Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кузбасский государственный технический университет
имени Т.Ф. Горбачева»

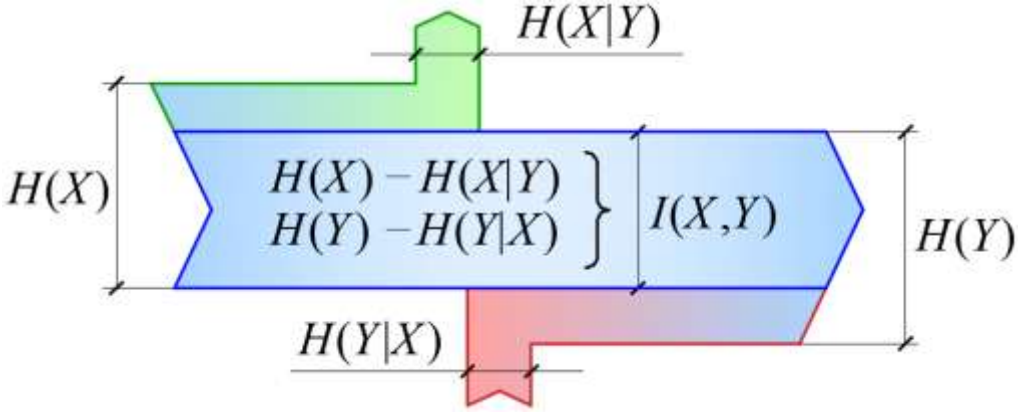
Кафедра информационных и автоматизированных производственных систем

Составитель
Д. Е. Турчин

ТЕОРИЯ ИНФОРМАЦИИ

**Методические указания к самостоятельной работе
для студентов очной формы обучения**

Рекомендовано учебно-методической комиссией направления
подготовки бакалавра 09.03.02 «Информационные системы
и технологии» в качестве электронного издания
для использования в учебном процессе



Кемерово 2016

Рецензенты:
Чичерин Иван Владимирович – кандидат технических наук, доцент, председатель учебно-методической комиссии направления 09.03.02 «Информационные системы и технологии»

Турчин Денис Евгеньевич
Теория информации: методические указания к самостоятельной работе [Электронный ресурс]: для студентов направления подготовки 09.03.02 «Информационные системы и технологии» очной формы обучения / сост. Д. Е. Турчин; КузГТУ. – Электрон.дан. – Кемерово, 2016. – Систем. требования: Pentium IV; ОЗУ 256 Мб; WindowsXP; мышь. – Загл. с экрана.

Изложено содержание самостоятельных практических работ, порядок их выполнения, а также контрольные вопросы к ним.

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Пособие предназначено для студентов второго курса направления подготовки бакалавра 09.03.02 «Информационные системы и технологии», изучающих дисциплину «Теория информации».

Самостоятельная работа по дисциплине включает в себя выполнение четырёх практических работ, а также изучение отдельных тем теоретического материала.

Первая практическая работа связана с определением количества информации в дискретных сообщениях. Во второй работе изучаются простейшие методы эффективного кодирования (метод Шеннона-Фано и метод Хафмана). Третья работа посвящена помехоустойчивому кодированию с помощью кодов Хэмминга. В четвертой работе рассматривается шифрование информации методами перестановки и замены.

САМОСТОЯТЕЛЬНЫЕ ПРАКТИЧЕСКИЕ РАБОТЫ

1. ОПРЕДЕЛЕНИЕ КОЛИЧЕСТВА ИНФОРМАЦИИ В ДИСКРЕТНЫХ СООБЩЕНИЯХ

1.1. Цель и задачи работы

Цель работы – приобрести умение определять количество информации в дискретных сообщениях.

Основные задачи работы:

• научиться определять частное и среднее количество информации в дискретном сообщении;

• научиться определять энтропию дискретных сообщений, генерируемых источником без памяти.

Работа рассчитана на 2 часа.

1.2. Основные теоретические сведения

1.2.1. Дискретные источники сообщений без памяти. Частное количество информации. Формула Хартли

Система передачи сообщений. Дискретные источники сообщений без памяти.

В теории информации при оценке количества информации в сообщениях используется вероятностный подход, который основан на использовании понятий и методов теории вероятностей.

Для оценки количества информации в сообщении важно знать модель источника, который сформировал это сообщение.

Источник сообщений является составной частью ***системы передачи сообщений*** (рис. 1.1), которая также может включать в себя кодер и декодер источника сообщений, канал связи, кодер и декодер канала связи, а также получателя сообщений.

Источник сообщений, который может в каждый момент времени случайным образом принять одно из конечного множества возможных состояний, называют ***дискретным источником сообщений***. Каждому состоянию источника соответствует условное обозначение в виде символа.

Множество всех символов $S = \{s_1, s_2, ..., s_m\}$, доступных источнику сообщений, называют ***алфавитом*** этого источника, а общее число символов m – ***объемом алфавита***.

Простейшим видом дискретных источников сообщений являются ***источники без памяти*** (с нулевой памятью), в которых текущее состояние источника не зависит от его предшествующих состояний.

Рис. 1.1. Структура системы передачи сообщений

Дискретный источник сообщений без памяти в общем случае характеризуется ***ансамблем*** (вероятностной схемой) A, то есть полной совокупностью состояний с вероятностями их появления, составляющими в сумме единицу:

$$A = \begin{pmatrix} s_1 & s_2 & \dots & s_m \\ p_1 & p_2 & \dots & p_m \end{pmatrix}; \quad \text{причем} \quad \sum_{i=1}^{m} p_i = 1.$$

Частное количество информации. Формула Хартли.

Важным является вопрос о том, сколько информации поступает при приеме одного из символов алфавита источника. Путем несложных рассуждений можно доказать, что количество информации I, переносимое одним символом, и вероятность p появления этого символа в сообщении связаны следующей зависимостью:

$$I = \log_a(1/p) = -\log_a p; \qquad (1.1)$$

где основание a логарифма $\log_a p$ определяет единицу измерения количества информации:

• $a = 2 \rightarrow$ бит;

• $a = 10 \rightarrow$ дит (хартли);

• $a = e \approx 2{,}72\ldots \rightarrow$ нат.

Количество информации, задаваемое формулой (1.1), называют ***частным*** (собственным) ***количеством информации***.

❒ ***Пример* 1.1. *Определение частного количества информации.***

***Задача* 1.** Бросаются две монеты. Рассматриваются события:

A_1 – выпадение герба на первой монете;

A_2 – выпадение герба на второй монете.

Требуется найти количество информации, получаемое от события, которое заключается в выпадении только одного герба.

По теореме сложения вероятностей получим

$$p(A_1 + A_2) = p(A_1) + p(A_2) - p(A_1A_2) = \frac{1}{2} + \frac{1}{2} - \frac{1}{4} = \frac{3}{4}.$$

По формуле (1.1) частное количество информации будет

$$I = -\log_2 0{,}75 = 0{,}415 \text{ (бит)}.$$

***Задача* 2.** В урне находятся 9 белых шаров (A) и 16 черных шаров (B). Из урны вынимаются (одновременно или последовательно) два шара. Найти количество информации, получаемое от осуществления события, что оба шара являются белыми.

По теореме умножения вероятностей получим

$$p(A_1A_2) = p(A_1)\,p(A_2 \mid A_1) = \frac{9}{(9+16)}\,\frac{8}{(9+16-1)} = 0{,}12.$$

Отсюда по формуле (1.1) частное количество информации будет

$$I = -\log_2 0{,}12 = 3{,}059 \text{ (бит)}. \text{ ❒}$$

Общее число неповторяющихся сообщений, которое может быть составлено из алфавита объемом m путем комбинирования по n символов в сообщении будет

$$N = m^n.$$

При равной вероятности появления символов алфавита в сообщении количество информации в этом сообщении можно найти по ***формуле Хартли***:

$$I = \log_2 N = \log_2 m^n = n \cdot \log_2 m. \quad \text{(бит)} \qquad (1.2)$$

1.2.2. Энтропия дискретного источника сообщений и её свойства. Формула Шеннона

Энтропия дискретного источника сообщений. Формула Шеннона.

