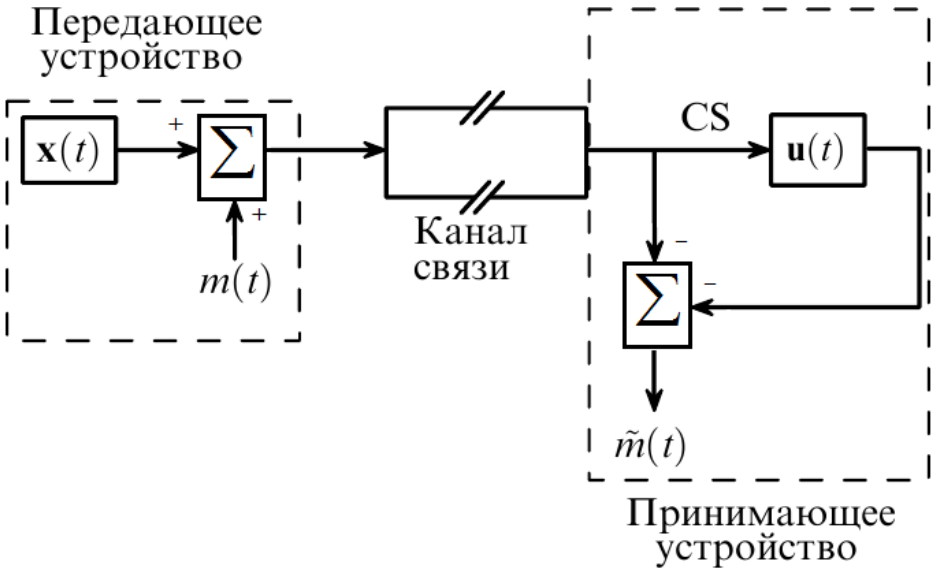


Рисунок 11.13 – Схема скрытой передачи информации с помощью хаотической маскировки (CS – полная хаотическая синхронизация).

11.7.1.2. Переключение хаотических режимов. Одна из схем переключения хаотических режимов приведена на рисунке 11.13. Передающее устройство содержит два хаотических генератора, $x_1(t)$ и $x_2(t)$, которые могут быть разными или одинаковыми, но с различающимися параметрами, однако в интересах конфиденциальности передачи данных предпочтительнее использовать последние; более того, сигналы, генерируемые этими системами должны иметь сходные спектральные и статистические свойства. Полезный цифровой сигнал $m(t)$, представленный последовательностью бинарных битов 0/1, используется для переключения передаваемого сигнала, т.е. сигнал, производимый первым хаотическим генератором, кодирует, например, бинарный бит 0, а сигнал от второго генератора хаоса соответственно – бинарный бит 1. Полученный таким образом сигнал передаётся по каналу связи на принимающее устройство. В зависимости от числа генераторов, находящихся на принимающей стороне канала связи, различают несколько схем скрытой передачи данных на основе переключения хаотических режимов. В схеме, представленной на рисунке 11.4, принимающее устройство содержит один хаотический генератор $x(t)$, идентичный любому из передающих, например первому. Параметры генераторов должны быть выбраны таким образом, чтобы генерируемые ими сигналы приводили к возникновению режима полной хаотической синхронизации лишь в том случае, если передаётся только бинарный бит 0 (или только бинарный бит 1). Так же как и при хаотической маскировке, восстановленный $\tilde{m}(t)$ получается после прохождения через вычитающее устройство сигнала, передаваемого по каналу связи, и синхронного отклика хаотического генератора принимающего устройства.

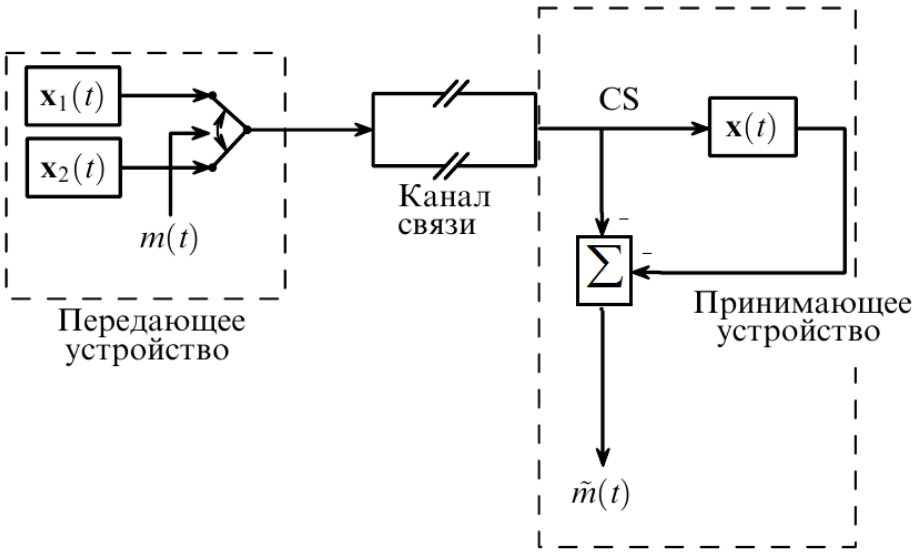


Рисунок 11.14 – Схема скрытой передачи информации на основе переключения хаотических режимов.

Другие схемы скрытой передачи информации с использованием переключения хаотических режимов, которые основаны на той же идее, отличаются от описанной выше схемы только строением и работой принимающего устройства. Например, известны схемы, принимающее устройство содержит два хаотических генератора, идентичных передающим генераторам, и, следовательно, два вычитающих устройства для детектирования полезного сигнала. В этом случае полезный сигнал диагностируется по наличию или отсутствию хаотических колебаний в сигналах на выходе принимающего устройства.

Такие схемы передачи данных оказываются более устойчивыми к шумам в канале связи, чем схемы с хаотической маскировкой, но их устойчивость к шумам, тем не менее, остаётся весьма ограниченной. Принципиальным недостатком таких схем является возникновение переходных процессов при переключении (длительность которых может быть весьма продолжительной), что проявляется во временной задержке включения в синхронный режим принимающего генератора. Поэтому такие схемы являются достаточно медленными. Кроме того, степень секретности (конфиденциальности) таких схем является довольно низкой.

11.7.1.3 Нелинейное подмешивание информационного сигнала к хаотическому. Усовершенствования метода хаотической маскировки были направлены на повышение секретности и конфиденциальности передачи информации. В результате было предложено несколько способов, которые можно объединить общим названием "нелинейное подмешивание информационного сигнала к хаотическому". Особенностью работы таких схем является непосредственный ввод информационного сигнала в передающую систему и его участие в формировании выходного сигнала.

Среди схем, в которых применяются различные операции ("сложение–вычитание", "деление–умножение", "сложение по модулю с основанием 2",

"преобразование напряжение – ток" и др.), наибольшее распространение сейчас получили схемы, использующие "сложение–вычитание" [125, 146]. В таких схемах информационный сигнал подмешивается к хаотическому и участвует тем самым в формировании сложного поведения системы. Наиболее простым и технически реализуемым способом обеспечения "нелинейного подмешивания" является установка на передающей стороне канала связи дополнительного хаотического генератора, идентичного первому передающему и взаимно связанного с ним. Схема реализации такого способа скрытой передачи данных приведена на рисунке 11.15.

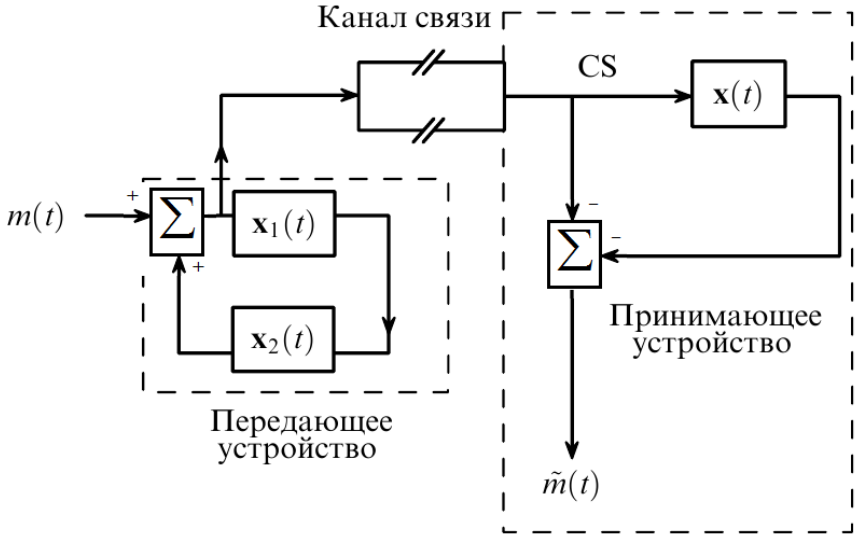


Рисунок 11.15 – Схема скрытой передачи информации посредством нелинейного подмешивания информационного сигнала к хаотическому.

Итак, передающая сторона содержит два идентичных по управляющим параметрам хаотических генератора, $x_1(t)$ и $x_2(t)$. Информационный сигнал $m(t)$ подмешивается к сигналу, производимому одним из генераторов передающего устройства (или к обоим сигналам одновременно). В результате прохождения по кольцу обратной связи (обеспечиваемого взаимной связью генераторов передающего устройства) сигнал претерпевает нелинейные изменения. Таким образом, по каналу связи будет передаваться сигнал, полученный в результате нелинейного подмешивания информационного сигнала к хаотическому.

Принимающее устройство, как и в рассмотренных выше схемах, содержит хаотический генератор $x(t)$, идентичный по управляющим параметрам передающим генераторам. Сигнал, поступающий по каналу связи на принимающее устройство, синхронизует принимающий генератор в случае передачи бинарного бита 0 (и не синхронизует при передаче бинарного бита 1). После прохождения через вычитающее устройство сигналов от передающего и принимающего генераторов детектируется восстановленный сигнал $\tilde{m}(t)$.

Важным преимуществом таких схем перед схемами, основанными на хаотической маскировке, является возможность варьирования уровня вводимого информационного сообщения, что позволяет управлять качеством передачи

информации (т.е. варьировать точность дешифрации исходного информационного сообщения принимающей стороной). Однако увеличение качества передачи информации влечёт за собой потерю её конфиденциальности, что является существенным недостатком. Кроме того, такие схемы характеризуются достаточно низкой устойчивостью к шумам в канале связи и расстройке управляющих параметров изначально идентичных хаотических генераторов. Необходимость обеспечения идентичности трёх генераторов хаоса, два из которых находятся на разных сторонах канала связи, представляет собой труднорешаемую техническую задачу, а следовательно, является ещё одним недостатком такой схемы.

Кроме того, зависимость передаваемого сигнала от информационного, поскольку передающий генератор по сути является неавтономной системой, что не гарантирует формирования им именно хаотического сигнала при изменении тех или иных параметров схемы, может приводить к потере конфиденциальности.

11.7.1.4 Модулирование управляющих параметров передающего генератора информационным сигналом. Схемы на основе модулирования управляющих параметров, или адаптивные методы, – естественный этап при переходе от дискретной модуляции управляющего параметра передающего генератора в схеме с переключением хаотических режимов (см. раздел 11.7.1.2) к модуляции непрерывным сигналом. При этом роль модулирующего сигнала играет информационный сигнал. Необходимым условием реализации таких схем является предварительное определение допустимого диапазона изменения параметра и нормирование модулирующего информационного сигнала. Частным случаем является использование бинарного цифрового сигнала в качестве информационного и модулирование им управляющего параметра передающего генератора. Схема скрытой передачи информации таким способом приведена на рисунке 11.6.

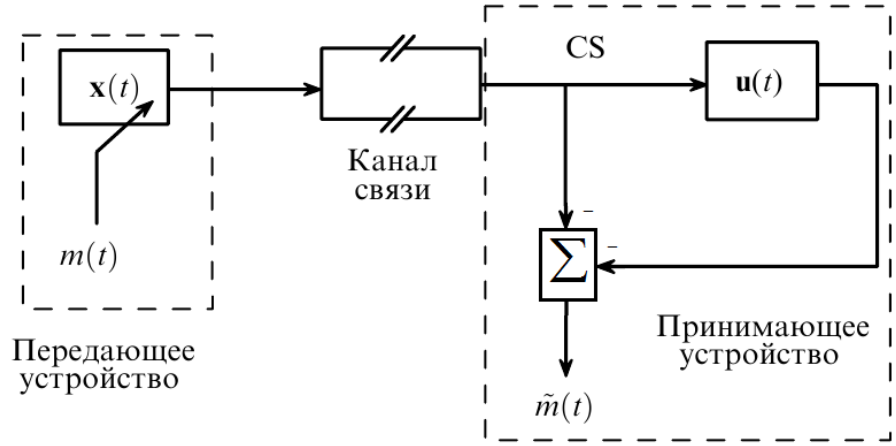


Рисунок 11.16 – Схема скрытой передачи информации путём модулирования управляющего параметра передающего генератора информационным сигналом.

Принцип её работы аналогичен принципу работы схемы на основе переключения хаотических режимов, описанной в разделе 11.7.1.2. Полезный циф-

ровой сигнал $m(t)$ модулирует один из параметров передающего генератора $x(t)$ таким образом, чтобы в зависимости от передаваемого бинарного бита 0 (1) между передающим $x(t)$ и принимающим $u(t)$ генераторами существовал (отсутствовал) режим полной хаотической синхронизации. Тогда после прохождения через вычитающее устройство сигналов передающего и принимающего устройств детектируется восстановленный сигнал $m(t)$. Для возможности реализации режима полной синхронизации управляющие параметры принимающего генератора должны быть выбраны идентичными управляющим параметрам передающего (точнее, одному из наборов параметров передающего генератора, отвечающему, например, бинарному биту 0).

Особенности работы, достоинства и недостатки схем, основанных на модулировании управляющих параметров, являются теми же, что и в случае схем с переключениями. Однако для рассматриваемой схемы техническая реализация несколько упрощается благодаря наличию на передающей стороне канала связи только одного генератора.

11.7.2. Способ скрытой передачи информации на основе обобщённой синхронизации

Одной из немногих работ, в которых используется режим обобщённой синхронизации для скрытой передачи информации, является работа [109]. Принципиальная схема реализации такого способа скрытой передачи данных приведена на рисунке 11.17. Передающая сторона содержит два хаотических генератора, ведущий $x(t)$ и ведомый $u(t)$, которые могут быть неидентичными. Сигнал с ведущего генератора передаётся на ведомый, причём его интенсивность модулируется полезным цифровым сигналом $m(t)$ таким образом: если передаётся бинарный бит 0, то между ведущим и ведомым генераторами устанавливается режим обобщённой синхронизации, а если передаётся бинарный бит 1, то режим обобщённой синхронизации между ними разрушается. На принимающей стороне канала связи находится так называемый вспомогательный хаотический генератор $v(t)$, идентичный ведомому по управляющим параметрам. Сигнал с ведущего генератора по каналу связи передаётся на вспомогательный, что обеспечивает возникновение режима обобщённой синхронизации между ними, причём интенсивность передаваемого по каналу связи сигнала должна совпадать с интенсивностью сигнала, поступающего к ведомой системе при передаче бинарного бита 0. Сигнал с ведомого генератора уже по другому каналу связи передаётся принимающей стороне. Так же как и в способах скрытой передачи данных, основанных на режиме полной хаотической синхронизации, принимающая сторона имеет в своём распоряжении как хаотический сигнал, содержащий полезную информацию, так и сигнал без неё. Поэтому можно легко выделить полезный цифровой сигнал $\tilde{m}(t)$ простым вычитанием одного сигнала из другого.

Нетрудно видеть, что в такой схеме скрытой передачи информации активно используется метод вспомогательной системы, что требует наличия двух идентичных по управляющим параметрам хаотических генераторов. Так же

как и в схемах, основанных на режиме полной хаотической синхронизации, эти генераторы располагаются на разных сторонах канала связи, что представляет собой существенную проблему с точки зрения технической реализации данного метода. Небольшая расстройка значений управляющих параметров в этих системах приводит к появлению шумов десинхронизации, делая такую схему неработоспособной (под шумом десинхронизации понимается сигнал $\Delta x = x_2\ x_1$, где $x_{1,2}(t)$ – сигналы, поступающие на вычитающее устройство, в данном случае сигналы с ведомого и вспомогательного генераторов хаоса плюс шумы канала связи. При наличии синхронного режима $\Delta x = 0$).Кроме того, реализация двух каналов связи является существенным недостатком не только из–за дополнительных затрат при реализации, но и вследствие того, что наличие двух каналов способствует появлению дополнительных шумов в канале связи (возможно, даже совершенно другой природы), искажающих передаваемый сигнал. Поэтому такая схема скрытой передачи данных характеризуется достаточно низкой устойчивостью к шумам в канале связи и является труднореализуемой на практике.

Возникают также проблемы с конфиденциальностью передачи информации. Понятно, что использование другого типа синхронного поведения, а также наличие дополнительного канала связи, с этой точки зрения, играют положительную роль. Однако, так же как и в схемах на основе нелинейного подмешивания информационного сигнала к хаотическому (см. раздел 11.7.1.3), повышение качества передаваемой информации влечёт за собой потерю конфиденциальности. Но эта проблема здесь является менее существенной по сравнению с аналогичной проблемой для схем, основанных на режиме полной хаотической синхронизации (см. раздел 11.7.1). Повысить конфиденциальность передачи информации можно с помощью использования нескольких типов синхронного поведения одновременно. Например, предложены способы скрытой передачи данных, использующие одновременно режимы обобщённой и полной хаотической синхронизации.

11.7.3. Способ скрытой передачи информации на основе фазовой хаотической синхронизации

Принципиальная схема реализации такого способа приведена на рисунке 11.18.

На передающей стороне канала связи находятся два идентичных взаимосвязанных хаотических генератора.