Важной характеристикой дискретного источника сообщений является ***энтропия*** H, которая характеризует неопределенность того, что данный источник находится в одном из своих состояний $s_1, s_2, \ldots, s_m$.

Энтропия $H(A)$ дискретного источника сообщений, задаваемого ансамблем A, может быть определена следующим образом:

$$H(A) = -\sum_{i=1}^{m} p(s_i) \log_2 p(s_i), \text{ (бит/символ);} \qquad (1.3)$$

где $p(s_i)$ – вероятность появления символа s_i из алфавита объемом m.

Количество информации в сообщении, состоящем из n символов и выдаваемым источником сообщений с энтропией H, может быть найдено по ***формуле Шеннона***:

$$I = n \cdot H = -n \sum_{i=1}^{m} p_i \log_2 p_i, \text{ (бит).} \qquad (1.4)$$

При равной вероятности появления символов в сообщении, т. е. при $p_i = 1/m$ формула Шеннона переходит в формулу Хартли (1.1).

В случае равной вероятности появления символов на выходе источника сообщений его энтропия будет наибольшей. Соответ-

ственно количество информации передаваемой в сообщении от такого источника также будет наибольшим.

❐ ***Пример* 1.2. *Определение среднего количества информации в сообщении*.**

Требуется определить количество информации в сообщении длиной n = 47 символов, которое было сформировано источником сообщений, описываемым следующим ансамблем:

$$A = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ 0{,}2 & 0{,}3 & 0{,}45 & 0{,}05 \end{pmatrix}.$$

Энтропия источника сообщений по формуле (1.3) будет

$H(A) = -(0{,}2\cdot\log_2 0{,}2 + 0{,}3\cdot\log_2 0{,}3 + 0{,}45\cdot\log_2 0{,}45 +$

$+ 0{,}05\cdot\log_2 0{,}05) = 0{,}464 + 0{,}521 + 0{,}518 + 0{,}216 =$

$= 1{,}719$ (бит/символ).

Отсюда по формуле (1.4) получим следующее количество информации

$$I = 47\cdot 1{,}72 = 80{,}84 \text{ (бит)}. \❐$$

Свойства энтропии дискретного источника сообщений.

Энтропия дискретного источника сообщений обладает следующими свойствами:

• Энтропия является вещественной и неотрицательной величиной, так как вероятности $p(s_i)$ (i = 1, 2, … , m) в формуле (1.3) изменяются в интервале от 0 до 1.

• Энтропия обращается в ноль, если вероятность одного из состояний источника сообщений равна единице. Это соответствует случаю, когда исход опыта может быть предсказан с полной достоверностью, то есть когда отсутствует всякая неопределенность.

• Энтропия дискретного источника сообщений максимальна, когда все состояния источника равновероятны:

$$H_{\max} = \log_2 m.$$

• Энтропия сообщения, состоящего из нескольких частных независимых сообщений, равна сумме энтропий составляющих его сообщений.

$$H(A,B) = H(A) + H(B).$$

❒ ***Пример* 1.3. *Определение условий получения максимального количества информации*.**

Дана некоторая физическая система *S*, которая может находиться в одном из двух состояний: s_1 (обычное) или s_2 (аномальное). В системе *S* протекает циклический процесс, в ходе которого система с небольшой вероятностью p = 0,002 в каждом из циклов может перейти в состояние s_2. Через *k* циклов производится проверка состояния системы. Требуется определить значение *k*, при котором количество информации, получаемое при выяснении состояния системы, будет наибольшим.

Рассматриваемая система может быть описана следующим ансамблем:

$$S = \begin{pmatrix} s_1 & s_2 \\ 1-(1-p)^k & (1-p)^k \end{pmatrix}.$$

Очевидно, что информация, доставляемая выяснением состояния системы *S*, будет максимальна, когда оба состояния s_1 и s_2 равновероятны:

$$1-(1-p)^k = (1-p)^k,$$

приведя подобные в последнем выражении и прологарифмировав его по основанию 2, получим следующую формулу

$$k = \left\lceil \frac{-1}{\log_2(1-p)} \right\rceil;$$

где $\lceil ... \rceil$ – операция округления результата до целого в большую сторону.

Отсюда количество циклов будет

$$k = \left\lceil \frac{-1}{\log_2(1-0{,}002)} \right\rceil = 347. \; ❒$$

1.3. Порядок выполнения работы и варианты заданий

Основные этапы выполнения работы.

Данная практическая работа предполагает выполнение следующих этапов:

1. Изучить методические указания к практической работе.

2. Определить частное количество информации, получаемое при осуществлении заданных событий (табл. 1.1). Для нахождения вероятности события следует использовать теоремы о сложении и умножении вероятностей (см. приложение П.1).

3. Определить количество информации в сообщении, выданным источником сообщений с объемом алфавита m. Принять, что символы алфавита источника сообщений равновероятны. Объем алфавита m источника вычисляется следующим образом:

$$m = |V - 10| + 2;$$

где V – число равное номеру варианта задания.

Число символов n в сообщении необходимо найти по формуле:

$$n = V + 5.$$

4. Определить энтропию дискретного источника сообщений, алфавит которого состоит из восьми независимых символов s_1, s_2, … , s_8. Известны вероятности появления символов p_1, p_2, … , p_8 (табл. 1.2).

5. Используя формулу Шеннона, определить количество информации, содержащейся в сообщении, которое состоит из фамилии, имени и отчества студента (между словами сообщения допускаются пробелы). Вероятности появления букв русского алфавита взять из табл. П.1. Дополнительно необходимо учесть наличие статистической связи между парами и тройками букв.

6. Определить условие, при котором получаемое количество информации будет максимальным (таблица 1.3).

7. Оформить и защитить отчет по практической работе.

Индивидуальные варианты заданий.

Таблица 1.1.

Задачи на определение частного количества информации

№ вар.	Формулировка задачи
	***Задача* 1**. Производится однократное подбрасывание двух игральных костей. Определить количество информации, которое будет получено при наступлении следующих событий:
1, 2, 3	• выпала одна шестерка;
4, 5, 6	• ни на одной кости не было числа кратного трем;
7, 8, 9	• выпало одно четное число;
10, 11, 12	• ни на одной кости не было шести;
13, 14, 15	• выпала одна двойка;
16, 17, 18	• ни на одной кости не было четного числа;
19, 20, 21	• выпало одно число кратное трем;
22, 23, 24	• выпали одинаковые цифры.
	***Задача* 2**. Из полной колоды карт (52 листа, 4 масти) последовательно вынимают четыре карты без возвращения в колоду. Требуется определить количество информации, получаемое при осуществлении следующих событий:
1, 13	• первые две карты красные, остальные две – черные;
2, 14	• все четыре карты разных мастей;
3, 15	• первые две карты тузы, остальные две – шестерки;
4, 16	• все четыре карты с цифрами;
5, 17	• все четыре карты одной масти;
6, 18	• первые две карты одной масти, остальные две – другой масти;
7, 19	• все четыре карты красные;
8, 20	• все четыре карты с разными буквами;
9, 21	• первая карта шестерка, вторая – семерка, третья – девятка, четвертая – десятка;
10, 22	• все четыре карты с буквой;
11, 23	• первые две карты с буквой, остальные две – с цифрой;
12, 24	• все четыре карты – тузы.

Примечание.

Карты с буквами – Туз, Король, Дама, Валет. Карты с цифрами – двойка, тройка, … , десятка.

Таблица 1.2.

Вероятности появления независимых символов s_1, s_2, … , s_8

№ вар.	Вероятности для символов $s_1, s_2, \dots, s_8$							
	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
1	0,21	0,12	0,05	0,08	0,03	0,14	0,2	0,17
2	0,13	0,04	0,19	0,16	0,18	0,12	0,13	0,05
3	0,2	0,15	0,16	0,04	0,19	0,05	0,12	0,09
4	0,14	0,1	0,04	0,17	0,12	0,16	0,05	0,22
5	0,06	0,21	0,03	0,23	0,13	0,12	0,14	0,08
6	0,08	0,19	0,22	0,2	0,09	0,03	0,17	0,02
7	0,09	0,18	0,06	0,12	0,15	0,1	0,07	0,23
8	0,22	0,13	0,12	0,15	0,07	0,04	0,19	0,08
9	0,1	0,24	0,08	0,07	0,17	0,05	0,16	0,13
10	0,04	0,17	0,15	0,11	0,1	0,09	0,13	0,21
11	0,17	0,05	0,16	0,22	0,08	0,19	0,09	0,04
12	0,12	0,09	0,08	0,14	0,16	0,21	0,04	0,16
13	0,23	0,1	0,07	0,06	0,05	0,12	0,22	0,15
14	0,16	0,08	0,05	0,18	0,11	0,15	0,07	0,2
15	0,24	0,12	0,13	0,09	0,14	0,08	0,15	0,05
16	0,11	0,04	0,19	0,16	0,18	0,12	0,13	0,07
17	0,09	0,23	0,05	0,21	0,15	0,1	0,12	0,05
18	0,2	0,15	0,1	0,14	0,08	0,06	0,17	0,1
19	0,15	0,06	0,16	0,23	0,07	0,19	0,08	0,06
20	0,22	0,12	0,06	0,07	0,04	0,13	0,2	0,16
21	0,13	0,08	0,1	0,09	0,21	0,19	0,05	0,15
22	0,05	0,16	0,21	0,11	0,1	0,07	0,1	0,2
23	0,1	0,25	0,09	0,1	0,17	0,06	0,11	0,12
24	0,08	0,2	0,07	0,13	0,14	0,09	0,05	0,24

Таблица 1.3

Задачи на определение условий, при которых получаемое количество информации будет максимальным

№ задачи	Формулировка задачи
I	По цели может быть произведено n независимых выстрелов; вероятность поражения цели при каждом выстреле равна p. После k-го выстрела ($1 \le k \le n$) производится разведка, сообщающая, поражена или не поражена цель; если она поражена, то стрельба по ней прекращается. Определить число k, при котором количество информации, доставляемое разведкой, является максимальным.

	№ вар.	**1**	**5**	**9**	**13**	**17**	**21**
	p	0,2	0,15	0,07	0,04	0,19	0,05
II	На производственном предприятии за смену изготавливается n единиц продукции. Через k единиц продукции ($1 \le k \le n$) выполняется контроль их качества, при котором выявляется брак. Требуется определить k из того условия, чтобы количество информации, получаемое при контроле, было максимально. Вероятность брака составляет p.						
	№ вар.	**2**	**6**	**10**	**14**	**18**	**22**
	p	0,08	0,18	0,22	0,17	0,06	0,13
III	Техническое устройство за один цикл работы может выйти из строя с вероятностью p. За время T устройство успевает сделать n циклов. Для обнаружения отказа через k циклов ($1 \le k \le n$) производится диагностика устройства. Найти k, при котором количество информации, получаемой в ходе диагностики, будет максимальным.						
	№ вар.	**3**	**7**	**11**	**15**	**19**	**23**
	p	0,1	0,23	0,08	0,02	0,21	0,25
IV	За чётную и нечётную недели у студенческой группы проводится n занятий. Каждое из этих занятий с вероятностью p может быть отменено или перенесено. Для контроля над проведением занятий учебный отдел через k занятий ($1 \le k \le n$) может провести проверку. Определить число k, при котором количество информации, получаемой в ходе проверки, будет максимальным.						
	№ вар.	**4**	**8**	**12**	**16**	**20**	**24**
	p	0,12	0,09	0,11	0,04	0,16	0,14

Требования к отчёту.

Отчёт по практической работе должен содержать следующие пункты:

1. Титульный лист с указанием названия работы, фамилий и инициалов выполнившего и проверившего работу.
2. Цель и задачи работы.
3. Расчёты частного количества информации, полученного при наступлении событий, приведённых в табл. 1.1.
4. Расчёт среднего количества информации в сообщении, выданным источником сообщений с равновероятными символами алфавита.
5. Расчёт энтропии дискретного источника сообщений без памяти на основе данных из табл. 1.2.

6. Расчёт среднего количества информации в сообщении, состоящего из фамилии, имени и отчества студента при отсутствии статистической связи между буквами, а также при наличии статистической связи между парами и тройками букв.

7. Расчёт числа опытов, при котором получаемое количество информации будет максимальным.

1.4. Контрольные вопросы и задачи

Теоретические вопросы.

1. Что называют дискретным источником сообщений?
2. Какие дискретные источники сообщений называют источниками без памяти?
3. Как между собой связаны вероятность единичного события и количество информации, получаемое от осуществления этого события?
4. Что характеризует энтропия дискретного источника сообщений?
5. Каким образом энтропия может быть определена для дискретного источника сообщений?
6. Как можно найти количество информации, содержащееся в дискретном сообщении?
7. В каком случае формула Хартли применима для определения количества информации в сообщении?
8. При каких условиях энтропия дискретного источника сообщений равна нулю?
9. В каком случае энтропия дискретного источника сообщений максимальна?

Практические задачи.

1. Для источника сообщений без памяти, задаваемого ансамблем A (табл. 1.4) требуется определить энтропию, а также среднее количество информации в сообщении a, получаемом от источника A.

Таблица 1.4

Варианты заданий для расчёта энтропии и среднего количества информации

Вариант	Ансамбль источника сообщений	Сообщение
Нечётный	$A=\begin{pmatrix} a_1 & a_2 & a_3 \\ 0{,}1 & 0{,}6 & 0{,}3 \end{pmatrix}$	$\boldsymbol{a} = a_2a_1a_3a_2a_3$.
Чётный	$A=\begin{pmatrix} a_1 & a_2 & a_3 \\ 0{,}5 & 0{,}3 & 0{,}2 \end{pmatrix}$	$\boldsymbol{a} = a_1a_3a_2a_1$.

2. Изменить ансамбль источника сообщений в задаче 1, таким образом, чтобы количество информации в сообщении было максимальным. Определить чему будет равно максимальное количество информации.

2. ЭФФЕКТИВНОЕ КОДИРОВАНИЕ ИСТОЧНИКОВ СООБЩЕНИЙ

2.1. Цель и задачи работы

Цель работы – приобрести умение выполнять эффективное кодирование источников сообщений с помощью методов Шеннона-Фано и Хаффмана.

Основные задачи работы:

• научиться кодировать источники сообщений с помощью метода Шеннона-Фано;

• освоить кодирование источников сообщений при помощи метода Хаффмана.

Работа рассчитана на 4 часа.

2.2. Основные теоретические сведения

2.2.1. Понятие эффективного кодирования. Код Шеннона-Фано

Понятие эффективного кодирования.

Для того чтобы эффективнее использовать канал связи (устройство хранения), следует так преобразовывать порожденную источником сообщений информацию, чтобы ее передача (хранение) сопровождалась наименьшими затратами. Такое преобразование информации называется ***кодированием источника сообщений***.