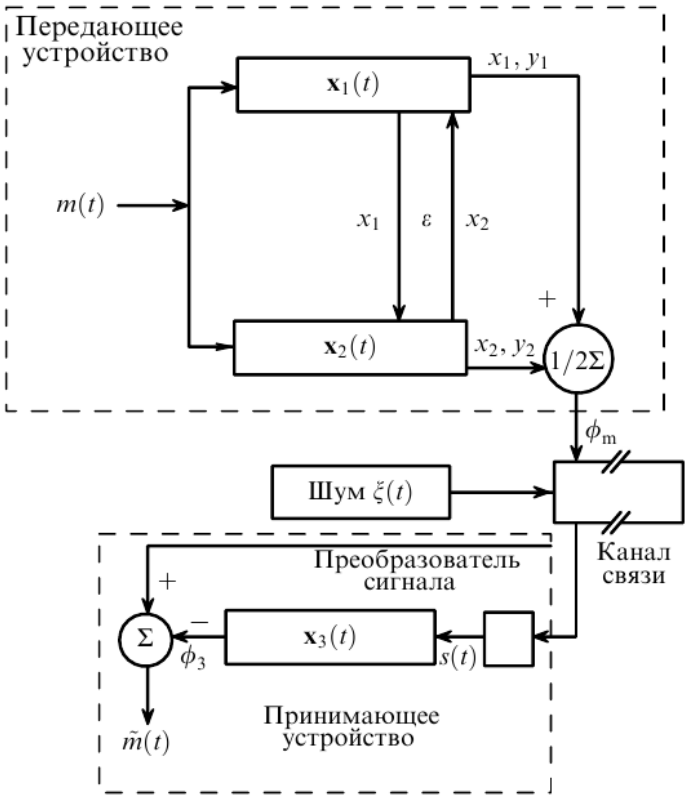


Рисунок 11.18 – Схема скрытой передачи информации на основе фазовой хаотической синхронизации.

Один из управляющих параметров этих генераторов (один и тот же в обеих системах) модулируется полезным цифровым сигналом $m(t)$. В качестве передаваемого сигнала используется мгновенная фаза $\phi_m(t)$ сигнала $x_m(t)$, представляющего собой среднее значение сигналов $x_{1,2}(t)$, генерируемых этими системами. Полученный таким образом сигнал $\phi_m(t)$, содержащий полезную информацию, передаётся по каналу связи (в котором он подвергается влиянию шумов) на принимающее устройство, содержащее хаотический генератор $x_3(t)$, идентичный генераторам передающего устройства, что обеспечивает возникновение режима фазовой синхронизации между ними. В качестве сигнала, непосредственно воздействующего на принимающий генератор хаоса, используется сигнал $s(t)$. Восстановленный сигнал $\tilde{m}(t)$ получают в результате анализа поведения разности фаз $\Delta\phi = \phi_m - \phi_3$ соответствующих сигналов.

Как видно из приведённого описания схемы скрытой передачи информации на основе фазовой синхронизации, принцип её работы существенно отличается от принципа работы схем, рассмотренных в разделе 11.7.1. Тем не менее большая часть недостатков, свойственных схемам на основе полной хаотической синхронизации, здесь остаётся. Кроме того, этот способ обладает существенными дополнительными сложностями с точки зрения технической реализации (например, экспериментальное определение

фазы хаотических сигналов, создание сигнала $s(t)$, наличие дополнительных идентичных генераторов на различных сторонах канала связи). Поэтому на этой схеме мы более подробно останавливаться не будем.

11.7.4. Сверхустойчивый к шумам способ скрытой передачи информации

Анализ схем, рассмотренных выше, показывает, что, несмотря на использование различных типов синхронного поведения для скрытой передачи информации, специфические особенности этих способов, их характерные различия, достоинства и недостатки в той или иной степени присущи всем известным сейчас схемам. Это в первую очередь:

– требование высокой степени идентичности к хаотическим генераторам, располагающимся на разных сторонах канала связи;

– низкая устойчивость к шумам в канале связи;

– низкая конфиденциальность, т.е. возможность в ряде случаев реконструкции параметров передающего генератора по сигналу, передаваемому по каналу связи (особенно для схем на основе полной хаотической синхронизации), с последующим восстановлением информационного сигнала.

В этом подразделе мы рассмотрим способ скрытой передачи информации, который во многом лишён всех вышеупомянутых недостатков. Более того, этот способ обладает значительной устойчивостью к шумам и, как следствие, характеризуется достаточно высокой степенью конфиденциальности. Способ основан на обобщённой синхронизации, однако в отличие от способа, рассмотренного в разделе 11.7.2, он учитывает все особенности режима обобщённой синхронизации, и поэтому обладает принципиальными достоинствами по сравнению с известными аналогами. Принципиальная схема реализации такого способа приведена на рисунке 11.19.

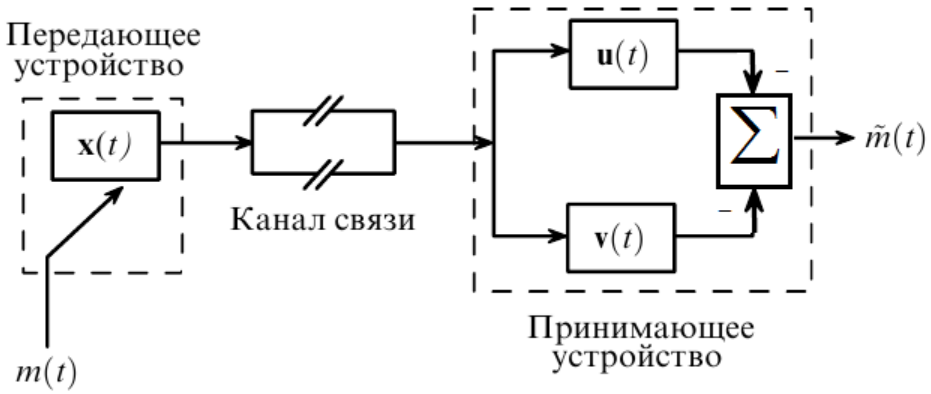


Рисунок 11.19 – Схема реализации сверхустойчивого к шумам способа скрытой передачи информации на основе обобщённой хаотической синхронизации.

Способ скрытой передачи информации заключается в следующем. Информационный сигнал $m(t)$ кодируется в виде бинарного кода. Один или

несколько управляющих параметров передающего генератора $x(t)$ модулируются бинарным сигналом таким образом, чтобы характеристики передаваемого сигнала изменялись незначительно. Полученный таким образом сигнал передаётся по каналу связи. Здесь он подвергается искажению под влиянием шумов. Приёмник, который находится на другой стороне канала связи, представляет собой два идентичных генератора $u(t)$ и $v(t)$, способных находиться в режиме обобщённой синхронизации с передающим генератором. Сигнал с канала связи поступает на генераторы приёмника. Полученные на выходе сигналы проходят через вычитающее устройство, и затем детектируется восстановленный полезный сигнал $\tilde{m}(t)$.

Модуляция управляющих параметров передающего генератора должна быть осуществлена таким образом, чтобы в зависимости от передаваемого бинарного бита 0(1) между передающим и принимающим генераторами существовал (отсутствовал) режим обобщённой синхронизации. Например, если режим обобщённой синхронизации наблюдается в том случае, если передаётся бинарный бит 0, тогда оба принимающих генератора будут демонстрировать идентичные колебания, а после прохождения через вычитающее устройство будет наблюдаться отсутствие хаотических колебаний, т.е. бинарный бит 0. Наоборот, при передаче бинарного бита 1 обобщённая синхронизация не наблюдается, а колебания принимающих генераторов являются неидентичными. Тогда после прохождения через вычитающее устройство будет наблюдаться ненулевая амплитуда хаотических колебаний, т.е. бинарный бит 1.

Принципиальным достоинством рассматриваемого способа скрытой передачи данных является отсутствие требования идентичности генераторов на разных сторонах канала связи. Два идентичных генератора располагаются на принимающей стороне. Следует отметить, что наличие идентичных генераторов на одной стороне канала связи позволяет легко осуществить их юстировку, что снижает требование к степени идентичности генераторов, а следовательно, упрощает техническую реализацию схемы.

Кроме того, сигналы, поступающие на генераторы принимающего устройства, всегда будут одинаковыми, даже при наличии шума в канале связи. Следовательно, шум не должен оказывать сильного влияния на порог возникновения режима обобщённой синхронизации. Эта особенность позволяет говорить о возможности создания устойчивых к шумам способов скрытой передачи данных на основе режима обобщённой синхронизации.

11.7.5. Сравнение известных способов скрытой передачи информации

Проведём сравнительный анализ работоспособности способов скрытой передачи информации с помощью хаотической синхронизации, рассмотренных в настоящем обзоре. Для проверки эффективности этих методов при наличии шума используем численное моделирование и оценим некоторые количественные характеристики работоспособности схем.

Основными количественными характеристиками работоспособности схем скрытой передачи информации являются:

а) Критическое значение SNR_c отношения энергии на бит к спектральной плотности мощности шума (SNR), при котором схема передачи данных становится неработоспособной, т.е. оказывается невозможным восстановление исходного полезного цифрового сигнала $m(t)$ по получаемому на выходе сигналу $\tilde{m}(t)$. Отношение энергии на бит к спектральной плотности шума, которое вводится в рассмотрение для цифровых систем связи, является аналогом отношения сигнал/шум в аналоговой связи:

$$SNR = 10 lg \frac{E_b}{N_0} \quad дБ, \qquad (11.10)$$

где E_b – энергия сигнала, приходящаяся на один бит передаваемой информации, N_0 – спектральная плотность мощности шума. При этом энергия, приходящаяся на один бит, описывается как:

$$E_b = P_x T, \qquad (11.11)$$

где P_x – мощность передаваемого сигнала в отсутствие шума, T – время передачи одного бита, а спектральная мощность шума определяется как:

$$N_0 = \frac{P_ш}{\Delta f}, \qquad (11.12)$$

где $P_ш$ – мощность шума в канале связи, Δf – ширина полосы пропускания канала. Так как шумы неизбежно присутствуют в каналах связи реальных устройств, оценка работоспособности схем передачи информации при наличии шумов является очень важной и актуальной задачей.

б) для характеристики степени устойчивости схем скрытой передачи информации по отношению к внешним шумам в цифровых системах связи достаточно часто используют, наряду с характеристикой, описанной выше, зависимость вероятности ошибки на бит (BER – Bit Error Rate) от отношения энергии на бит к спектральной плотности мощности шума. Вероятность ошибки на бит характеризует качество передачи информации и представляет собой количество ошибок, отнесённое к числу переданных битов. Предположим, что схема корректно передаёт бинарный бит 0 с вероятностью P_{00} и бинарный бит 1 с вероятностью P_{11}. Тогда ошибочное диагностирование бинарного бита 1 при передаче бинарного бита 0 характеризуется вероятностью $P_{01} = 1 - P_{00}$, а вероятность $P_{10} = 1 - P_{11}$ характеризует ошибочное диагностирование бинарного бита 0 при передаче бинарного бита 1. Если символы появляются в передаваемой последовательности с вероятностями P_0 и P_1 соответственно, то вероятность ошибки на бит вычисляется следующим образом:

$$BER = 2(P_{01}P_0 + P_{10}P_1) \tag{11.13}$$

причем вероятности P_{01} и P_{10} зависят от типа и параметров системы связи.

в) максимальное значение РМС расстройки управляющих параметров (РМ, %) генераторов, которые изначально должны быть идентичными. Как уже обсуждалось в разделах 11.7.1 и 11.7.2, в большинстве случаев такие генераторы должны располагаться на различных сторонах канала связи. Ввиду сложности технической реализации таких устройств влияние расстройки их управляющих параметров на эффективность работы способов передачи информации является весьма актуальной проблемой.

г) максимальный уровень нелинейных искажений в канале связи, при котором схема работает:

$$ND = 10 \lg \frac{P_x}{P_y} \quad дБ. \tag{11.14}$$

Здесь P_x – мощность сигнала $x(t)$ на выходе передающего генератора, P_y – мощность сигнала $y(t)$ на входе принимающего устройства. Традиционно в численных расчётах используются нелинейные искажения в виде кубической нелинейности $y = x(1 - ax^2)$.

Для того чтобы количественно сравнить способы скрытой передачи информации, описанные в настоящем разделе, оценим вышеупомянутые характеристики для всех рассмотренных схем.

Результаты расчёта количественных характеристик работоспособности схем представлены в таблице 11.1.

Как видно из таблицы, схема 11.19, описанная в разделе 11.7.4, становится неработоспособной при отношении энергии на бит к спектральной плотности шума $SNR_c = -10,01$ дБ, в то время как для других рассмотренных нами схем SNR_c оказывается положительным. То есть при наличии в канале связи шумов определённого уровня (даже если мощность шумов меньше мощности передаваемого сигнала) большинство схем становится неработоспособным. Понятно, что значения таких характеристик будут меняться от схемы к схеме. Из схем 11.13–11.17 лучшими в этом отношении являются схемы на основе переключения хаотических режимов и модулирования управляющих параметров (схемы 11.14 и 11.16, $SNR_c = 30,76$ дБ). Но положительное значение отношения энергии на бит к спектральной плотности шума свидетельствует об ограниченной устойчивости к шумам и деструктивной роли шума при передаче информации.

Схема 11.19, описанная в разделе 11.7.4, обладает значительной устойчивостью к шумам в канале связи. При этом, ещё более искажая передаваемый сигнал, шум препятствует третьей стороне декодировать информационное сообщение. В этом случае можно говорить о конструктивной роли шума в повышении конфиденциальности передачи информации, тогда как в остальных случаях роль шума является деструктивной.

Таблица 11.1 – Критические значения отношения энергии на бит к спектральной плотности шума SNR_c, расстройки управляющих параметров PM_c изначально идентичных генераторов и уровня нелинейных искажений в канале связи ND_c

№ рисунка	Название схемы	SNR_c, дБ	PM_c, %	ND_c, дБ	Раздел*	Литература
11.13	Хаотическая маскировка	56,48	0,30	1,03	11.7.1.1	[25]
11.14	Переключение хаотических режимов	30,76	2,00	23,3	11.7.1.2	[145]
11.15	Нелинейное подмешивание	64,99	0,30	0,26	11.7.1.3	[146]
11.16	Модулирование управляющих параметров	30,76	2,00	23,3	11.7.1.4	[147]
11.18	Схема на основе режима фазовой синхронизации	32,40	0,80	10,7	11.7.3	[155]
11.17	Схема на основе режима обобщённой синхронизации	39,52	1,00	7,75	11.7.2	[109]
11.19	Сверхустойчивая к шумам схема	–10,01	2,00	27,2	11.7.4	[162]

Справедливость вышеприведённых рассуждений подтверждается также зависимостью вероятности ошибки на бит от спектральной плотности мощности шума для различных схем скрытой передачи информации. Указанные зависимости представлены на рисунке 11.20.

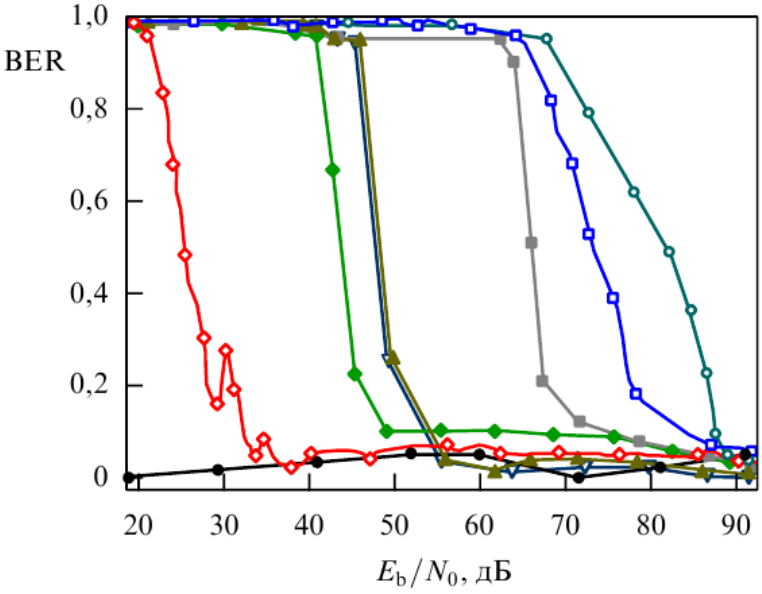


Рисунок 11.20 – Зависимости вероятности ошибки на бит (BER) от отношения энергии на бит к спектральной плотности мощности шума (E_b / N_0)) для различных схем

скрытой передачи информации: ○– хаотическая маскировка, ♦ – переключение хаотических режимов (модулирование управляющих параметров), ■ – нелинейное подмешивание, ◊ – схема на основе режима фазовой синхронизации (кривая частично перенесена из работы [155], ▲ – схема на основе режима обобщённой синхронизации.

При расчёте вероятности ошибки на бит пороговое значение, позволяющее восстановить исходную последовательность бинарных битов по сигналу $\tilde{m}(t)$, выбиралось фиксированным независимо от интенсивности шума, воздействующего на систему, тогда как при определении характеристик, представленных в таблице, оно менялось. В то же время, как видно из рисунка 11.20, для различных способов скрытой передачи информации (схемы 11.13–11.18 таблицы) вероятность ошибки на бит достаточно быстро становится равной 1, тогда как для сверхустойчивого к шуму способа (схема 11.19) она оказывается близкой к 0, вне зависимости от интенсивности шума, воздействующего на систему, что достаточно хорошо согласуется с результатами, представленными выше.

Оценим теперь влияние расстройки управляющих параметров на эффективность работы рассмотренных способов скрытой передачи данных.

Подобные оценки позволяют заключить, что схема на основе режима обобщённой синхронизации (см. раздел 11.7.4) будет оставаться работоспособной до тех пор, пока генераторы принимающего устройства не будут расстроены более чем до 2 % по параметру ω_u. Конечно, это не столь большая величина, и в этом отношении рассматриваемая схема имеет конкурентов, которыми снова являются схемы передачи информации на основе переключения хаотических режимов и модулирования управляющих параметров (схемы 11.14 и 11.16 в таблице соответственно). Однако схема 11.19 и с этой точки зрения обладает принципиальным преимуществом перед схемами 11.14 и 11.16. Изначально идентичные хаотические генераторы в схемах 2 и 4 таблицы (так же как и во всех остальных, кроме схемы 9) должны располагаться на различных сторонах канала связи (для возможности реализации режима полной синхронизации между ними). В схеме 11.19 идентичные генераторы располагаются только на принимающей стороне канала связи, что позволяет легко осуществить при необходимости их юстировку.