При кодировании источника сообщений каждому символу алфавита $S = \{s_1, s_2, \dots, s_m\}$ источника соответствует последовательность символов алфавита $X = \{x_1, x_2, \dots, x_K\}$ кодера этого источника. Упорядоченная последовательность символов $\bar{x}_i = (x_{j_1}, x_{j_2}, ..., x_{j_L})$, соответствующая определенному символу s_i, называется ***кодовым словом***.

Под ***кодом*** понимают множество всех кодовых слов $\{\bar{x}_1, \dots, \bar{x}_N\}$.

Число символов L в кодовом слове называют ***длиной кодового слова.***

Средней длиной кодовых слов называется величина:

$$\bar{L} = \sum_{i=1}^{m} p(s_i) L_i ; \qquad (2.1)$$

где $p(s_i)$ – вероятность появления символа s_i из алфавита S размером m; L_i – длина кодового слова, соответствующего символу s_i.

Кодирование, при котором обеспечивается минимальная средняя длина кодовых слов, называется ***эффективным*** (оптимальным). В эффективном коде символу источника, встречающемуся чаще всего, присваивается наиболее короткое кодовое слово.

Эффективное кодирование базируется на ***теореме Шеннона о кодировании источников***, согласно которой можно так закодировать символы источника сообщений, что средняя длина $\bar{L}$ кодовых слов будет удовлетворять условию:

$$\frac{H}{\log_2 K} \le \bar{L} \le \frac{H}{\log_2 K} + 1; \qquad (2.2)$$

где H – энтропия источника сообщений, K – основание кода.

Из теоремы Шеннона следует, что минимальная средняя длина кодовых слов определяется соотношением:

$$\bar{L}_{\min} = \frac{H}{\log_2 K}; \qquad (2.3)$$

Для двоичного кода ($K = 2$) $\bar{L}_{\min} = H$.

Эффективностью E кода называется отношение $\bar{L}_{\min}$ к реально достигнутой в данном коде средней длине кодовых слов $\bar{L}$:

$$E = \frac{\bar{L}_{\min}}{\bar{L}} = \frac{H}{\bar{L} \log_2 K}. \qquad (2.4)$$

Избыточностью R кода называют величину:

$$R = 1 - \frac{\bar{L}_{\min}}{\bar{L}} = 1 - E. \qquad (2.5)$$

Метод Шеннона-Фано.

Для случая отсутствия статистической взаимосвязи между символами методы построения эффективных кодов впервые были предложены Шенноном и Фано. Поскольку эти методы существенно не отличаются, то общий метод получил название ***метода Шеннона-Фано***.

При построении двоичного кода метод Шеннона-Фано сводится к следующему:

1. Символы алфавита источника сообщений располагаются в порядке убывания вероятностей.

2. Алфавит источника разбивается на две группы таким образом, чтобы суммарные вероятности символов обеих групп были по возможности равны. Первой группе присваивается знак «1», второй – «0».

3. Каждую из образованных групп вновь делят на две части с приблизительно равными суммарными вероятностями и присваивают им 1 и 0. Таким образом, получают вторые цифры кода.

4. Процесс повторяется до тех пор, пока в каждой подгруппе не останется по одному символу.

Метод Шеннона-Фано не всегда приводит к однозначному построению кода. От указанного недостатка свободен метод Хаффмана.

❒ ***Пример* 2.1. *Построение кода Шеннона-Фано*.**

Требуется получить двоичный код Шеннона-Фано для источника без памяти, который описывается следующим ансамблем:

$$S = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ 0{,}14 & 0{,}07 & 0{,}12 & 0{,}11 & 0{,}22 & 0{,}05 & 0{,}16 & 0{,}13 \end{pmatrix}.$$

Также для кода необходимо определить эффективность E.

Дополнительно найдем энтропию $H(S)$, эффективность $E(S)$ и избыточность $R(S)$ источника S, которые по формулам (1.3), (2.2) и (2.3) будут:

$$H(S) = -(0{,}14\cdot\log_2 0{,}14 + 0{,}07\cdot\log_2 0{,}07 + 0{,}12\cdot\log_2 0{,}12 +$$

$$+ 0{,}11\cdot\log_2 0{,}11 + 0{,}22\cdot\log_2 0{,}22 + 0{,}05\cdot\log_2 0{,}05 + 0{,}16\cdot\log_2 0{,}16 +$$

$$+ 0{,}13\cdot\log_2 0{,}13) = 0{,}397 + 0{,}269 + 0{,}367 + 0{,}350 + 0{,}481 + 0{,}216 +$$

+ 0,423 + 0,383 = 2,885 (бит/символ);

$H_{max} = \log_2 8 = 3$ (бит/символ).

$E(S) = 2{,}885 / 3 = 0{,}962$;

$R(S) = 1 - 0{,}962 = 0{,}038$.

Для получения кода Шеннона-Фано построим таблицу, в которой по шагам будем производить разбиение символов на группы с близкими суммарными вероятностями (табл. 2.1).

Таблица 2.1.

Получение эффективного кода по методу Шеннона-Фано

s_i	$p(s_i)$	1		2		3		4		$\bar{x}_i$	L_i
s_5	0,22	1	0,52	1	0,22					**11**	2
s_7	0,16			0	0,30	1	0,16			**101**	3
s_1	0,14					0	0,14			**100**	3
s_8	0,13	0	0,48	1	0,25	1	0,13			**011**	3
s_3	0,12					0	0,12			**010**	3
s_4	0,11			0	0,23	1	0,11			**001**	3
s_2	0,07					0	0,12	1	0,07	**0001**	4
s_6	0,05							0	0,05	**0000**	4

Кодовое дерево, соответствующее полученному коду Шеннона-Фано, представлено на рис. 2.1.

Найдем по формуле (2.1) среднюю длину кодовых слов:

$\bar{L} = 0{,}14 \cdot 3 + 0{,}07 \cdot 4 + 0{,}12 \cdot 3 + 0{,}11 \cdot 3 + 0{,}22 \cdot 2 + 0{,}05 \cdot 4 +$

$+ 0{,}16 \cdot 3 + 0{,}13 \cdot 3 = 2{,}9.$

Отсюда эффективность кода Шеннона-Фано по формуле (2.4) будет:

$E = 2{,}885 / 2{,}9 = 0{,}995.$ ❐

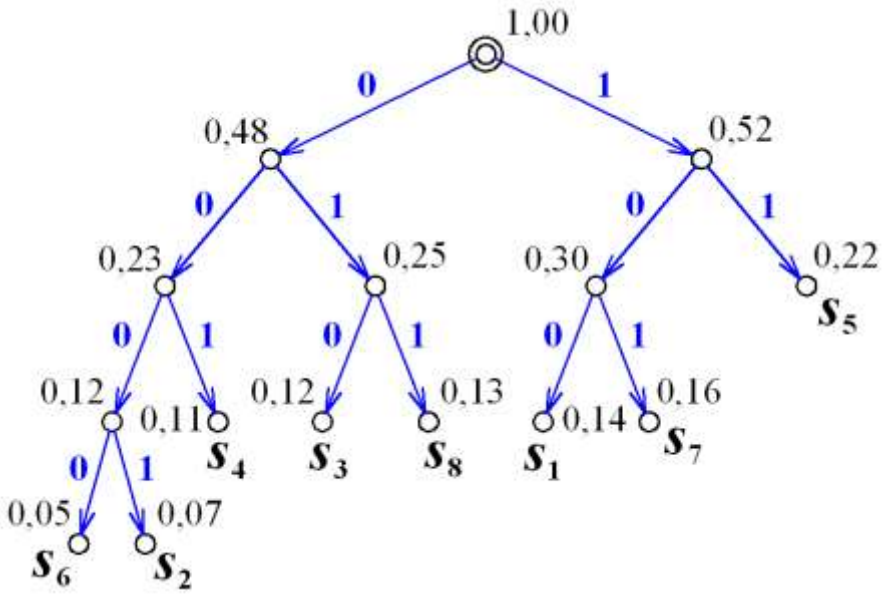


Рис. 2.1. Кодовое дерево для кода Шеннона-Фано

2.2.3. Кодирование источников сообщений с помощью метода Хаффмана

Кодирование дискретного источника сообщений без памяти по Хаффману.

Наиболее известным и широко используемым методом эффективного кодирования является метод Хаффмана.

Коды Хаффмана играют важную роль в кодировании изображений, звука и видео. Они являются составной частью стандартов JPEG, MPEG и H.261.

При кодировании источника сообщений без памяти двоичным кодом метод Хаффмана сводится к следующему:

1. Символы алфавита источника сообщений выписываются в столбец в порядке убывания вероятностей.

2. Два последних символа объединяются в один вспомогательный символ, которому приписывается суммарная вероятность.

3. Вероятности символов, участвующих в объединении и полученная суммарная вероятность вновь располагаются в порядке убывания вероятностей в дополнительном столбце, а два последних символа объединяются.

4. Процесс продолжается до тех пор, пока не будет получен единственный вспомогательный символ с суммарной вероятностью, равной 1.

Для получения кодового слова, соответствующего данному символу необходимо проследить путь перехода по строкам и столбцам полученной таблицы.

❒ ***Пример.* 2.3. *Построение кода Хаффмана для источника без памяти.***

Требуется получить двоичный код Хаффмана для источника без памяти, модель которого описана в примере 2.1. Для этого построим таблицу пошагового выполнения алгоритма (табл. 2.2).

Таблица 2.2.

Получение эффективного кода по методу Хаффмана

s_i	1	2	3	4	5	6	7	8
s_5	0,22 →	0,22 →	0,23 →	0,25 →	0,30 →	0,45 →	0,55 (1) →	1,00
s_7	0,16 →	0,16 →	0,22 →	0,23 →	0,25 →	0,30 (1) →	0,45 (0)	
s_1	0,14 →	0,14 →	0,16 →	0,22 →	0,23 (1) →	0,25 (0)		
s_8	0,13 →	0,13 →	0,14 →	0,16 (1) →	0,22 (0)			
s_3	0,12 →	0,12 →	0,13 (1) →	0,14 (0)				
s_4	0,11 →	0,12 (1) →	0,12 (0)					
s_2	0,07 (1) →	0,11 (0)						
s_6	0,05 (0)							

Кодовое дерево, полученное по методу Хаффмана, показано на рис. 2.2.

По кодовому дереву можно найти следующие кодовые слова:

$s_1 \to 110$; $s_2 \to 0111$;

$s_3 \to 100$; $s_4 \to 010$;

$s_5 \to 00$; $s_6 \to 0110$;

$s_7 \to 111$; $s_8 \to 101$.

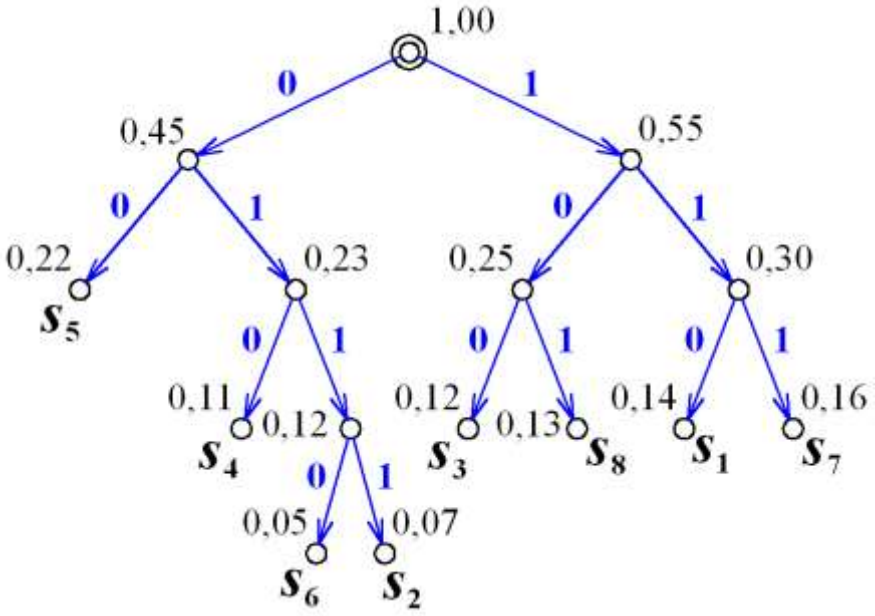


Рис. 2.2. Кодовое дерево для кода Хаффмана

Средняя длина $\bar{L}$ кодовых слов по формуле (2.1) будет:

$\bar{L}$ = 0,14·3 + 0,07·4 + 0,12·3 + 0,11·3 + 0,22·2 + 0,05·4 +

+ 0,16·3 + 0,13·3 = 2,9.

Эффективность кода Шеннона-Фано при известной энтропии источника (пример 4.1) найдем по формуле (2.4):

E = 2,885 / 2,9 = 0,995,

что совпадает с результатом из примера 2.1.

Таким образом, для заданной модели источника сообщений эффективность кодов, полученных по методам Шеннона-Фано и Хаффмана, одинакова. ❐

2.3. Порядок выполнения работы и варианты заданий

Основные этапы выполнения работы.

Данная практическая работа предполагает выполнение следующих этапов:

1. Изучить методические указания к самостоятельной практической работе.

2. Определить энтропию, эффективность и избыточность для дискретного источника без памяти с алфавитом $S = \{s_1, s_2, \ldots, s_{12}\}$. Вероятности появления символов источника сообщений приведены в табл. 2.3.

3. Получить в форме таблицы двоичный код Шеннона-Фано. Построить кодовое дерево для кода Шеннона-Фано и определить эффективность кода.

4. Получить эффективный код для заданного распределения вероятностей символов (табл. 2.3) с помощью метода Хаффмана.

5. Сравнить эффективности метода Шеннона-Фано и метода Хаффмана. Сделать выводы о полученных результатах.

6. Оформить и защитить отчет по практической работе.

Индивидуальные варианты заданий.