Что касается устойчивости к нелинейным искажениям в канале связи, то и по этой характеристике рассмотренная в разделе 11.7.4 схема превосходит все известные аналоги. Понятно, что чем больше влияние нелинейных искажений на сигнал, тем выше должен быть максимально допустимый уровень нелинейных искажений, при котором способ скрытой передачи данных будет оставаться ещё работоспособным. Как видно из таблицы, максимальный уровень нелинейных искажений для схемы 11.19 $ND_c = 27,2$ дБ. Наиболее близкими показателями опять обладают способы скрытой передачи информации на основе переключения хаотических режимов и модулирования

управляющих параметров (схемы 11.14 и 11.16), однако устойчивость схемы 11.19 к нелинейным искажениям оказывается несколько выше. Кроме того, схемы 11.14 и 11.16 обладают ограниченной устойчивостью к шумам, в то время как устойчивость схемы 11.19 является практически неограниченной в реальных пределах.

11.7.6. Экспериментальная реализация схем передачи информации с помощью хаотической синхронизации

Следует отметить, что большинство публикаций в этой области свидетельствуют о том, что работы по применению хаотической синхронизации носит теоретический и лабораторно–экспериментальный характер. Наиболее продвинутыми являются работы по скрытой передаче информации на большие расстояния с помощью хаотической синхронизации на основе существующих коммерческих волоконно–оптических каналов связи.

Остановимся на описании этого эксперимента более подробно, так как данную работу следует признать одной из наиболее интересных на сегодня экспериментальных реализаций идей передачи информации на основе полной хаотической синхронизации в оптическом диапазоне.

В качестве источников хаотических сигналов с большой размерностью аттрактора и высокой информационной энтропией использовались генераторы на основе полупроводниковых диодов с запаздывающей обратной связью, которая реализовывалась двумя различными способами, а именно, рассматривались электронно– оптическая и полностью оптическая обратные связи. Использование двух типов обратной связи в генераторе хаоса оптического диапазона позволило реализовать два способа передачи информации, описанных выше: нелинейное подмешивание информационного сигнала к хаотическому и модулирование параметров хаотического сигнала информационным сигналом. В первом случае (схема соответствующего эксперимента приведена на рисунке 11.21) излучение лазерного диода ЬО проходит через интегрированный электро–оптический интерферометр Маха–Зендера MZI, который управляется электро–оптической запаздывающей обратной связью (включающей в себя линию задержки DL, фотодиод PD и электронный усилитель A) и работает как нелинейный модулятор лазерного излучения. Информационный сигнал в этом случае подмешивается к сигналу обратной связи через волоконно–оптический смеситель OFC. Выходной хаотический сигнал генератора, содержащий информационное сообщение, дополнительно усиливается перед подачей в канал связи для достижения необходимого уровня мощности. Во втором случае (рисунок 11.22) источником лазерного излучения снова является лазерный диод LD, а оптическая обратная связь реализуется с помощью зеркала R, коэффициент отражения которого нелинейно зависит от интенсивности падающего излучения. Длина внешнего резонатора системы составляла 6 м.

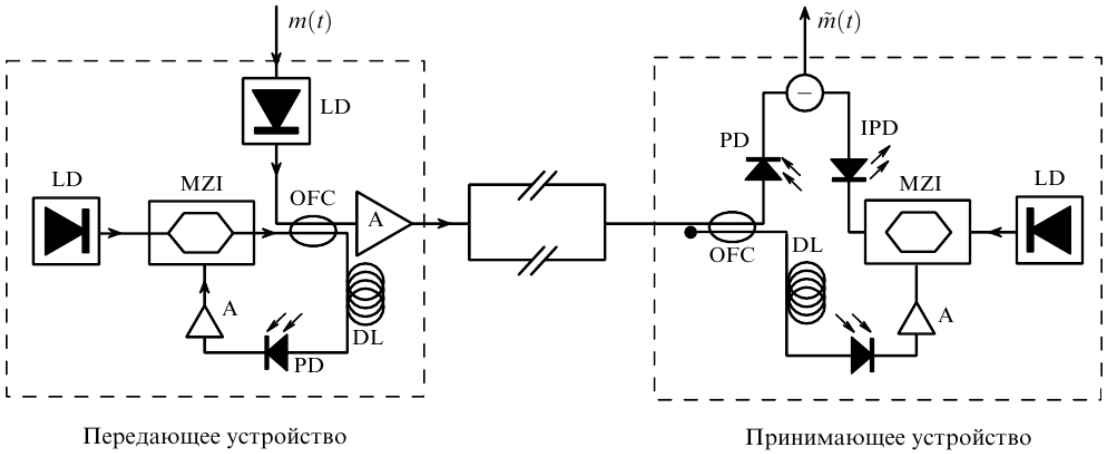


Рисунок 11.21 – Экспериментальная схема скрытой передачи информации в оптическом диапазоне на основе нелинейного подмешивания сигнала к хаотическому

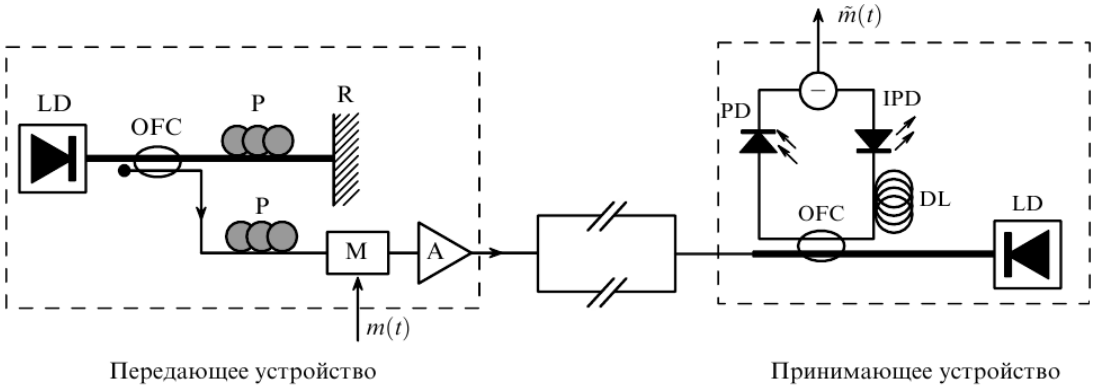


Рисунок 11.22 – Экспериментальная схема скрытой передачи информации в оптическом диапазоне на основе модулирования параметров передающего генератора информационным сигналом

В резонатор помещался поляризатор P для обеспечения необходимой поляризации света, отражённого от зеркала R с переменным коэффициентом отражения. Информационный сигнал вводился посредством модуляции параметров хаотического сигнала с помощью модулятора M. Затем передаваемый по оптическому каналу сигнал усиливался, как и в первой схеме. В схемах также имелись не показанные на рисунках фильтры для подавления влияния шумов, связанных со спонтанной эмиссией.

В обеих схемах полезный информационный сигнал декодировался в принимающем устройстве путём достижения режима полной хаотической синхронизации между генераторами на различных сторонах канала связи. Основной проблемой эксперимента стало создание близких к идентичным генераторов на обоих концах канала связи. В обеих схемах наибольшие сложности при достижении идентичности генераторов вызывали активные элементы генераторных схем, а именно полупроводниковые лазеры. В экспериментах использовались лазеры с длинами волн 1552,0 и 1552,9 нм в приёмном и передающем устройствах соответственно. Для обеспечения одинаковых длин волн и стабильной

работы лазерных диодов для каждого из лазеров подбирался свой температурный режим, который поддерживался в течение всего эксперимента. Подбор с высокой степенью идентичных пассивных элементов не вызывал принципиальных сложностей. При достаточно точной настройке параметров оптических генераторов хаоса в однонаправленно связанной системе наблюдалось устойчивое возникновение режимов полной хаотической синхронизации при достаточной мощности сигнала, подаваемого на принимающее устройство [232]. Было показано, что расстройка генераторов на различных сторонах канала связи для устойчивости работы схемы передачи информации (для достижения полной хаотической синхронизации) не может превышать 3 %. При передаче сигнала на большие расстояния возникала проблема разрушения режима синхронизации из–за влияния дисперсии в волоконно–оптическом канале связи (она составляла в рассматриваемом эксперименте порядка –850 пс $нм^{-1}$). В связи с этим на выходе канала используемой коммерческой волоконно-оптической сети перед подачей сигнала на приёмное устройство вводились отрезки волоконно- оптических линий с дисперсией другого знака, компенсирующие дисперсионное искажение сигнала, а также дополнительные усилители для обеспечения необходимого уровня мощности (данные элементы не показаны на рисунках 11.21 и 11.22).

Вероятность ошибки на бит (BER) для данной экспериментальной схемы с нелинейным подмешиванием информационного сигнала к хаотическому составляет 10^{-7} при скорости передачи информации 3 Гб $с^{-1}$. Аналогичные результаты получены и при использовании схемы с модулированием параметров, однако при той же величине BER = 10^{-7} скорость передачи оказалась почти в три раза меньше ($\sim$ 1 Гб $с^{-1}$).

Полученные результаты представляются весьма важными и открывают серьёзные перспективы применения схем передачи информации с использованием хаотической синхронизации в оптическом диапазоне в современных информационно–телекоммуникационных системах. Все исследования были выполнены на основе уже существующих телекоммуникационных сетей, т.е. внедрение соответствующих технологий не требует принципиальной замены уже созданного оборудования, что чрезвычайно важно для практического использования. К недостаткам рассматриваемой схемы следует отнести сравнительно низкую скорость передачи данных, что, как обсуждалось выше, связано с общими недостатками подобных схем скрытой передачи информации, а именно с необходимостью достижения режимов полной хаотической синхронизации, которые весьма чувствительны к расстройке хаотических генераторов на различных сторонах телекоммуникационного канала и шумам, всегда присутствующим в реальных экспериментах.

В заключение раздела 11.7 следует сказать, что генераторы хаоса могут быть построены на основе систем Росслера, Чуя, Лоренца, кольцевых схем, клистронов, и лазеров.

12. КОДИРОВАНИЕ ИНФОРМАЦИИ ПРИ ПЕРЕДАЧЕ ПО ДИСКРЕТНОМУ КАНАЛУ С ПОМЕХАМИ

12.1. Постановка задачи

Действие помехи на дискретный сигнал с конечным числом элементов приводит к количественным и качественным изменениям его структуры (к ошибкам в числе и состоянии элементов). Чтобы обеспечить в таких условиях передачу информации с заданной достоверностью, возможные ошибки должны быть обнаружены, а если требуется, то и исправлены.

Теория помехоустойчивого кодирования базируется на результатах исследований, проведенных Шенноном и сформулированных им в виде теоремы:

1. При любой производительности источника сообщений, меньшей, чем пропускная способность канала, существует такой способ кодирования, который позволяет обеспечить передачу всей информации, создаваемой источником сообщений, со сколь угодно малой вероятностью ошибки.

2. Не существует способа кодирования, позволяющего вести передачу информации со сколь угодно малой вероятностью ошибки, если производительность источника сообщений больше пропускной способности канала.

С доказательством теоремы можно ознакомиться в [4]. Теорема устанавливает теоретический предел возможной эффективности системы при достоверной передаче информации. Ею опровергнуто казавшееся интуитивно правильным представление о том, что достижение сколь угодно малой вероятности ошибки в случае передачи информации по каналу с помехами возможно лишь при введении бесконечно большой избыточности, то есть при уменьшении скорости передачи до нуля. Из теоремы следует, что помехи в канале не накладывают ограничений на точность передачи. Ограничение накладывается только на скорость передачи, при которой может быть достигнута сколь угодна высокая достоверность передачи.

Теорема неконструктивна в том смысле, что в ней не затрагивается вопрос о путях построения кодов, обеспечивающих указанную идеальную передачу.

Следует отметить, что при любой конечной скорости передачи информации вплоть до пропускной способности сколь угодно малая вероятность ошибки достигается лишь при безграничном увеличении длительности кодируемых последовательностей знаков. Таким образом, безошибочная передача при наличии помех возможна лишь теоретически. На практике степень достоверности и эффективности ограничивается двумя факторами: размерами и стоимостью аппаратуры кодирования и временем задержки передаваемого сообщения. В настоящее время используются относительно простые методы кодирования, которые не реализуют возможностей, указанных теорией. Однако постоянно растущие требования к достоверности передачи и внедрение специализированных больших интегральных схем позволяют надеяться на то, что будут созданы эффективные устройства кодирования.

Таким образом, помехоустойчивое кодирование представляет собой обширную область теоретических и прикладных исследований. К числу основных задач, возникающих при этом, относятся отыскание кодов, эффективно исправляющих ошибки требуемого вида, нахождение методов кодирования и декодирования, способов их реализации.

Все коды разделяются на коды с обнаружением ошибок и коды с исправлением ошибок (корректирующие коды).

Задачи корректирующего кодирования обычно решают при следующих предположениях: избыточность эффективного кода равна нулю, кодирование выполняется двоичными сигналами, характеристики дискретного двоичного канала известны, канал является симметричным, то есть вероятности перехода нуля в единицу и единицы в нуль одинаковы.

Коды с обнаружением и исправлением ошибок подробно будут рассматриваться в курсе «Телемеханика». Здесь мы остановимся лишь на основных характеристиках корректирующих кодов и на базе одного из кодов покажем принцип обнаружения и исправления искажений.

12.2. Классификация корректирующих кодов

Для коррекции ошибок неравномерные коды почти не применяют, поэтому в дальнейшем рассматриваются только равномерные корректирующие коды. Их общая классификация приведена на рис. 12.1.

Корректирующие коды делятся на два больших класса: блочные и непрерывные. Кодовая последовательность блочных кодов состоит из отдельных комбинаций (блоков), которые кодируются и декодируются независимо.

Непрерывные коды представляют непрерывную последовательность кодовых символов, ее разделение на отдельные кодовые комбинации не производится. Блочные и непрерывные коды бывают разделимыми и неразделимыми. В разделимых блочных кодах информационные и проверочные символы занимают всегда одни и те же определенные позиции (разряды). Обозначают эти коды как (n,k) – коды, где n – длина комбинации, k – число информационных символов. В неразделимых кодах нельзя точно указать место информационных и проверочных разрядов. Представителем этого класса являются сверточные (рекуррентные коды).

Среди разделимых кодов выделяют систематические и несистематические. Систематическими кодами называют (n, k) – коды, в которых $r = n - k$ проверочных символов являются линейными комбинациями информационных. Такое формирование кодовых комбинаций существенно упрощает техническую реализацию устройств кодирования и декодирования – кодеков. Поэтому систематические коды являются одними из наиболее распространенных. Так как новую разрешенную кодовую комбинацию можно получить линейным преобразованием двух других разрешенных комбинаций, то такие коды часто называют линейными.

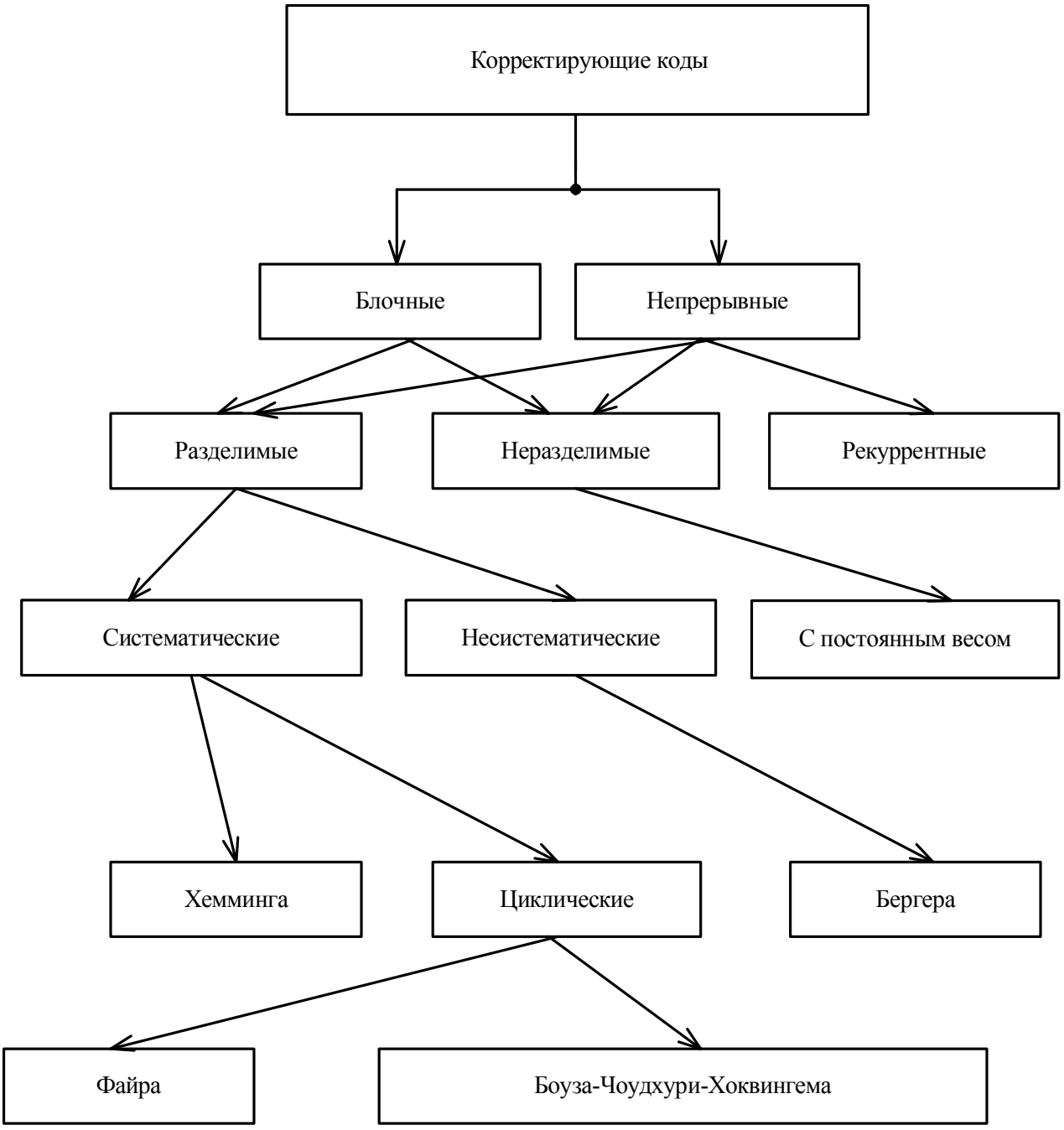


Рис. 12.1. Классификация корректирующих кодов

12.3. Основные характеристики корректирующих кодов

К основным характеристикам корректирующих кодов относят: избыточность кода, кодовое расстояние, число обнаруживаемых и исправляемых ошибок, корректирующие возможности кодов.