Таблица 2.3

Вероятности появления символов дискретного источника сообщений

№ вар.	**Алфавит источника сообщений**											
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}
1	0,14	0,06	0,05	0,08	0,13	0,04	0,01	0,09	0,15	0,02	0,11	0,12
2	0,11	0,05	0,09	0,10	0,12	0,03	0,02	0,08	0,15	0,07	0,14	0,04
3	0,13	0,07	0,05	0,06	0,15	0,04	0,11	0,02	0,12	0,16	0,08	0,01
4	0,02	0,11	0,12	0,01	0,09	0,15	0,08	0,13	0,04	0,14	0,06	0,05
5	0,07	0,14	0,04	0,02	0,08	0,15	0,10	0,12	0,03	0,11	0,05	0,09
6	0,16	0,08	0,01	0,11	0,02	0,12	0,06	0,15	0,04	0,13	0,07	0,05
7	0,01	0,09	0,15	0,02	0,11	0,12	0,14	0,06	0,05	0,08	0,13	0,04
8	0,02	0,08	0,15	0,07	0,14	0,04	0,13	0,07	0,05	0,06	0,15	0,04
9	0,11	0,02	0,12	0,16	0,08	0,01	0,07	0,05	0,13	0,06	0,15	0,04
10	0,06	0,05	0,14	0,13	0,04	0,08	0,15	0,01	0,09	0,12	0,02	0,11
11	0,09	0,11	0,05	0,03	0,10	0,12	0,15	0,02	0,08	0,14	0,04	0,07
12	0,05	0,13	0,07	0,15	0,04	0,06	0,02	0,12	0,11	0,04	0,07	0,14
13	0,12	0,02	0,11	0,09	0,15	0,01	0,13	0,04	0,08	0,06	0,05	0,14
14	0,14	0,04	0,07	0,15	0,02	0,08	0,03	0,10	0,12	0,09	0,11	0,05
15	0,04	0,07	0,14	0,02	0,12	0,11	0,15	0,04	0,06	0,05	0,13	0,07
16	0,05	0,07	0,13	0,04	0,15	0,06	0,12	0,02	0,11	0,01	0,08	0,16
17	0,1	0,03	0,05	0,09	0,14	0,04	0,01	0,08	0,16	0,04	0,12	0,14
18	0,12	0,07	0,08	0,11	0,16	0,01	0,04	0,06	0,13	0,09	0,1	0,03
19	0,11	0,08	0,07	0,04	0,14	0,05	0,13	0,02	0,1	0,15	0,09	0,02
20	0,03	0,12	0,14	0,02	0,08	0,15	0,1	0,11	0,03	0,12	0,05	0,05

№ вар.	Алфавит источника сообщений											
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}
21	0,08	0,13	0,05	0,01	0,06	0,14	0,11	0,13	0,06	0,1	0,04	0,09
22	0,15	0,09	0,02	0,13	0,02	0,1	0,08	0,16	0,01	0,12	0,06	0,06
23	0,03	0,11	0,16	0,05	0,14	0,15	0,09	0,01	0,04	0,07	0,13	0,02
24	0,06	0,01	0,12	0,09	0,16	0,02	0,11	0,03	0,08	0,05	0,15	0,12

Требования к отчёту.

Отчёт по практической работе должен содержать следующие пункты:

1. Титульный лист с указанием названия работы, фамилий и инициалов выполнившего и проверившего работу.
2. Цель и задачи работы.
3. Таблица с кодированием по методу Шеннона-Фано и полученные кодовые слова.
4. Кодовое дерево для кода Шеннона-Фано, а также его эффективность.
5. Таблица с кодированием по методу Хаффмана и полученные кодовые слова.
6. Кодовое дерево для кода Хаффмана, а также его эффективность.
7. Вывод о том, какой из методов кодирования является более эффективным.

2.4. Контрольные вопросы и задачи

Теоретические вопросы.

1. В чем заключается сущность эффективного кодирования?
2. Каковы основные задачи эффективного кодирования?
3. Как определяется средняя длина кодового слова?
4. Чему равна нижняя граница эффективного кодирования?
5. Как определяется эффективность и избыточность кода?
6. Как осуществляется кодирование источника по методу Шеннона-Фано?
7. В каком случае метод Шеннона-Фано гарантированно обеспечивает получение эффективного кода?

8. С помощью какой операции в методе Хаффмана обеспечивается получение вспомогательных символов?

9. Как получают кодовые слова по методу Хаффмана?

Практические задачи.

1. Для источника без памяти, заданного ансамблем ***S*** (табл. 2.4), требуется построить код Хаффмана и соответствующее ему кодовое дерево. Определить эффективность и избыточность полученного кода.

Таблица 2.4

Варианты заданий для расчёта энтропии и среднего количества информации

Вариант	**Ансамбль источника сообщений без памяти**
Нечётный	$S = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ 0{,}35 & 0{,}2 & 0{,}4 & 0{,}05 \end{pmatrix}$
Чётный	$S = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ 0{,}1 & 0{,}15 & 0{,}5 & 0{,}25 \end{pmatrix}$

3. ПОСТРОЕНИЕ И ДЕКОДИРОВАНИЕ КОДОВ ХЭММИНГА

3.1. Цель и задачи работы

Цель работы – приобрести умение строить и декодировать коды Хэмминга для исправления одиночных ошибок в кодовых словах и обнаружения двойных ошибок.

Основные задачи работы:

• научиться строить и декодировать двоичный код Хэмминга для исправления одиночных ошибок;

• освоить построение и декодирование расширенного кода Хэмминга для обнаружения двойных ошибок.

Работа рассчитана на 4 часа.

3.2. Основные теоретические сведения

3.2.1. Блоковые корректирующие коды и их характеристики. Коды Хэмминга

Блоковые корректирующие коды и их характеристики.

Блоковыми называют помехоустойчивые коды, в которых процедура кодирования заключается в разбиении входной последовательности информационных символов на блоки, содержащие m символов. Каждому информационному блоку длиной m сопоставляется k проверочных символов. Полученное кодовое слово из $n = m + k$ символов называют кодовым блоком.

Число несовпадающих позиций в двух кодовых словах $\bar{x}$ и $\bar{y}$ называется ***расстоянием Хэмминга*** $d(\bar{x}, \bar{y})$ между этими словами.

Для двоичных кодовых слов расстояние Хэмминга может быть получено как число единиц в сумме в кодовых словах по модулю 2. Правила сложения по модулю 2 определяются следующим образом:

$$0 \oplus 0 = 0;\ \ 0 \oplus 1 = 1;\ \ 1 \oplus 0 = 1;\ \ 1 \oplus 1 = 0.$$

Важной характеристикой корректирующего блокового кода C является ***кодовое расстояние***, которое принимается равным

наименьшему расстоянию Хэмминга между словами данного кода:

$$d(C) = \min\{\, d(\bar{x}, \bar{y}) :\ \bar{x}, \bar{y} \in C; \bar{x} \neq \bar{y}\}. \qquad (3.1)$$

Для блоковых кодов справедливы следующие утверждения:

1. Для того чтобы блоковый код C позволял обнаруживать все комбинации из t или менее ошибок, необходимо и достаточно, чтобы его кодовое расстояние было равно $d(C) = t + 1$.

2. Для того чтобы блоковый код C позволял исправлять все комбинации из t или менее ошибок необходимо и достаточно, чтобы его кодовое расстояние было равно $d(C) = 2t + 1$.

Для практических расчетов при определении числа проверочных символов k в коде с кодовым расстоянием $d(C) = 3$ используют следующие формулы:

если известна длина полного кодового слова n, то

$$k = \lceil \log_2(n+1) \rceil; \qquad (3.2)$$

если при расчетах удобнее исходить из заданного числа информационных символов m, то

$$k = \lceil \log_2\{(m+1) + \lceil \log_2(m+1) \rceil\} \rceil; \qquad (3.3)$$

где $\lceil ... \rceil$ – округление числа, стоящего в скобках, до целого в большую сторону.

Для блоковых кодов с $d(C) = 4$

$$k \geq 1 + \log_2(n+1); \qquad (3.4)$$

или

$$k \geq 1 + \lceil \log_2\{(m+1) + \log_2(m+1)\} \rceil. \qquad (3.5)$$

Двоичные коды Хэмминга.

Систематические коды представляют собой блоковые корректирующие коды, в которых информационные и проверочные символы расположены по строго определенной системе и всегда занимают определенные места в кодовых словах.

Наиболее простыми систематическими кодами, получившими широкое практическое применение, являются коды ***Хэммин-***

га[1], имеющие кодовое расстояние $d = 3$. Данные коды позволяют исправлять одиночные ошибки в кодовых словах.

Двоичные коды Хэмминга строятся следующим образом:

1. Определяется число k проверочных символов из условия (3.3).