Принцип обнаружения и исправления ошибок кодами хорошо иллюстрируется с помощью геометрических моделей. Любой n – элементный двоичный код можно представить n–мерным кубом (рис.12.2), в котором каждая вершина отображает кодовую комбинацию, а длина ребра куба соответствует одной

единице. В таком кубе расстояние между вершинами (кодовыми комбинациями) измеряется минимальным количеством ребер, находящимся между ними, обозначается d и называется кодовым расстоянием Хемминга.

Таким образом, кодовое расстояние – это минимальное число элементов, в которых любая кодовая комбинация отличается от другой (по всем парам кодовых слов). Например, код состоит из комбинаций 1011, 1101, 1000 и 1100. Сравнивая первые две комбинации, путем сложения их по модулю два находим, что $d = 2$. Сравнение первой и третьей комбинаций показывает, что и в этом случае $d = 2$. Наибольшее значение $d = 3$ обнаруживается при сравнении первой и четвертой комбинаций, а наименьшее $d = 1$ – второй и четвертой, третьей и четвертой комбинаций. Таким образом, для данного кода минимум расстояния $d_{\min} = 1$.

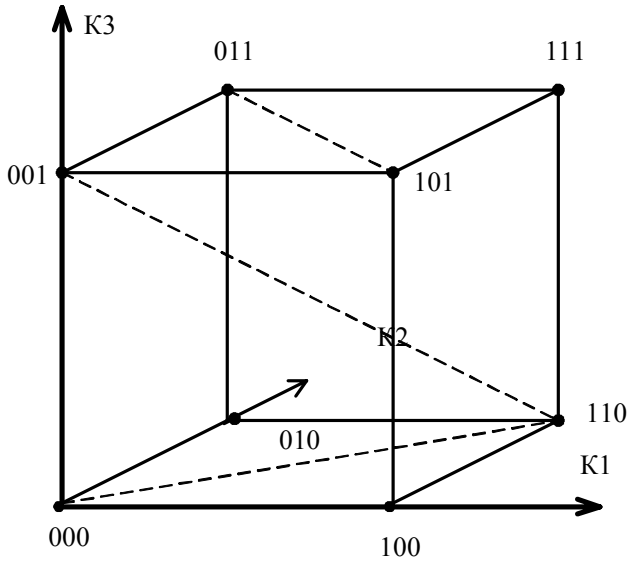


Рис. 12.2. Геометрическая модель двоичных кодов

В общем случае кодовое расстояние между двумя комбинациями двоичного кода равно числу единиц, полученных при сложении этих комбинаций по модулю два. Такое определение кодового расстояния удобно при большой разрядности кодов. Так, складывая комбинации

$$\begin{array}{r} 1010111 \\ \oplus \\ \underline{0101101} \\ 1111010 \end{array},$$

определим, что кодовое расстояние между ними $d = 5$.

Если использовать все восемь кодовых комбинаций, записанных в вершинах куба, то образуется двоичный код на все сочетания. Такой код является непомехоустойчивым, так как любое искажение приводит к кодовой комбинации, принадлежащей данному сомножеству, а следовательно, искажение не будет обнаружено. Если же уменьшить число используемых комбинаций с восьми до четырех, то появится возможность обнаружения одиночных ошибок. Для этого

выберем только такие комбинации, которые отстоят друг от друга на расстоянии $d = 2$, например, 000, 110, 011, 101. Остальные кодовые комбинации не используются. Если передавалась комбинация 101, а принята комбинация 100, то очевидно, что при приеме произошла ошибка. Таким образом, для обнаружения искажений необходимо все кодовые комбинации разделить на две группы: разрешенные и запрещенные.

Из приведенного примера можно сделать вывод, что способность кода обнаруживать и исправлять ошибки обусловлена наличием избыточности, которая обеспечивает минимальное расстояние между любыми двумя разрешенными комбинациями $d_{\min} >= 2$, т.е. к исходной k – элементной комбинации необходимо добавить r контрольных символов, сформированных по определенным правилам.

Пусть на вход кодирующего устройства поступает последовательность из k информационных двоичных символов. На выходе ей соответствует последовательность из $n = k + r$ двоичных символов.

Всего может быть 2^k разрешенных кодовых комбинаций и 2^n различных выходных последовательностей, а следовательно, $2^n - 2^k$ возможных выходных последовательностей для передачи не используются и являются запрещенными комбинациями.

Искажение информации в процессе передачи сводится к тому, что некоторые из переданных символов заменяются другими. Так как каждая из 2^k разрешенных комбинаций в результате действия помех может трансформироваться в любую другую, то всего имеется $2^k \times 2^n$ возможных случаев передачи (рис. 10.3). В это число входят: 2^k случаев безошибочной передачи (на рис. 10.3 обозначены жирными линиями):

$2^k(2^k - 1)$ случаев перехода в другие разрешенные комбинации, что соответствует необнаруженным ошибкам (на рис. 10.3 обозначены пунктирными линиями);

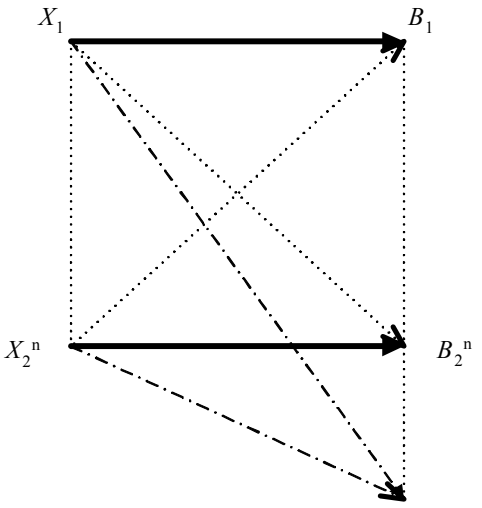


Рис. 12.3. Возможные варианты трансформаций кодовых комбинаций X_i в B_j

$2^k(2^n - 2^k)$ случаев перехода в запрещенные комбинации, которые могут быть обнаружены (на рис. 12.3 обозначены штрихпунктирными линиями). Следовательно, число обнаруживаемых ошибочных кодовых комбинаций от общего числа возможных случаев передачи составляет:

$$\frac{2^k(2^n - 2^k)}{2^k * 2^n} = 1 - \frac{2^k}{2^n}. \qquad (12.1)$$

Отношение числа исправляемых кодом ошибочных кодовых комбинаций к числу обнаруживаемых ошибочных комбинаций равно:

$$\frac{2^n - 2^k}{2^k(2^n - 2^k)} = \frac{1}{2^k}. \qquad (12.2)$$

Из приведенных выше рассуждений можно сделать вывод, что коррекция ошибок возможна благодаря введению избыточности, которая определяется выражением:

$$R_r = \frac{r}{n} = \frac{n-k}{n} = 1 - \frac{k}{n}. \qquad (12.3)$$

Эта величина показывает, какую часть от общего числа символов кодовой комбинации составляют контрольные символы. В теории кодирования величину k/n принято называть скоростью передачи [5]. Необходимо отметить, что величина k/n характеризует относительную скорость передачи информации. Если производительность источника информации равна $\overline{H}(x)$ бит/с, то скорость передачи после кодирования этой информации окажется равной

$$V_k = \frac{\overline{H}(x)k}{n}. \qquad (12.4)$$

Для обнаружения или исправления значительного числа ошибок, необходимо иметь код с большим числом проверочных символов. При этом существенно возрастает длительность кодовых блоков, что приводит к задержке информации при передаче и приеме.

Если длительность помехи превосходит длительность элементарного сигнала, то искажается несколько символов, и такой вид искажений называется пакетом (пачкой) искажений. Под пакетом искажений длиной b понимается такой вид комбинации помехи, в которой между крайними разрядами, пораженными помехами, содержатся $b - 2$ разряда. Например, при $b = 4$ пакет искажений может иметь вид: 1001 (поражены только два крайних символа), 1111 (поражены все символы), 1011 (не поражен лишь один символ). При любом варианте непременным условием пакета данной длины является поражение крайних символов.

Кодовое расстояние, как отмечалось выше, является основной характеристикой корректирующей способности данного кода. Если код используется

только для обнаружения ошибок кратности m, то необходимо и достаточно, чтобы минимальное расстояние удовлетворяло условию

$$d_{\min} \geq m + 1. \tag{12.5}$$

Под кратностью ошибки понимают число позиций кодовой комбинации, в которых под действием помехи одни символы оказались замененными другими.

Для исправления ошибки кратностью S необходимо, чтобы минимальное расстояние удовлетворяло условию

$$d_{\min} \geq 2S + 1. \tag{10.6.}$$

Корректирующие коды можно одновременно использовать и для обнаружения и для исправления ошибок. В этом случае кодовое расстояние должно удовлетворять условию

$$d_{\min} = m + S + 1, \tag{10.7}$$

где всегда должно выполняться условие $m > S$.

Вопрос о минимально необходимой избыточности, при которой код обладает нужными корректирующими свойствами, является одним из важнейших в теории кодирования. Для некоторых частных случаев Хемминг указал простые соотношения, позволяющие определить необходимое число проверочных символов:

$$r_{d=3} \geq \in \log(n+1), \tag{12.8}$$

$$r_{d=3} \geq \in \log((k+1) + \in \log(k+1)), \tag{12.9}$$

$$r_{d=4} \geq \in \log 2n, \tag{12.10}$$

где знак $\in$ означает округление в большую сторону.

12.4. Способы введения избыточности в сигнал

Для корректирования ошибок можно применять те же способы, что и для повышения скорости передачи информации. Все они направлены на увеличение объема сигнала и приближение его к объему канала. Если объем сигнала равен объему канала, то корректирования ошибок можно добиться только путем уменьшения скорости передачи информации, так как часть сигналов может быть использована для корректирования. Корректирующее кодирование использует по существу все виды избыточности сигналов: временную, частотную и энергетическую. Если длина кодовой комбинации не фиксирована (скорость передачи информации не фиксирована), то для корректирования ошибок используют временную избыточность – кроме информационных символов, дополнительно вводят еще проверочные.

Если скорость передачи информации фиксирована, ввести проверочные символы в кодовую комбинацию бинарного кода можно, лишь уменьшая длительность элементарных сигналов, что ведет к расширению их спектра. Следовательно, в этом случае корректирующее кодирование использует частотную избыточность. Чтобы отношение сигнал/шум с уменьшением длительности импульсов не падало, необходимо увеличить амплитуду импульсов. Увеличивая амплитуду укороченного импульса, можно настолько увеличить его энергию, что вероятность ошибки при его приеме уменьшится по сравнению с вероятностью при приеме импульса неукороченной длительности. Так вводится энергетическая избыточность.

12.5. Систематические коды

Систематическими кодами называются блочные (n, k) коды, у которых k (обычно первые) разряды представляют собой двоичный неизбыточный код, а последующие r–контрольные разряды сформированы путем линейных комбинаций над информационными.

Основное свойство систематических кодов: сумма по модулю два двух и более разрешенных кодовых комбинаций также дает разрешенную кодовую комбинацию.

Правило формирования кода обычно выбирают так, чтобы при декодировании имелась возможность выполнить ряд проверок на четность для некоторых определенным образом выбранных подмножеств информационных и контрольных символов каждой кодовой комбинации. Анализируя результаты проверок, можно обнаружить или исправить ошибку ожидаемого вида.

Информацию о способе построения такого кода содержит проверочная матрица, которая составляется на базе образующей матрицы.

Образующая матрица M состоит из единичной матрицы размерностью $k \times k$ и приписанной к ней справа матрицы дополнений размерностью $k \times r$:

$$M = \left\| \begin{array}{ccccc|ccc} 1 & 0 & 0 & \ldots & 0 & b_{11} & b_{12} & \cdots b_{1r} \\ 0 & 1 & 0 & \ldots & 0 & b_{21} & b_{22} & \cdots b_{2r} \\ 0 & 0 & 1 & \ldots & 0 & b_{31} & b_{32} & \cdots b_{3r} \\ \multicolumn{5}{c|}{\ldots\ldots\ldots\ldots} & \cdots & \cdots & \cdots \cdots \\ 0 & 0 & 0 & \ldots & 1 & b_{k1} & b_{k2} & \cdots b_{kr} \end{array} \right\|. \qquad (12.11)$$

Размерность матрицы дополнений выбирается из выражения (12.8) или (12.9). Причем вес w (число ненулевых элементов) каждой строки матрицы дополнений должен быть не меньше, чем $d_{\min} - 1$.

Проверочная матрица N строится из образующей матрицы следующим образом. Строками проверочной матрицы являются столбцы матрицы дополнений образующей матрицы. К полученной матрице дописывается справа единичная

матрица размерностью $r \times r$. Таким образом, проверочная матрица размерностью $r \times k$ имеет вид:

$$N = \begin{Vmatrix} b_{11}\, b_{21}\, b_{31} \cdots b_{k1} & 1\ 0\ 0 \cdots 0 \\ b_{12}\, b_{22}\, b_{32} \cdots b_{k2} & 0\ 1\ 0 \cdots 0 \\ \cdots\ \cdots\ \cdots \cdots \cdots & \cdots\cdots\cdots\cdots \\ b_{1r}\, b_{2r}\, b_{3r} \cdots b_{kr} & 0\ 0\ 0 \cdots 1 \end{Vmatrix}. \qquad (12.12)$$

Единицы, стоящие в каждой строке, однозначно определяют, какие символы должны участвовать в определении значения контрольного разряда. Причем единицы в единичной матрице определяют номера контрольных разрядов.

Пример 12.1. Получить алгоритм кодирования в систематическом коде всех четырехразрядных кодовых комбинаций, позволяющий исправлять единичную ошибку. Таким образом, задано число информационных символов $k = 4$ и кратность исправлений $S = 1$. По выражению (12.9) определим число контрольных символов:

$$r \geq \in \log((4 + 1) + \in \log(4 + 1)) = 3.$$

Минимальное кодовое расстояние определим из выражения (12.6):

$$d_{\min} \geq 2 \cdot 1 + 1 = 3.$$

Строим образующую матрицу:

$$M = \begin{Vmatrix} 1\ 0\ 0\ 0 & 0\ 1\ 1 \\ 0\ 1\ 0\ 0 & 1\ 1\ 0 \\ 0\ 0\ 1\ 0 & 1\ 0\ 1 \\ 0\ 0\ 0\ 1 & 1\ 1\ 1 \end{Vmatrix}.$$

Проверочная матрица будет иметь вид

$$N = \begin{Vmatrix} a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7 \\ 0\ 1\ 1\ 1\ 1\ 0\ 0 \\ 1\ 1\ 0\ 1\ 0\ 1\ 0 \\ 1\ 0\ 1\ 1\ 0\ 0\ 1 \end{Vmatrix}.$$

Обозначим символы, стоящие в каждой строке, через $a_i(a_1 a_2 a_3 a_4 a_5 a_6 a_7)$. Символы a_5, a_6 и a_7 примем за контрольные, так как они будут входить только в одну из проверок.

Составим проверки для каждого контрольного символа. Из первой строки имеем

$$a_5 = a_2 \oplus a_3 \oplus a_4. \qquad (12.13)$$

Из второй строки получим алгоритм для формирования контрольного символа a_6:

$$a_6 = a_1 \oplus a_2 \oplus a_4. \qquad (12.14)$$

Аналогично из третьей строки получим алгоритм для формирования контрольного символа a_7:

$$a_7 = a_1 \oplus a_3 \oplus a_4. \qquad (12.15)$$

Нетрудно убедиться, что все результаты проверок на четность по выражениям (12.13)–(12.15) дают нуль, что свидетельствует о правильности составления образующей и проверочной матриц.

Пример 12.2. На основании алгоритма, полученного в примере 12.3, закодировать кодовую комбинацию $G(x) = 1101 = a_1a_2a_3a_4$ в систематическом коде, позволяющим исправлять одиночную ошибку. По выражениям (12.13), (12.14) и (12.15) найдем значения для контрольных символов a_5, a_6 и a_7:

$$a_5 = 1 \oplus 0 \oplus 1 = 0,$$
$$a_6 = 1 \oplus 1 \oplus 1 = 1,$$
$$a_7 = 1 \oplus 0 \oplus 1 = 0. \qquad (12.16)$$

Таким образом, кодовая комбинация $F(X)$ (10.16) в систематическом коде будет иметь вид

$$F(X) = 1\ 1\ 0\ 1\ 0\ 1\ 0\ . \qquad (12.17)$$

На приемной стороне производятся проверки *Si* принятой кодовой комбинации, которые составляются на основании выражений (12.13), (12.14) и (12.15):

$$S_1 = a_2 \oplus a_3 \oplus a_4 \oplus a_5,$$
$$S_2 = a_1 \oplus a_2 \oplus a_4 \oplus a_6,$$
$$S_3 = a_1 \oplus a_3 \oplus a_4 \oplus a_7.$$

Если синдром (результат проверок на четность) $S_1S_2S_3$ будет нулевого порядка, то искажений в принятой кодовой комбинации $F'(X)$ нет. При наличии искажений синдром $S_1S_2S_3$ указывает на искаженный символ.

Рассмотрим всевозможные состояния $S_1S_2S_3$:

$S_1\ S_2\ S_3$
0 0 0 – искажений нет,
1 0 0 – искажен символ a_5,
0 1 0 – искажен символ a_6,
0 0 1 – искажен символ a_7,
1 1 0 – искажен символ a_2,
0 1 1 – искажен символ a_1,
1 1 1 – искажен символ a_4,
1 0 1 – искажен символ a_3. (12.19)

Пример 12.3. Кодовая комбинация $F(X)$ = 1 1 0 1 0 1 0 (пример 12.2) при передаче была искажена и приняла вид $F'(X)$ = 1 1 1 1 0 1 0 = $a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7$. Декодировать принятую кодовую комбинацию.

Произведем проверки согласно выражениям (10.18):

$$S_1 = 1 \oplus 1 \oplus 1 \oplus 0 = 1,$$
$$S_2 = 1 \oplus 1 \oplus 1 \oplus 1 = 0,$$
$$S_3 = 1 \oplus 1 \oplus 1 \oplus 0 = 1.$$

Полученный синдром $S_1S_2S_3$ = 101 согласно (10.19) свидетельствует об искажении символа a_3. Заменяем этот символ на противоположный и получаем исправленную кодовую комбинацию $F(X)$ = 1 1 0 1 0 1 0, а исходная кодовая комбинация имеет $G(X)$ = 1 1 0 1, что совпадает с кодовой комбинацией, подлежащей кодированию в примере 12.2.

12.6. Рекуррентные коды

Эти коды относятся к непрерывным кодам, в которых операции кодирования и декодирования производятся непрерывно над последовательностью информационных символов без деления на блоки. Рекуррентные коды применяются для обнаружения и исправления пакетов ошибок. В данном коде после каждого информационного элемента следует проверочный элемент. Проверочные элементы формируются путем сложения по модулю два двух информационных элементов, отстоящих друг от друга на шаг сложения, равный b.

Рассмотрим процесс кодирования на примере кодовой комбинации, приведенный на рис. 12.4 (верхняя строка), если шаг сложения b = 2. Процесс образования контрольных символов показан на этом же рисунке (нижняя строка).

Кодирование производимое кодером, схема которого приведена на рис. 12.5. Как видно из рисунка 12.6 входные символы задерживаются на 2b тактов, что эквивалентно дописыванию 2b нулей перед входной кодовой комбинацией (рис. 12.4).

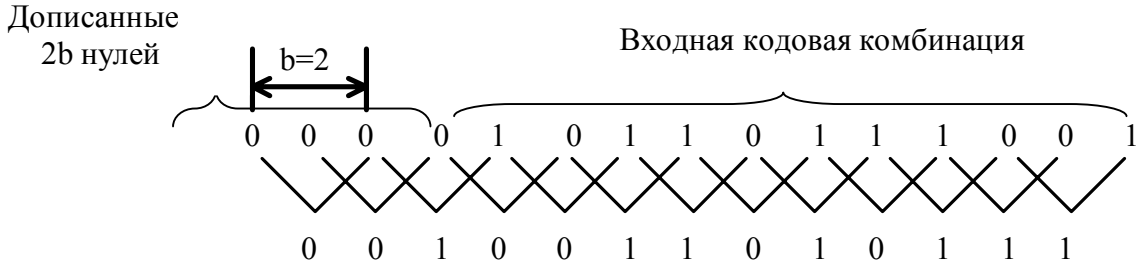


Рис. 12.4. Схема построения рекуррентного кода

Кодирование и декодирование производятся с помощью регистров сдвига и сумматоров по модулю два.

На выходе кодирующего устройства (рис. 12.5) получим последовательность символов

$$1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ . \qquad (12.20)$$

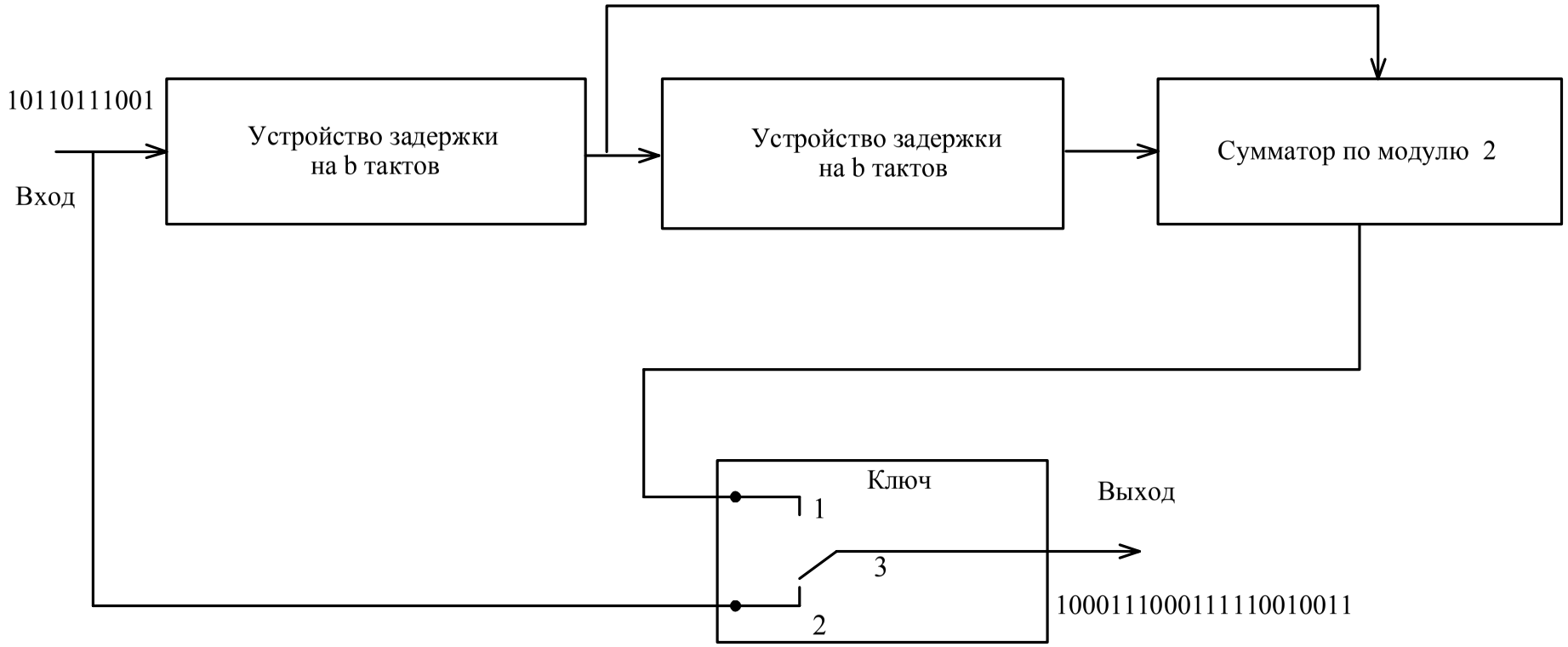


Рис. 12.5 . Структурная схема кодера рекуррентного кода

Эта последовательность поступает в дискретный канал связи.

Структурная схема декодера приведена на рис. 12.6.

Процесс декодирования заключается в выработке контрольных символов из информационных, поступивших на декодер, и их сравнении с контрольными символами, пришедшими из канала связи. В результате сравнения вырабатывается корректирующая последовательность, которая и производит исправление информационной последовательности.

Рассмотрим этот процесс более подробно. Пусть из дискретного канала связи на вход подается искаженная помехами последовательность (искаженные символы обозначены сверху чертой)

$$1\ 0\ 0\ 0\ 1\ \overline{0}\ 1\ \overline{1}\ \overline{1}\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1. \qquad (12.21)$$

Устройство разделения (рис. 12.4) разделяет последовательность (12.21) на информационные последовательности:

Дописанные 2b нулей

$$\overbrace{0\ 0\ 0\ 0}\ 1\ 0\ 1\ 1\ \overline{1}\ 1\ 1\ 1\ 0\ 0\ 1 \qquad (12.22)$$

и контрольные символы:

$$0\ 0\ \overline{0}\ \overline{1}\ 0\ 1\ 1\ 0\ 1\ 0\ 1. \qquad (12.23)$$

Последовательности символов (12.22) и (12.23) содержат ошибочные символы, которые подчеркнуты сверху. Формирователь контрольных символов из (10.22), по тем же правилам, что и при кодировании входной комбинации, выдает последовательность символов

$$0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1, \qquad (12.24)$$

которая в сумме по модулю два с последовательностью (10.23) дает исправляющую последовательность

$$0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0. \qquad (12.25)$$

Исправляющая последовательность (12.25) подается на инвертор, который выдает последовательность (12.26) и одновременно поступает на устройство задержки на b и $2b$ тактов. На выходе устройств задержки появляются последовательности (12.27) и (12.28) соответственно. На выходе схемы совпадения получаем последовательность (10.29)

$$1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ \ldots \qquad (12.26)$$
$$.\ .\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ \ldots \qquad (12.27)$$
$$.\ .\ .\ .\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ \ldots \qquad (12.28)$$
$$0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ \ldots \qquad (12.29)$$

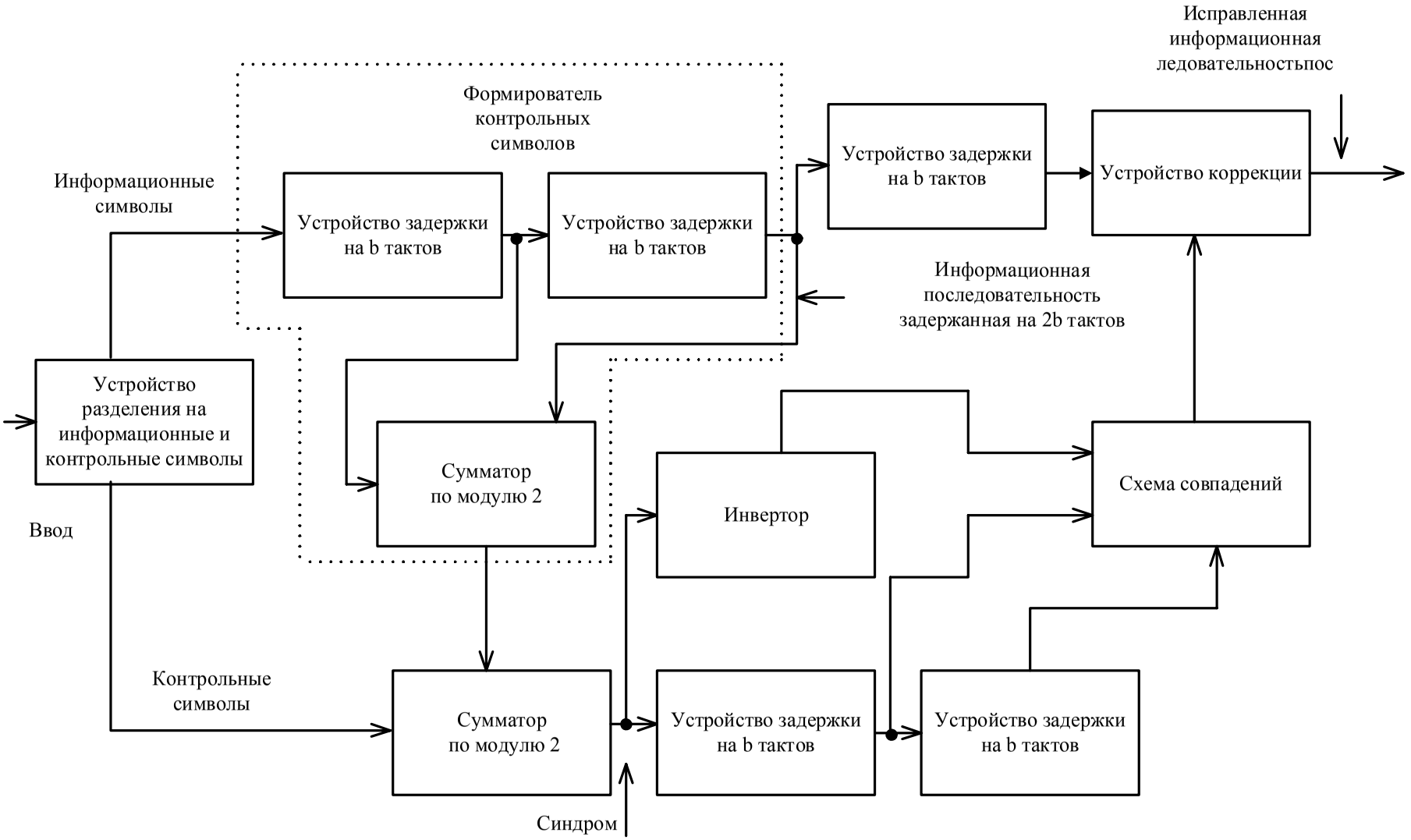


Рис. 12.6 . Структурная схема декодера рекуррентного кода

Точки в последовательностях слева обозначают задержку символов на соответствующее число тактов. Единица на выходе схемы совпадения возникает только в тех случаях, когда на все его три входа подаются единицы. Она представляет собой команду исправить ошибку. Исправленная последовательность вырабатывается устройством коррекции в виде суммы по модулю два последовательности (12.29) и (12.22), задержанной на b тактов:

$$\begin{array}{r} 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\ \underline{\ldots\ldots\ldots\ 1\,0\,1\,1\,1\,1\,1\,1\,1\,0\,0\,1} \\ 1\,0\,1\,1\,0\,1\,1\,1\,0\,0\,1 \end{array}. \qquad (12.30)$$

Точки в последовательности слева означают задержку на шесть тактов относительно входа в устройство разделения на информационные и контрольные символы.

После автоматического исправления последовательность (12.30) совпадает с последовательностью на рис. 12.4 (верхняя строка). Как следует из (12.29), на пути информационных символов включено $3b = 6$ ячеек регистра сдвига. При этом для вывода всех ошибочных символов необходим защитный интервал $6b + 1 = 13$ символов.

Рассмотренный код позволяет исправлять пакет ошибок длиной $l = 2b = 4$.

В заключение следует отметить, что рекуррентный код находит применение в системах связи, для передачи факсимильных сообщений.

12.7. Сверточные коды

Методы кодирования и декодирования, рассмотренные в подразделе 12.5, относились к блочным кодам. При использовании таких кодов информационная последовательность разбивается на отдельные блоки, которые кодируются независимо друг от друга. Таким образом, закодированная последовательность становится последовательностью независимых слов одинаковой длины.

При использовании сверточных кодов поток данных разбивается на гораздо меньшие блоки длиной $\boldsymbol{k}$ символов (в частном случае $\boldsymbol{k_0 = 1}$), которые называются *кадрами информационных символов*.

Кадры информационных символов кодируются *кадрами кодовых символов* длиной $\boldsymbol{n_0}$ символов. При этом кодирование кадра информационных символов в кадр кодового слова производится *с учетом предшествующих* $\boldsymbol{m}$ *кадров информационных символов*. Процедура кодирования, таким образом, связывает между собой последовательные кадры кодовых слов. Передаваемая последовательность становится одним полубесконечным кодовым словом.

Основными характеристиками сверточных кодов являются величины:

- $\boldsymbol{k_0}$ *– размер кадра информационных символов;*
- $\boldsymbol{n_0}$ *– размер кадра кодовых символов;*
- $\boldsymbol{m}$ *– длина памяти кода;*
- $\boldsymbol{k = (m+1) \cdot k_0}$ *– информационная длина слова;*
- $\boldsymbol{n = (m+1) \cdot n_0}$ *– кодовая длина блока.*

Кодовая длина блока – это *длина кодовой последовательности, на которой сохраняется влияние одного кадра информационных символов*.

Наконец, сверточный код имеет еще один важный параметр – скорость ***R = k/n,*** которая характеризует степень избыточности кода, вводимой для обеспечения исправляющих свойств кода.

Как и блочные, сверточные коды могут быть систематическими и несистематическими и обозначаются как линейные сверточные ***(n,k)***–коды.

Систематическим сверточным кодом является такой код, для которого в выходной последовательности кодовых символов содержится без изменения породившая его последовательность информационных символов. В противном случае сверточный код является несистематическим.

Примеры схем кодеров для систематического ***(8,4)*** и несистематического сверточных ***(6,3)***–кодов приведены на рис. 12.7 и 12.8.

ля того, чтобы схема рис. 12.8 стала систематической, надо убрать один сумматор. Корректирующие свойства от того не изменятся, но у несистематического кода свёртка больше – это выгодно и нет информации в открытом виде.