2. Выбираются места расположения проверочных символов из условия, чтобы проверочные символы участвовали только в одной операции подсчета четности с целью упрощения процесса кодирования. Такими местами являются символы с номерами, являющиеся целыми степенями числа 2, т. е. 1, 2 ,4, 8, 16 и т. д. Символы в кодовых словах Хэмминга нумеруются слева направо.

3. Определяются значения символов слова, называемого ***синдромом***:

$$S_k S_{k-1} \ldots S_2 S_1,$$

из уравнений

$$S_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7 \oplus \ldots = 0,$$

то есть складываются по модулю 2 значения тех символов, двоичное представление номеров которых содержат 1 в последнем разряде:

$$S_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7 \oplus \ldots = 0,$$

то есть складываются значения тех символов, двоичное представление номеров которых содержат 1 в предпоследнем разряде.

Аналогично получаются выражения для нахождения значений S_3, S_4, ... , S_k.

Так

$$S_3 = x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus \ldots = 0;$$

$$S_4 = x_8 \oplus x_9 \oplus x_{10} \oplus x_{11} \oplus \ldots = 0;$$

где x_1, x_2, x_3, x_4, ... – значения символов с номерами 1, 2, 3, 4, ...

❒ ***Пример* 3.1. *Получение кодового слова Хэмминга*.**

Требуется получить кодовое слово Хэмминга для двоичного кодового слова

[1] Разработаны американским учёным Ричардом Вэсли Хэммингом в 1950 г.

$$\bar{x} = 011111.$$

Рассматриваемое кодовое слово содержит шесть информационных символов, то есть $m = 6$. Число проверочных символов k для кода Хэмминга, имеющего кодовое расстояние $d = 3$, по формуле (3.3) будет:

$$k = \lceil \log_2 \{(6+1) + \lceil \log_2 (6+1) \rceil\} \rceil = \lceil \log_2 (7+3) \rceil = 4.$$

Следовательно, кодовое слово Хэмминга для слова $\bar{x}$ = 011111 будет содержать 10 символов, то есть $n = 10$. При этом проверочными символами будут 1, 2, 4, 8, а информационными символами соответственно будут 3, 5, 6, 7, 9, 10.

Прономеруем и запишем значения информационных символов:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
k_1	k_2	m_1	k_3	m_2	m_3	m_4	k_4	m_5	m_6
		0		1	1	1		1	1

Определим значения проверочных символов. Значение символа $k_1 = x_1$ определяется из условия:

$S_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_9 = 0;$

$S_1 = x_1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = x_1 \oplus 1 = 0.$

Отсюда $k_1 = x_1 = 1$.

Значения символа $k_2 = x_2$ определяется из условия:

$S_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_{10} = 0;$

$S_2 = x_2 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = x_2 \oplus 1 = 0.$

Отсюда $k_2 = x_2 = 1$.

Значения символа $k_3 = x_4$ определяется из условия:

$S_3 = x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 0;$

$S_3 = x_4 \oplus 1 \oplus 1 \oplus 1 = x_4 \oplus 1 = 0.$

Отсюда $k_3 = x_4 = 1$.

Значение символа $k_4 = x_8$ определяется из условия:

$S_4 = x_8 \oplus x_9 \oplus x_{10} = 0;$

$S_4 = x_8 \oplus 1 \oplus 1 = x_8 \oplus 0 = 0.$

Отсюда $k_4 = x_8 = 0$.

В итоге кодовое слово Хэмминга для кодового слова $\bar{x}$ = 011111 будет иметь вид $\bar{x}_H$ = 1101111011. ❒

Местоположение ошибки, то есть определение символа с ошибкой осуществляется по значению синдрома. Если синдром состоит одних нулей, т. е.

$$S_k S_{k-1} ... S_2 S_1 = 00...00,$$

то ошибка отсутствует. Если в синдроме есть символы, отличные от 0, то это говорит о наличии ошибки. Например, если $S_4S_5S_2S_1$ = 1000, то это означает, что ошибка содержится в восьмом символе, так как $1000_2 = 8_{10}$.

Другими словами, синдром в коде Хэмминга определяет номер символа с ошибкой. Исправление ошибки осуществляется заменой 0 на 1 либо наоборот 1 на 0.

❒ ***Пример* 3.2.** ***Декодирование кодового слова Хэмминга.***

Проверим работу кода Хэмминга для полученного в примере 7.1 кодового слова, заменив в нем значение одного из символов, то есть введем одиночную ошибку в кодовое слово Хэмминга. Например, заменим значение пятого символа слова с 1 на 0, то есть вместо $\bar{x}_H$ = 1101111011 примем $\bar{x}_H$ = 1101**0**11011.

Определим значения символов синдрома S_1, S_2, S_3, S_4. Для этого прономеруем символы кодового слова Хэмминга:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	1	0	1	**0**	1	1	0	1	1

Отсюда значения символов синдрома будут:

$S_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_9 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$;

$S_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_{10} = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0$;

$S_3 = x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1$;

$S_4 = x_8 \oplus x_9 \oplus x_{10} = 0 \oplus 1 \oplus 1 = 0$.

Таким образом, синдром имеет следующий вид:

$$S_4S_3S_2S_1 = 0101.$$

Поскольку $0101_2 = 5_{10}$, то ошибка содержится в пятом символе, что и требовалось проверить. ❒

3.2.2. Расширенные коды Хэмминга

Расширенные коды Хэмминга.

Кроме описанных кодов Хэмминга существуют так называемые расширенные коды Хэмминга.

Расширенные коды Хэмминга получаются путём введения дополнительной проверки на чётность символов кодового слова. В результате кодовое расстояние увеличивается до $d = 4$, что позволяет данным кодам исправлять одну и обнаруживать две ошибки.

Для получения расширенного кода в конце каждого кодового слова Хэмминга следует добавить проверочный символ таким образом, чтобы сумма единиц в полученном слове всегда была четной.

В случае одной ошибки проверка по синдрому укажет номер ошибочного символа, а проверка на четность укажет наличие ошибки. Если проверка по синдрому укажет на наличие ошибки, а проверка на четность не фиксирует ошибку, то в кодовом слове присутствуют две ошибки.

❒ ***Пример* 3.3. *Получение и декодирование расширенного кодового слова Хэмминга.***

Требуется получить расширенное кодовое слово Хэмминга для слова $\bar{x}_H$ = 1101111011, полученного в примере 7.1. Для этого добавим в конце данного кодового слова бит четности.

Поскольку число единиц в кодовом слове четное (8 единиц), то бит четности будет 0. Отсюда расширенное кодовое слово Хэмминга запишется следующим образом:

$$\bar{x}_{HE} = 11011110110.$$

• ***Обнаружение и исправление одиночной ошибки.***

Сделаем в полученном кодовом слове одиночную ошибку:

1101111**0**10

Поскольку число единиц не чётное (7 единиц), то проверка на чётность свидетельствует о наличии ошибки.

Найдём для кодового слова с ошибкой синдром, предварительно пронумеровав его символы:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
1	1	0	1	1	1	1	0	**0**	1	0

Символы синдрома будут иметь следующие значения:

$S_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_9 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1;$

$S_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_{10} = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0;$

$S_3 = x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 1 \oplus 1 \oplus 1 \oplus 1 = 0;$

$S_4 = x_8 \oplus x_9 \oplus x_{10} = 0 \oplus 0 \oplus 1 = 1.$

Таким образом, получен синдром:

$$S_4S_3S_2S_1 = 1001.$$

Результат $1001_2 = 9_{10}$ соответствует ошибке в девятом символе.

Поскольку проверка на чётность фиксирует ошибку, то бит четности показывает номер ошибочного символа.