Возможны различные способы описания сверточных кодов, например, с

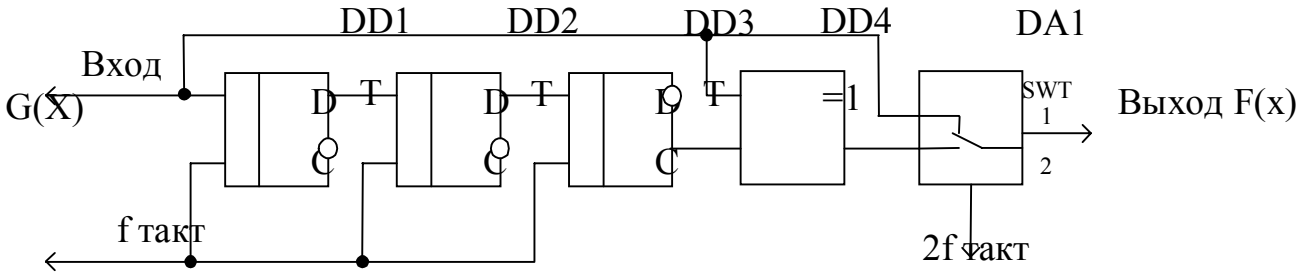


Рис. 10.7. Кодер систематического сверточного кода (8.4)

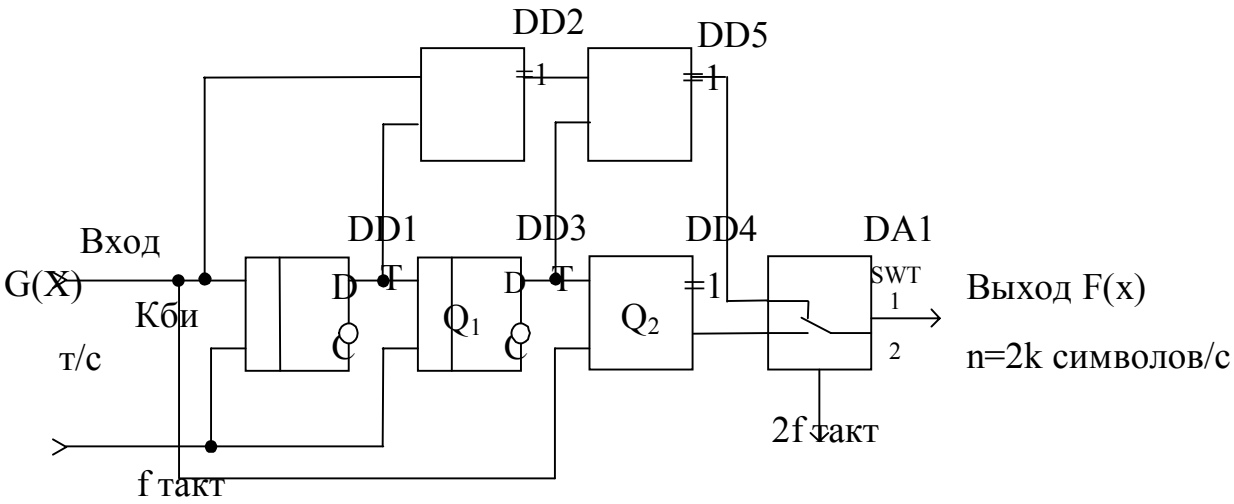


Рис. 12.8. Кодер несистематического сверточного кода (6.3)

помощью порождающей матрицы. Правда, в силу бесконечности кодируемой последовательности и порождающая матрица будет иметь бесконечные разме-

ры. Точнее, она будет состоять из бесконечного числа матриц $\underline{\boldsymbol{M}}$ для обычного блочного кода, расположенных вдоль главной диагонали полубесконечной матрицы. Вся остальная ее часть заполняется нулями.

Однако более удобным способом описания сверточного кода является его задание с помощью *импульсной переходной характеристики* эквивалентного фильтра или соответствующего ей *порождающего полинома*.

Импульсная переходная характеристика фильтра (***ИПХ***) (а кодер сверточного кода, по сути дела, является фильтром) есть реакция на единичное воздействие вида $\overline{\delta}$ = (***10000.....*** Для кодеров, изображенных на рис. 12.7 и 12.8, соответствующие импульсные характеристики будут иметь вид:

$$\boldsymbol{H_1 = (11.00.00.01.00.00\ldots}\ , \qquad (12.31)$$

$$\boldsymbol{H_2 = (11.10.11.00.00.00\ldots}\ . \qquad (12.32)$$

Еще одна форма задания сверточных кодов – это использование порождающих полиномов, однозначно связанных с ***ИПХ*** эквивалентного фильтра:

$$\boldsymbol{H_1(x) = 1 + x + x^7}, \qquad (12.33)$$

$$\boldsymbol{H_2(x) = 1 + x + x^2 + x^4 + x^5}. \qquad (12.34)$$

При этом кодовая последовательность F(x) на выходе сверточного кодера получается в результате свертки входной информационной последовательности G(x) с импульсной переходной характеристикой ***H***.

Рассмотрим примеры кодирования последовательностей с использованием импульсной характеристики эквивалентного фильтра.

Пусть G(x)=(***110*** ... , тогда для кодера с ***ИПХ*** $\boldsymbol{H_1}$ = (***11.00.00.01.00.00....***

11.00.00.01.00.00…
11.00.00.01.00…

F(x) = 11.11.00.01.01.00.00… ,

или G(x) = (***1011000…***

11.00.00.01.00.00.00…
11.00.00.01.00…
11.00.00.01…

F(x) = 11.00.11.10.00.01.01.00… .

Иногда удобнее рассматривать полный порождающий полином сверточного кода P***(x)*** как совокупность нескольких многочленов меньших степеней, соответствующих ячейкам выходного регистра кодера. Так, для кодера рис. 10.8 соответствующие частичные порождающие полиномы будут следующими:

$$P_1(x) = 1 + x + x^2, \qquad (12.35)$$

$$P_2(x) = 1 + x^2. \qquad (12.36)$$

Пусть, например, кодируется последовательность G(x) = (***1010***… .

Тогда на входе 1 ключа DA1 при кодировании будет последовательность $F_2(x)$ = (11011000…, а на входе 2 – $F_2(x)$ = (10001000… .

Легко заметить, что при этом справедливы равенства

$$\boldsymbol{F_1(x) = G(x) \cdot P_1(x),} \qquad (12.37)$$

$$\boldsymbol{F_2(x) = G(x) \cdot P_2(x).} \qquad (12.38)$$

Такая форма записи удобна, поскольку видна структура кодирующего устройства (по набору полиномов можно сразу синтезировать устройство).

12.7.1. Кодовое дерево и решетчатая диаграмма

Еще одним очень наглядным способом описания и иллюстрации работы сверточных кодов является использование так называемого кодового дерева.

Рассмотрим сверточный (***6.3***)–код со схемой, изображенной на рис. 12.8.

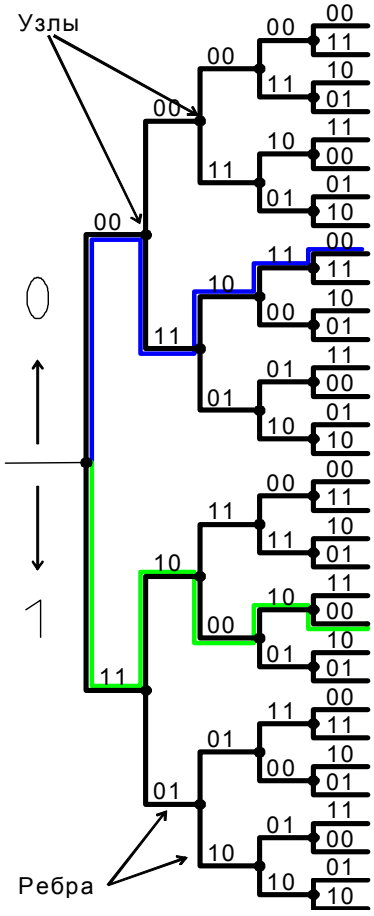


Рис. 12.9

Соответствующее этому кодеру *кодовое дерево* – последовательность выходных состояний кодера при подаче на его вход цепочки входных символов 0 и 1 – приведено на рис. 12.9.

На диаграмме рис. 12.9 изображены входные и выходные последовательности кодера: *входные – направлением движения по дереву (вверх – **0**, вниз – **1**), выходные – символами вдоль ребер дерева.* При этом кодирование состоит в движении вправо по кодовому дереву.

Например, входная последовательность ***G(x)*** = (***01000***…… кодируется как ***F = (0011101100000… ,*** последовательность ***G(x) = (1010100000***… – как F***(x) = (1110001000…*** .

Если внимательно посмотреть на строение приведенного на рис. 12.9 кодового дерева, можно заметить, что начиная с четвертого ребра его структура повторяется, т.е. каким бы ни был первый шаг, начиная с четвертого ребра значения выходных символов кодера повторяются. Одинаковые же узлы могут быть объединены, и тогда начиная с некоторого сечения размер кодового дерева перестанет увеличиваться.

Другими словами, выходная кодовая последовательность в определенный момент перестает зависеть от значений входных символов, введенных в кодер ранее.

Действительно, из рис. 12.9 видно, что, когда третий входной символ вводится в кодер, первый входной символ покидает сдвиговый регистр и не сможет в дальнейшем оказывать влияния на выходные символы кодера.

С учетом этого неограниченное кодовое дерево, изображенное на рис. 10.9,

переходит в ограниченную решетчатую диаграмму (кодовое дерево со сливающимися узлами) рис. 12.10.

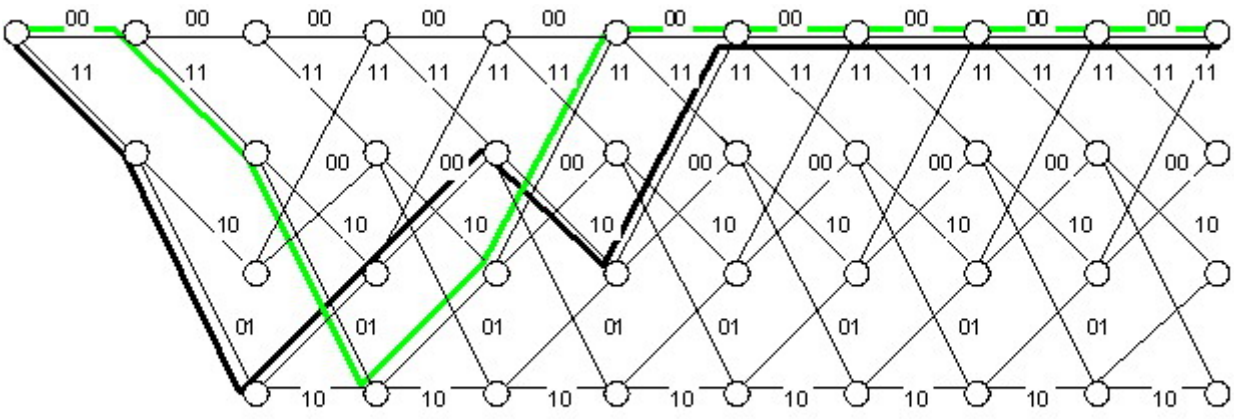


Рис. 12.10

Кодирование сверточными кодами с использованием решетчатой диаграммы, как и в случае с кодовым деревом, представляет собой чрезвычайно простую процедуру: *очередные символы входной последовательности определяют направление движения из узлов решетки: если 0, то идем по верхнему ребру, если 1 – по нижнему ребру*. Однако в отличие от кодового дерева решетчатая диаграмма не разрастается по ширине с каждым шагом, а имеет начиная с третьего сечения постоянную ширину.

В качестве примера закодируем с помощью решетчатой диаграммы несколько информационных последовательностей.

Пусть ***G(x)***=(***0110000… .*** Соответствующая ей кодовая последовательность будет иметь вид:

F(x) = *(001101011100… ,*

или ***G(x)*** = (***110100***… , тогда

F(x) = *(1101010010110000….*

Рассмотренный механизм кодирования входной кодовой последовательности в свёрточном коде достаточно просто реализуется кодером Трелисса.

12.7.2. Треллис-кодирование

Рассмотрим принципы треллис–кодирования на основе простейшего кодера, состоящего из двух запоминающих ячеек и сумматоров по модулю два (рис.12.8). Пусть на вход такого кодера поступает со скоростью k бит/с последовательность бит 0101110010. Если на выходе кодера установить ключ DA1, работающий с вдвое большей частотой, чем скорость поступления бит на вход кодера, то скорость выходного потока будет в два раза выше скорости входного потока. При этом считывающая ячейка DA1 за первую половину такта работы кодера считывает данные сначала с логического элемента DD5, а вторую поло-

вину такта – с логического элемента DD4. В результате каждому входному биту ставится в соответствие два выходных бита, то есть дибит, первый бит которого формируется элементом DD5, а второй – элементом DD4. По временной диаграмме состояния кодера нетрудно проследить, что при входной последовательности бит 0101110010 выходная последовательность будет 00 11 10 00 01 10 01 11 11 10 (рис. 12.11).

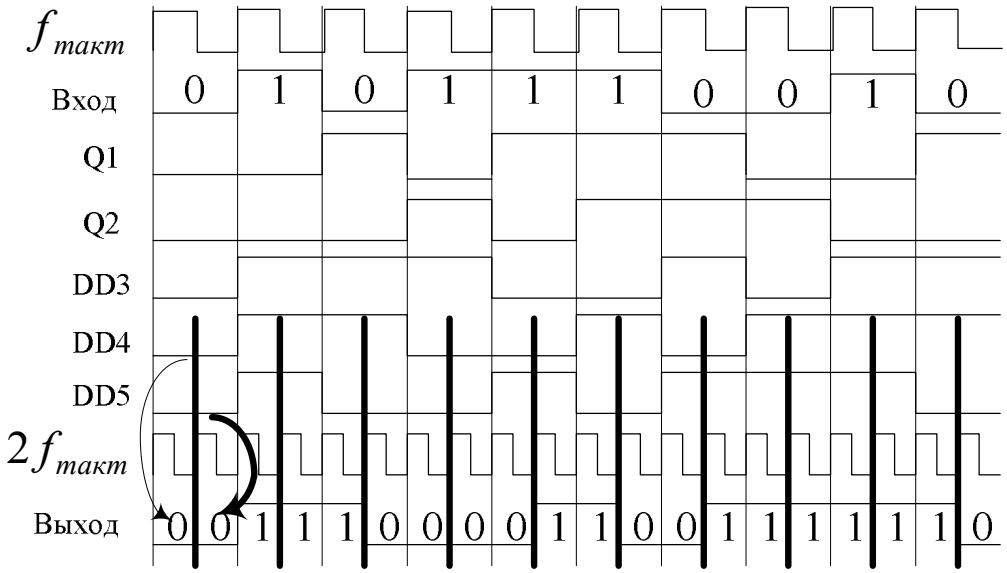


Рис. 12.11. Временные диаграммы работы простейшего кодера Треллиса

Отметим одну важную особенность принципа формирования дибитов. Значение каждого формируемого дибита зависит не только от входящего информационного бита, но и от двух предыдущих бит, значения которых хранятся в двух запоминающих ячейках. Действительно, если принято, что A_i – входящий бит, то значение элемента DD5 определится выражением $A_i \oplus A_{i-1} \oplus A_{i-2}$, а значение элемента DD4 – выражением $A_i \oplus A_{i-2}$. Таким образом, дибит формируется из пары битов, значение первого из которых равно $A_i \oplus A_{i-1} \oplus A_{i-2}$, а второго – $A_i \oplus A_{i-2}$. Следовательно, значение дибита зависит от трех состояний: значения входного бита, значения первой запоминающей ячейки и значения второй запоминающей ячейки. Такие кодеры получили название сверточных кодеров на три состояния (K = 3) с выходной скоростью ½.

Работу кодера удобно рассматривать на основе не временных диаграмм, а так называемой диаграммы состояния. Состояние кодера будем указывать с помощью двух значений – значения первой и второй запоминающих ячеек DD1 и DD3. К примеру, если первая ячейка хранит значение 1 (Q1=1), а вторая – 0 (Q2=0), то состояние кодера описывается значением 10. Всего возможно четыре различных состояния кодера: 00, 01, 10 и 11.

Пусть в некоторый момент времени состояние кодера равно 00. Нас интересует, каким станет состояние кодера в следующий момент времени и какой дибит

будет при этом сформирован. Возможны два исхода в зависимости от того, какой бит поступит на вход кодера. Если на вход кодера поступит 0, то следующее состояние кодера также будет 00, если же поступит 1, то следующее состояние (то есть после сдвига) будет 10. Значение формируемых при этом дибитов рассчитывается по формулам $A_i \oplus A_{i-1} \oplus A_{i-2}$ и $A_i \oplus A_{i-2}$. Если на вход кодера поступает 0, то будет сформирован дибит 00 ($0 \oplus 0 \oplus 0 = 0, 0 \oplus 0 = 0$), если же на вход поступает 1, то формируется дибит 11 ($1 \oplus 0 \oplus 0 = 1, 1 \oplus 0 = 0$). Приведенные рассуждения удобно представить наглядно с помощью диаграммы состояний (рис. 10.12), где в кружках обозначаются состояния кодера, а входящий бит и формируемый дибит пишутся через косую черту. Например, если входящий бит 1, а формируемый дибит 11, то записываем: 1/11.

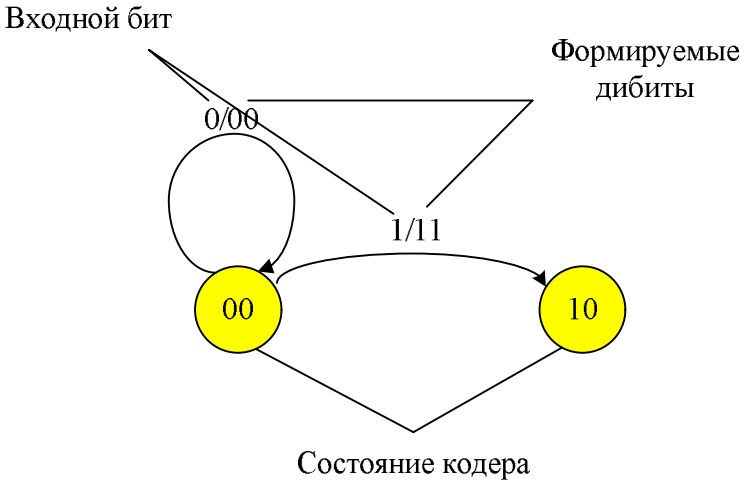


Рис. 12.12. Диаграмма возможных переходов кодера из начального состояния 00

Продолжая аналогичные рассуждения для всех остальных возможных состояний кодера, легко построить полную диаграмму состояний, на основе которой легко вычисляется значение формируемого кодером дибита.

Используя диаграмму состояний кодера, несложно построить временную диаграмму переходов для уже рассмотренной нами входной последовательности бит 0101110010. Для этого строится таблица, в столбцах которой отмечаются возможные состояния кодера, а в строках – моменты времени. Возможные переходы между различными состояниями кодера отображаются стрелками (на основе полной диаграммы состояний кодера – рис. 12.13), над которыми обозначаются входной бит, соответствующий данному переходу, и соответствующий дибит. Например, для первого момента времени диаграмма состояния кодера выглядит так, как показано на рис. 12.14. Жирной стрелкой отображен переход, соответствующий рассматриваемой последовательности бит.

Продолжая отображать возможные и реальные переходы между различными состояниями кодера, соответствующие различным моментам времени (рис. 12.15), получим полную временную диаграмму состояний кодера (рис. 12.16).

Основным достоинством изложенного выше метода треллис–кодирования является его помехоустойчивость. Как будет показано в дальнейшем, благодаря избыточности кодирования (вспомним, что каждому информационному биту ставится в соответствие дибит, то есть избыточность кода равна 2) даже в случае возникновения ошибок приема (к примеру, вместо дибита 11 ошибочно принят дибит 10) исходная последовательность бит может быть безошибочно восстановлена.

Для восстановления исходной последовательности бит на стороне приемника используется декодер Витерби.

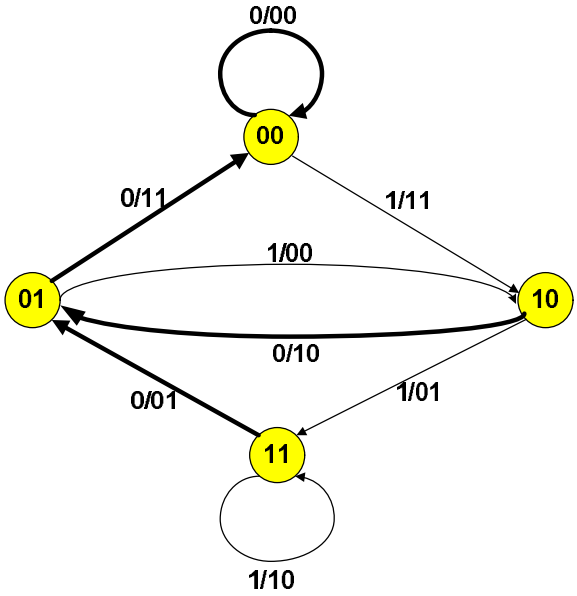


Рис. 12.13. Полная диаграмма состояния кодера

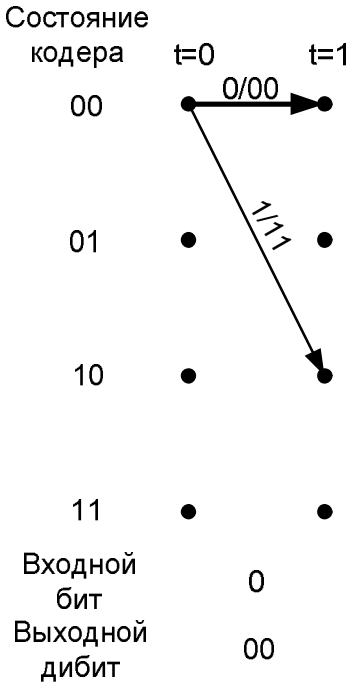


Рис. 12.14. Временная диаграмма состояния кодера для первого момента времени

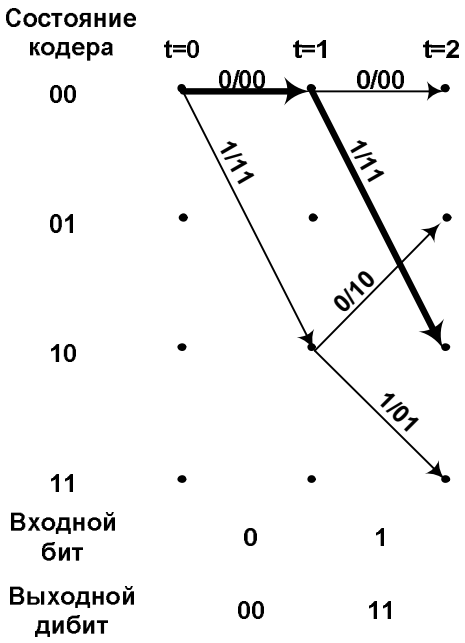


Рис. 12.15. Временная диаграмма состояния кодера для двух первых моментов времени

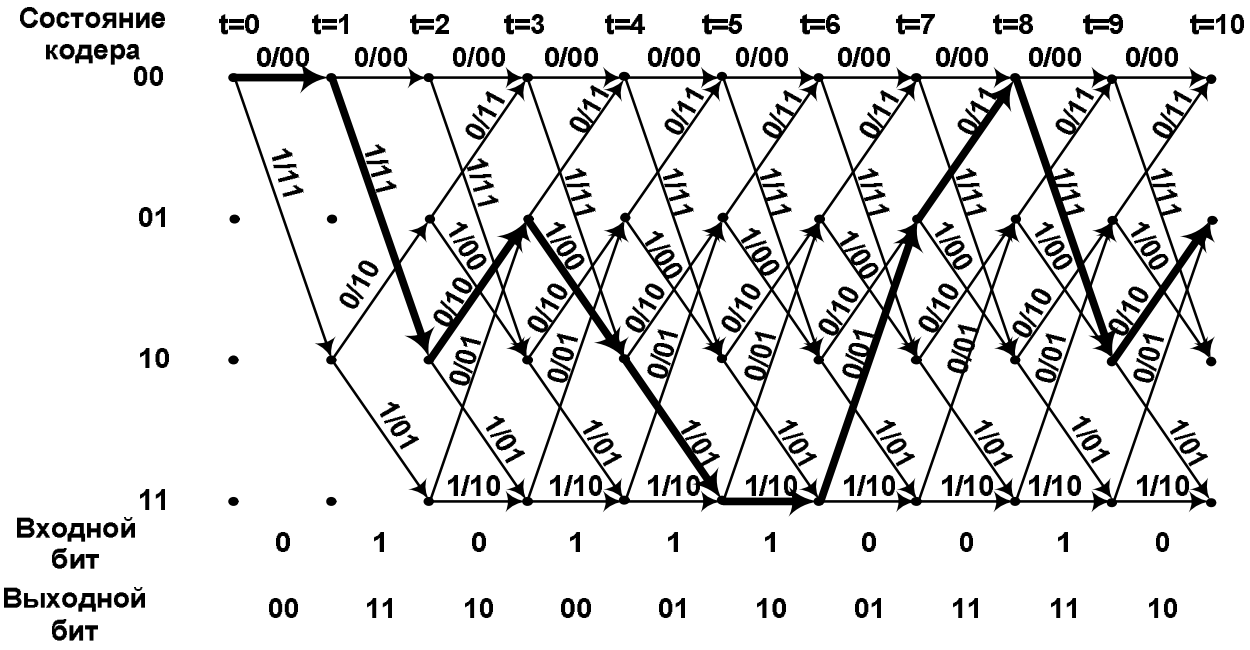


Рис. 12.16. Временная диаграмма состояния кодера
для рассматриваемой входной последовательности бит

12.7.3. Декодер Витерби

Декодер Витерби в случае безошибочного приема всей последовательности дибитов 00 11 10 00 01 10 01 11 11 10 будет обладать информацией об этой последовательности, а также о строении кодера (то есть о его диаграмме состояний) и о его начальном состоянии (00). Исходя из этой информации он должен восстановить исходную последовательность бит. Рассмотрим, каким образом происходит восстановление исходной информации.

Зная начальное состояние кодера (00), а также возможные изменения этого состояния (00 и 10), построим временную диаграмму для первого момента времени (рис. 12.17). На этой диаграмме из состояния 00 существует только два возможных пути, соответствующих различным входным дибитам. Поскольку входным дибитом декодера является 00, то, пользуясь диаграммой состояний кодера Треллиса, устанавливаем, что следующим состоянием кодера будет 00, что соответствует исходному биту 0.

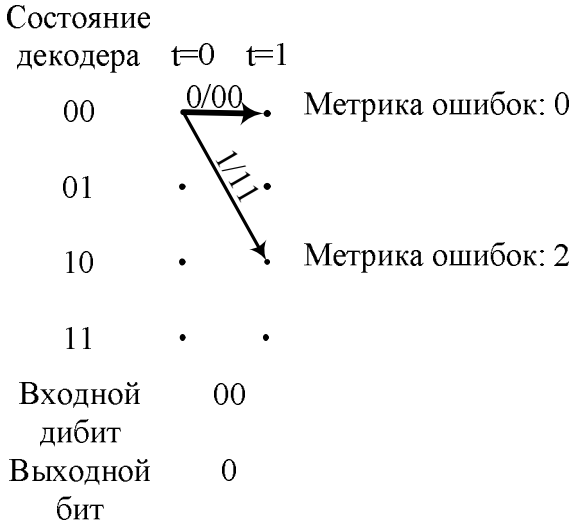


Рис. 12.17. Временная диаграмма возможных состояний декодера для первого момента времени

Однако у нас нет 100% гарантии того, что принятый дибит 00 является правильным, поэтому не стоит пока отметать и второй возможный путь из состояния 00 в состояние 10, соответствующий дибиту 11 и исходному биту 1. Два пути, показанные на диаграмме, отличаются друг от друга так называемой метрикой ошибок, которая для каждого пути рассчитывается следующим образом. Для перехода, соответствующего принятому дибиту (то есть для перехода, который считается верным), метрика ошибок принимается равной нулю, а для остальных переходов она рассчитывается по количеству отличающихся битов в принятом дибите и дибите, отвечающем рассматриваемому переходу. Например, если принятый дибит 00, а дибит, отвечающий рассматриваемому переходу, равен 11, то метрика ошибок для этого перехода равна 2.

Для следующего момента времени, соответствующего принятому дибиту 11, возможными будут два начальных состояния кодера: 00 и 10, а конечных состояния будет четыре: 00, 01, 10 и 11 (рис. 10.18). Соответственно для этих конечных состояний существует несколько возможных путей, отличающихся друг от друга метрикой ошибок. При расчете метрики ошибок необходимо учитывать метрику предыдущего состояния, то есть если для предыдущего момента времени метрика для состояния 10 была равной 2, то при переходе из этого состояния в состояние 01 метрика ошибок нового состояния (метрика всего пути) станет равной 2 + 1 = 3.

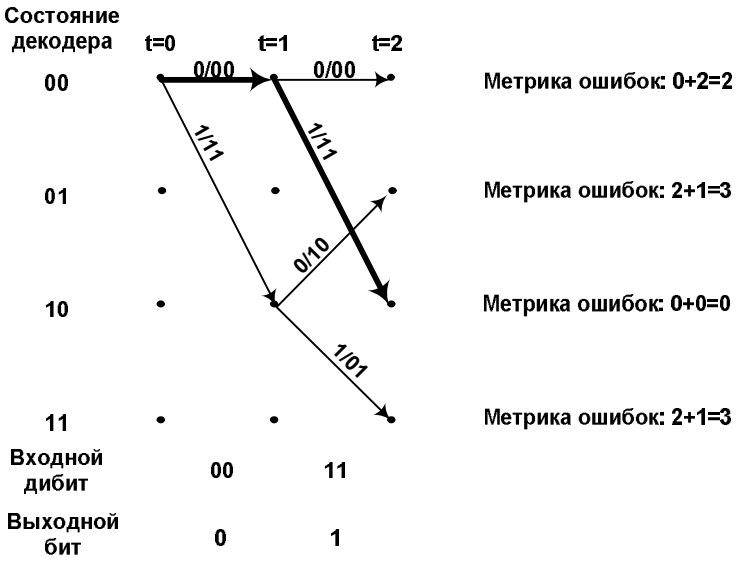


Рис. 12.18. Временная диаграмма возможных состояний декодера для первых двух моментов времени

Для следующего момента времени, соответствующего принятому дибиту 10, отметим, что в состояния 00, 01 и 11 ведут по два пути (рис. 12.19). В этом случае необходимо оставить только те переходы, которым отвечает меньшая метрика ошибок. Кроме того, поскольку переходы из состояния 11 в состояние 11 и в состояние 01 отбрасываются, переход из состояния 10 в состояние 11, отвечающий предыдущему моменту времени, не имеет продолжения, поэтому тоже может быть отброшен. Аналогично отбрасывается переход, отвечающий предыдущему моменту времени из состояния 00 в 00.

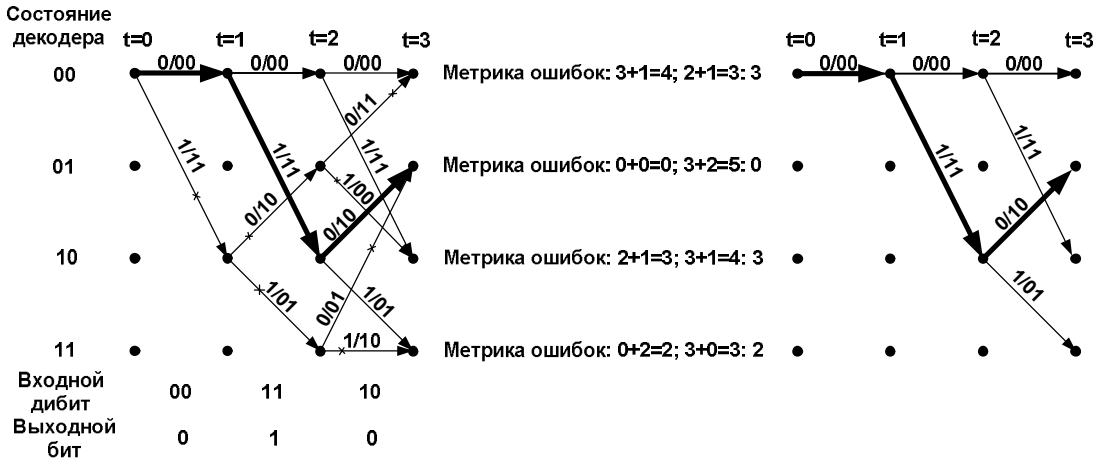


Рис. 12.19. Временная диаграмма возможных состояний декодера для первых трех моментов времени

Продолжая подобные рассуждения, можно вычислить метрику всех возможных путей и изобразить все возможные пути.

При этом количество самих возможных путей оказывается не так велико, как может показаться, поскольку большинство из них отбрасываются в процессе построения, как не имеющие продолжения (рис. 10.20). К примеру, на шестом такте работы декодера по описанному алгоритму остается всего четыре возможных пути.

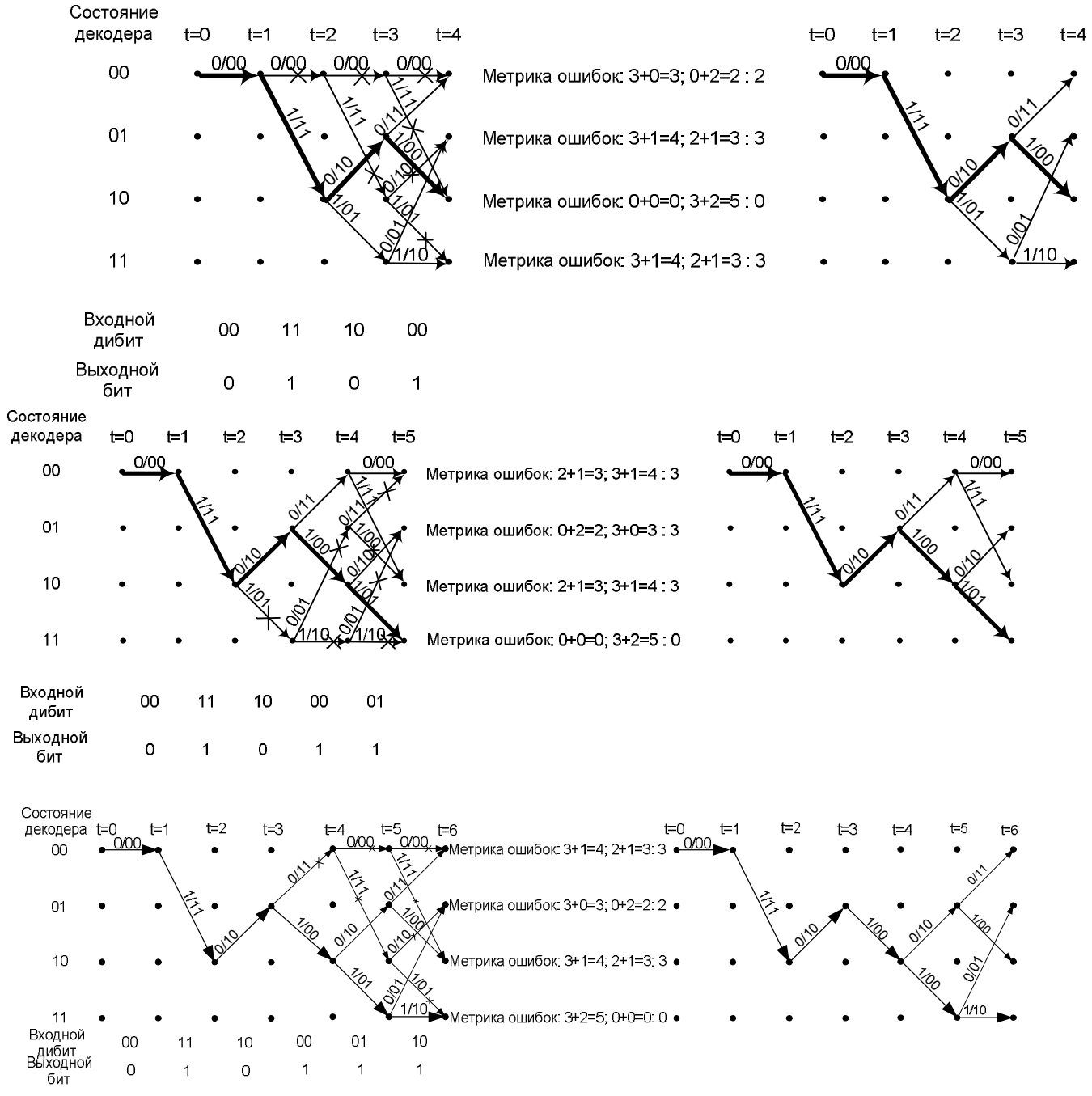


Рис. 12.20. Временная диаграмма возможных состояний декодера для четвертого, пятого и шестого тактов

Аналогично и на последнем такте работы декодера имеется всего четыре возможных пути (рис. 12.21), причем истинный путь, однозначно восстанавли-

вающий исходную последовательность битов 0101110010, соответствует метрике ошибок, равной 0.

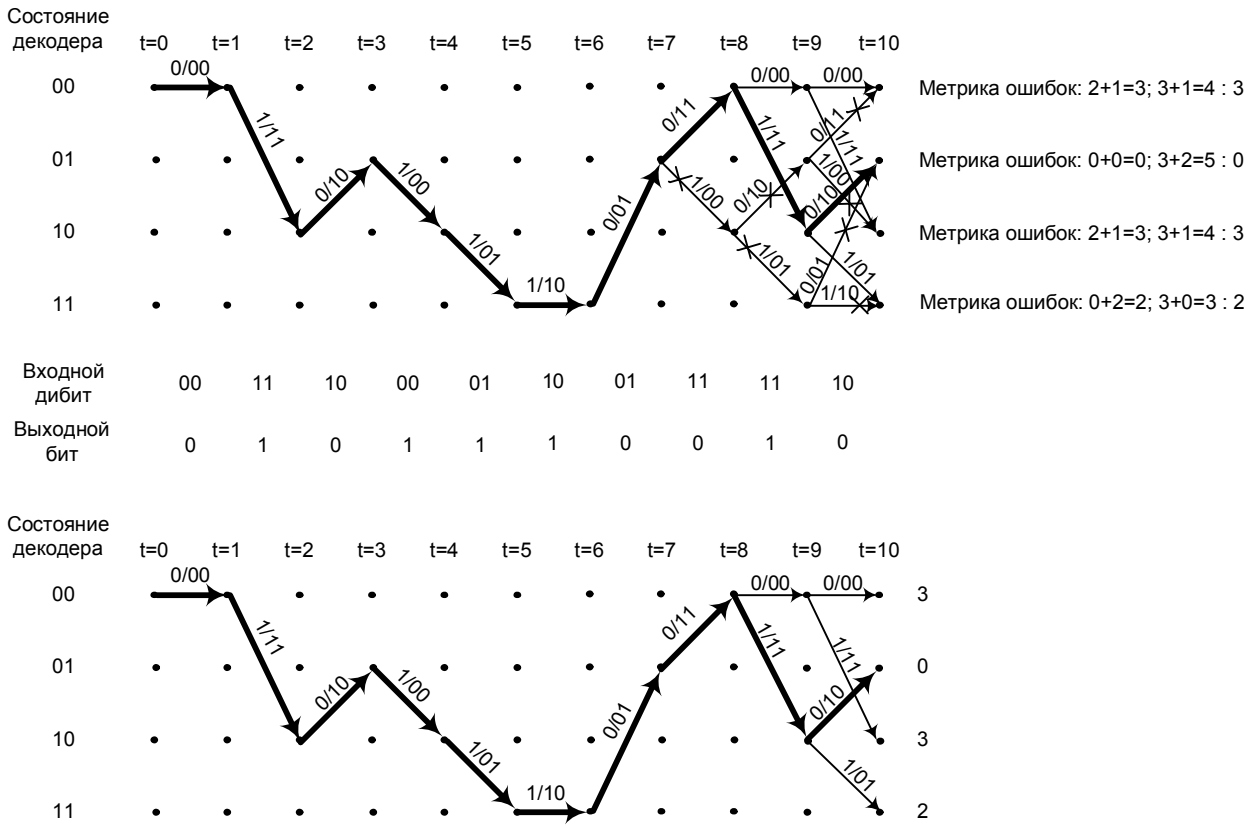


Рис. 12.21. Временная диаграмма возможных состояний декодера для последнего момента времени

При построении рассмотренных временных диаграмм удобно отображать метрику накопленных ошибок для различных состояний кодера в виде таблицы. Именно эта таблица и является источником той информации, на основе которой возможно восстановить исходную последовательность бит (табл. 12.1).

Таблица 12.1.

Метрика ошибок для различных состояний декодера

Состояния декодера	T=0	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10
00	–	0	2	3	2	3	3	3	0	2	3
01	–	–	3	0	3	2	2	0	3	3	0
10	–	2	0	3	0	3	3	3	2	0	3
11	–	–	3	2	3	0	0	2	3	3	2
Входной дибит		00	11	10	00	01	10	01	11	11	10
Выходной бит		0	1	0	1	1	1	0	0	1	0

В описанном выше случае мы предполагали, что все принятые декодером дибиты не содержат ошибок. Рассмотрим далее ситуацию, когда в принятой последовательности дибитов содержатся две ошибки. Пусть вместо правильной последовательности 00 11 10 00 01 10 01 11 11 10 декодер принимает последовательность 00 11 11 00 11 10 01 11 11 10, в которой третий и пятый дебит являются сбойными. Попробуем применить рассмотренный выше алгоритм Витерби, основанный на выборе пути с наименьшей метрикой ошибок, к данной последовательности и выясним, сможем ли мы восстановить в правильном виде исходную последовательность битов, то есть исправить сбойные ошибки.

Вплоть до получения третьего (сбойного) дибита алгоритм вычисления метрики ошибок для всех возможных переходов не отличается от рассмотренного ранее случая. До этого момента наименьшей метрикой накопленных ошибок обладал путь, отмеченный на рис. 10.22 полужирной линией. После получения такого дибита уже не существует пути с метрикой накопленных ошибок, равной 0. Однако при этом возникнут два альтернативных пути с метрикой, равной 1. Поэтому выяснить на данном этапе, какой бит исходной последовательности соответствует полученному дибиту, невозможно.

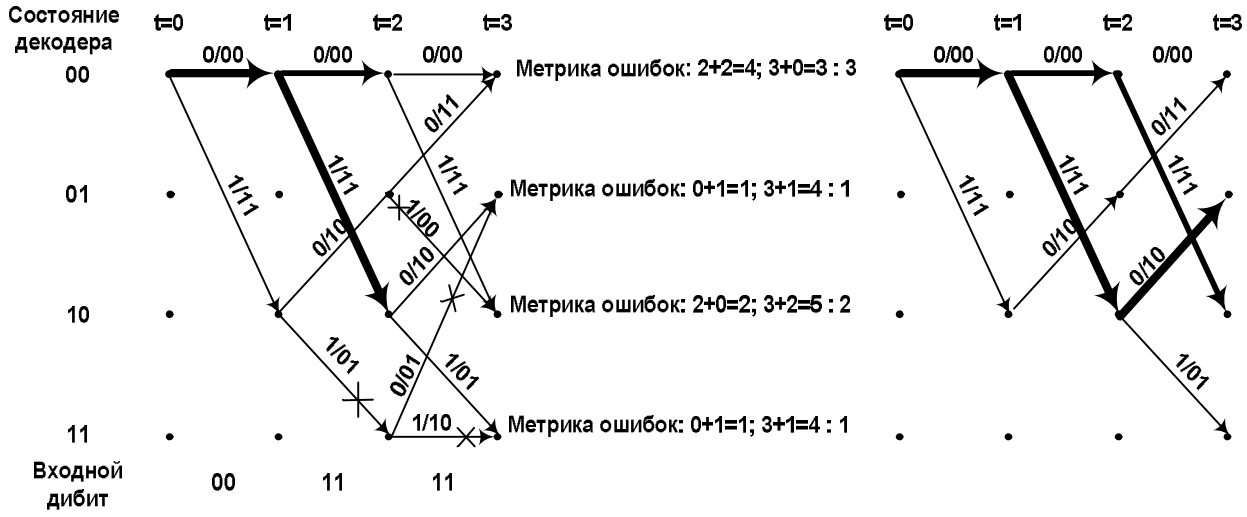


Рис. 12.22. Вычисление метрики накопленных ошибок при получении первого ошибочного дибита

Аналогичная ситуация возникнет и при получении пятого (также сбойного) дибита (рис. 10.23). В этом случае будет существовать уже три пути с равной метрикой накопленных ошибок, а установить истинный путь возможно только при получении следующих дибитов.

После получения десятого дибита количество возможных путей с различной метрикой накопленных ошибок станет достаточно большим (рис. 12.24), однако на приведенной диаграмме (с использованием табл. 12.2, где представлена метрика накопленных ошибок для различных путей) нетрудно выбрать единственный путь с наименьшей метрикой (на рис. 12.24 этот путь отмечен жирной линией). По данному пути, пользуясь диаграммой состояния треллис–кодера (см. рис. 12.13), можно однозначно восстановить исходную последова-

тельность бит 0101110010, невзирая на допущенные ошибки при получении дибитов.

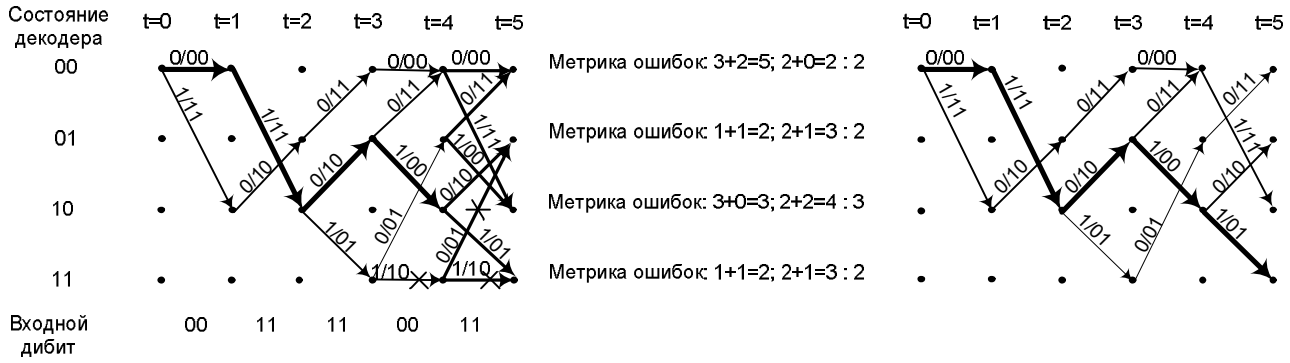


Рис. 12.23. Вычисление метрики ошибок при получении второго ошибочного дибита

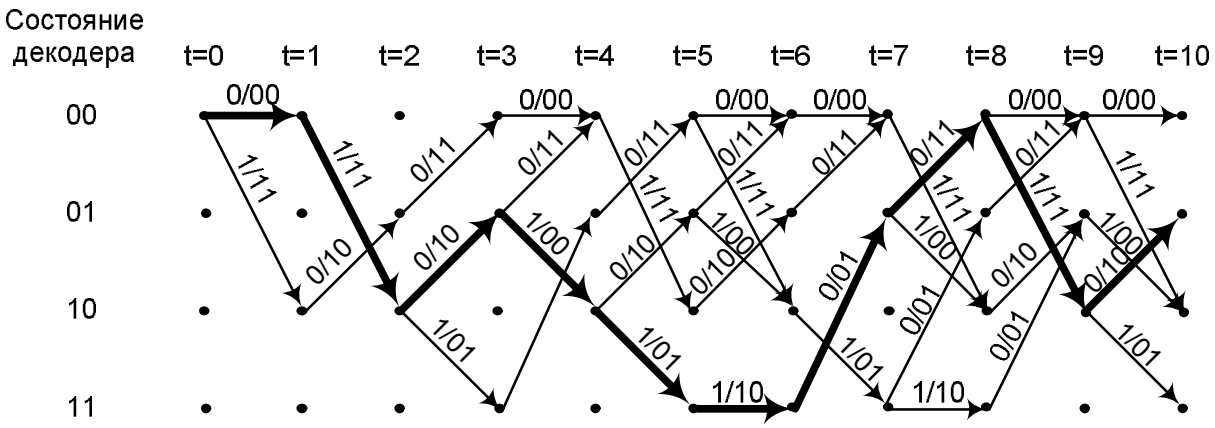


Рис. 12.24. Вычисление метрики ошибок при получении последнего дибита

Рассмотренный сверточный кодер Треллиса на четыре состояния и алгоритм Витерби являются простейшими примерами, иллюстрирующими основной принцип работы. В реальности используемые кодеры Треллиса (и в гигабитных адаптерах, и в модемах) гораздо более сложные, но именно благодаря их избыточности удается значительно повысить помехоустойчивость протокола передачи данных.

Таблица 12.2.

Метрика накопленных ошибок для различных путей

Состояния декодера	t=0	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10
00	–	0	2	3	3	2	3	4	2	4	5
01	–	–	3	1	2	2	3	2	4	5	2
10	–	2	0	2	1	3	3	4	4	2	5
11	–	–	3	1	2	2	2	3	4	5	4
Полученный дибит		00	11	11	00	11	10	01	11	11	10
Выходной бит		0	1	0 1*	1	0* 1	1	0	0	1	0

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Сформулируйте теорему Шеннона о кодировании в каналах связи с помехами.
2. Перечислите основные задачи, решаемые кодированием.
3. При каких предположениях решают задачи корректирующего кодирования?
4. Приведите классификацию корректирующих кодов.
5. Дайте определение блочным кодам.
6. Назовите основные характеристики корректирующих кодов.
7. Покажите принцип обнаружения ошибок на геометрической модели кодов.
8. Приведите выражение для определения числа контрольных символов.
9. Какие коды называются систематическими?
10. Как строится образующая и проверочная матрицы систематического кода?
11. Как получается алгоритм кодирования и декодирования в систематическом коде?
12. Какой принцип кодирования кодовых комбинаций в рекуррентном коде?
13. Какое минимальное расстояние должно быть между пакетами искажений, чтобы принятая кодовая комбинация была однозначно декодируемой?
14. Поясните принцип работы кодера сверхточного кода (8,4).
15. Приведите и поясните диаграмму переходов кодера (6,3).
16. В чём сущность декодирования кодовых сообщений по методу Витерби.

ЗАКЛЮЧЕНИЕ

Изложенные в конспекте лекций идеи и методы теории информации представляют интерес не только в плане решения задач, связанных с передачей и хранением информации. Теоретико–информационный подход приобрел значение метода исследования, позволяющего качественно и количественно сопоставлять специфические характеристики конкретных устройств и систем независимо от их физической сущности.

Трудами ученых и инженеров созданы и продолжают создаваться фундаментальные методы анализа, синтеза и оптимизации информационных систем. Развивается математическое моделирование процессов передачи сообщений, разрабатываются и внедряются новые методы анализа нестационарных непрерывных каналов, неоднородных асимметричных дискретных каналов, помехоустойчивые методы и алгоритмы передачи информации, в которых учитывается ненадежность аппаратуры и отклонение характеристик устройств от идеальных, строятся математические модели сложных шумовых ситуаций, интенсивно развивается корректирующее кодирование.

Методы теории информации все шире применяются для решения практических задач повышения качества передачи информации и эффективности систем и сетей связи, радионавигационных и радиолокационных систем, автоматизированных систем управления воздушным движением, вычислительных систем, измерительных комплексов и многих других.

Основные тенденции и перспективы теории информации следующие:

широкое внедрение цифровых методов передачи сообщений;

рост удельного веса алгоритмических и программных методов управления процессами передачи информации;

широкое использование корректирующего кодирования;

разработка методов оценки эффективности передачи информации с позиции системного подхода;

применение цифрового и статистического моделирования процессов передачи сообщения для анализа и синтеза информационных систем;

Глубокое и всестороннее развитие теории информации тесно связано с практическими потребностями информационной техники.

Следует ожидать, что идеи и методы теории информации будут успешно использоваться при создании сложных систем, объединяющих различные по целям, функциям и даже физическому воплощению подсистемы.

ЛИТЕРАТУРА

1. Дмитриев В. И. Прикладная теория информации.–М.:Высш.шк.,1989.–320 с.
2. Темников Ф.Е. и др. Теоретические основы информационной техники.–М.: Энергия, 1979.–512 с.
3. Герасименко В.А., Мясников В.А. Защита информации от несанкционированного доступа.–М.: МЭИ, 1984.
4. Шеннон К. Работа по теории информации и кибернетике.–М.: ИЛ, 1963.
5. Питерсон У.,Уэлдон Э. Коды, исправляющие ошибки.–М.: Мир, 1976.
6. Игнатов В.А. Теория информации и передачи сигналов: Учебник для вузов.–М.:Сов.радио,1979.–280с.
7. Пенин П.И., Филиппов Л.И. Радиотехнические системы передачи информации: Учеб. пособие для вузов.–М.:Радио и связь, 1984.–256 с.
8. Фельдбаум А.А. и др. Теоретические основы связи и управления.–М.: Физматгиз, 1963.–932 с.
9. Шувалов В.П. и др. Передача дискретных сообщений: Учебник для вузов.–М.: Радио и связь, 1990.–464 с.
10. Бовбель Е.И. и др. Элементы теории информации.–Мн.: БГУ, 1974.–111 с.
11. Ильин В.А. Телеуправление и телеизмерение: Учеб. пособие для вузов.–3–е изд.–М.: Энергоиздат, 1982.–560 с.
12. Брюс Шнайер. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си.–М.: Изд–во ТРИУМФ, 2002.–816 с.
13. Столлингс Вильям. Криптография и защита сетей.–М., 2002.
14. Молдовен и др. Криптография. Скоростные шифры. –М., 2003.
15. Маслянников М. Практическая криптография.–М., 2003.
16. Петраков А.В., Лагутин В.С. Телеохрана.–М.: Энергоатомиздат, 1998.–376 с.
17. Основы теории передачи информации. ч.1. Экономное кодирование. /В.И.Шульгин. – Учебн.пособ. – Харьков: Нац. аэрокосм. ун–т «Харьк. Авиац. Ин–т.», 2003–102с.
18. Основы теории передачи информации. ч.2. Экономное кодирование. /В.И.Шульгин. – Учебн.пособ. – Харьков: Нац. аэрокосм. ун–т «Харьк. Авиац. Ин–т.», 2003–87с.
19. Лидовский В.В, Теория информации: Учебное пособие. – М.: Компания Спутник +, 2004.–111с.