• ***Обнаружение двойной ошибки.***

Добавим в это кодовое слово двойную ошибку, которая изменит значения пятого и восьмого знаков. Полученное кодовое слово с ошибкой будет:

$$1101\mathbf{0}111\mathbf{1}10$$

Так как в кодовом слове 11010111110 четное число единиц (8 единиц), то проверка на четность не фиксирует ошибку.

Для получения синдрома прономеруем символы расширенного кодового слова Хэмминга:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
1	1	0	1	**0**	1	1	**1**	1	1	0

Отсюда значения символов синдрома будут:

$S_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_9 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1;$

$S_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_{10} = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0;$

$S_3 = x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1;$

$S_4 = x_8 \oplus x_9 \oplus x_{10} \oplus x_{11} = 1 \oplus 1 \oplus 1 \oplus 0 = 1.$

Синдром имеет следующий вид:

$$S_4S_3S_2S_1 = 1101.$$

Отсюда получим $1101_2 = 13_{10}$, что означает наличие ошибки, но не соответствует ни одному из символов кодового слова.

Поскольку синдром указывает на наличие ошибки, а бит четности ошибку не фиксирует, то в кодовом слове присутствует двойная ошибка, которая не может быть исправлена с помощью кода Хэмминга. ❐

3.3. Порядок выполнения работы и варианты заданий

Основные этапы выполнения работы.

Данная практическая работа предполагает выполнение следующих этапов:

1. Изучить методические указания к лабораторной работе.
2. Для заданных кодовых слов (табл. 3.1) построить кодовые слова Хэмминга.
3. Проверить работоспособности кода путем изменения значения одного из символов в полученных кодовых словах Хэмминга и вычисления синдромов для слов с одиночной ошибкой.
4. Получить одно слово для расширенного кода Хэмминга и произвести для него исправление одиночной ошибки и обнаружение двойной ошибки.
5. Оформить и защитить отчет по практической работе.

Индивидуальные варианты заданий.

Таблица 3.1

Варианты заданий для получения кодовых слов Хэмминга

Вар.	**Передаваемые кодовые слова**		
1	0111010	1101010	0101011
2	0001010	0001101	0010001
3	0010100	0011000	0011101
4	0011110	0110000	0100010
5	1100011	0001111	0011101
6	1001010	0011001	0100010
7	0001011	0000110	1010001
8	0010101	1000111	1111001
9	1011110	1011000	0100001
10	1000010	1001111	1011101
11	1000011	1100000	0011101
12	0011010	1101010	0101010
13	0000111	1000110	1001000
14	0001110	1001110	0010011
15	0011100	0011010	1011101
16	0010110	0101000	1100010
17	0001011	1101110	0001000
18	1011010	1011001	1100010
19	0001001	0000110	1110001
20	0010111	0100000	1101000
21	1010110	1010000	0110001
22	1000011	0100100	0001000
23	0000010	1100100	0111101
24	0000011	0100110	1010101

Требования к отчёту.

Отчёт по практической работе должен содержать следующие пункты:

1. Титульный лист с указанием названия работы, фамилий и инициалов выполнившего и проверившего работу.
2. Цель и задачи работы.
3. Получение кодовых слов Хэмминга.
4. Декодирование кодовых слов Хэмминга с исправлением одиночной ошибки.

5. Получение расширенного кодового слова Хэмминга и его декодирование с исправлением одиночной ошибки и с обнаружением двойной ошибки.

3.4. Контрольные вопросы и задачи

Теоретические вопросы.

1. Какие помехоустойчивые коды называют блоковыми?
2. Что такое расстояние Хэмминга?
3. Что называют кодовым расстоянием?
4. Какие коды называют линейными блоковыми?
5. Как определяется кодовое расстояние для линейного блокового кода?
6. Какие коды называют систематическими?
7. Сколько ошибок способен обнаруживать и исправлять код Хэмминга?
8. По какому правилу определяют число проверочных символов в коде Хэмминга?
9. В каких местах располагаются проверочные символы в кодовых словах Хэмминга?
10. По какому правилу строятся уравнения для нахождения проверочных символов в коде Хэмминга?
11. Какую информацию при декодировании кода Хэмминга дает синдром?
12. Каким образом обеспечивается обнаружение двойных ошибок в расширенном коде Хэмминга?

Практические задачи.

1. Используя код Хэмминга, для заданного информационного вектора получить кодовое слово. Сделать одиночную ошибку в кодовом слове и при декодировании произвести исправление этой ошибки.

Информационные векторы:

- $\boldsymbol{x}$ = **0110011** (нечётный вариант);
- $\boldsymbol{x}$ = **1101100** (чётный вариант).

4. ШИФРОВАНИЕ СООБЩЕНИЙ МЕТОДАМИ ПЕРЕСТАНОВКИ И ЗАМЕНЫ

4.1. Цель и задачи работы

Цель работы – приобрести умение шифровать сообщения с помощью простейших методов перестановки и замены, а также расшифровывать полученные криптограммы.

Основные задачи работы:

• освоить шифрование сообщений с помощью шифров перестановки с фиксированным периодом и вертикальной перестановки;

• научиться шифровать сообщения с помощью шифра Виженера.

Работа рассчитана на 4 часа.

4.2. Основные теоретические сведения

4.2.3. Основные понятия криптографии. Шифры перестановки

Основные понятия криптографии.

Основными понятиями криптографии являются такие понятия, как шифр, открытый и шифрованный текст, ключ, противник и др.

Шифр (*cipher*, криптографическая система) – совокупность заранее оговоренных способов преобразования исходного сообщения с целью его защиты от прочтения противником.

Исходное сообщение называют ***открытым текстом*** (*plaintext*). Сообщение, полученное после применение шифра, называется ***шифрованным текстом*** (*ciphertext*, шифротекст, закрытый текст, криптограмма).

Шифр включает в себя постоянную компоненту – алгоритм шифрования, который может являться общеизвестным, и сменную компоненту – ключ (*key*), который должен быть засекречен.

Ключ (криптографический ключ) – параметр, позволяющий выбрать одно конкретное преобразование из всех вариантов, предусмотренных алгоритмом шифрования.

Множество всех возможных ключей для данного шифра называют ***пространством ключей***.

Шифрованием (*encryption*) называется обратимое преобразование открытого текста в шифрованный текст с помощью ключа. Математически шифрование записывается следующим образом:

$$C = E_{k1}(M); \qquad (4.1)$$

где M – открытый текст; E – функция шифрования; k_1 – ключ шифрования; C – шифрованный текст.

Обратное шифрованию действие называется ***расшифрованием*** (*decryption*) и может быть записано следующим образом:

$$M = D_{k2}(C); \qquad (4.2)$$

где D – функция расшифрования, которая является обратной к функции E; k_2 – ключ расшифрования.

Для шифров, называемых симметричными, ключи k_1 и k_2 совпадают, то есть $k_1 = k_2 = k$.

Под ***дешифрованием*** понимают процесс получения информации из шифрованного текста без знания ключа. Дешифрованием занимается криптоаналитик (противник).

Криптоаналитик (*cryptoanalitic*) – человек, осуществляющих анализ шифрованного текста с целью получения открытого текста и/или ключа.

Атакой на шифр (*cryptoanalitic attack* – криптографической атакой) называется попытка криптоаналитика вызвать отклонения от нормального проведения обмена секретной информации. Успешно проведенная криптографическая атака называется ***вскрытием*** (взломом) ***шифра***.

Стойкостью шифра (*cryptographic strength* – криптографической стойкостью) называется характеристика шифра, определяющая его способность противостоять криптографическим атакам. В современных шифрах криптографическая стойкость определяется секретностью ключа.

На рис. 4.1 показана система передачи секретной информации и её основные элементы.

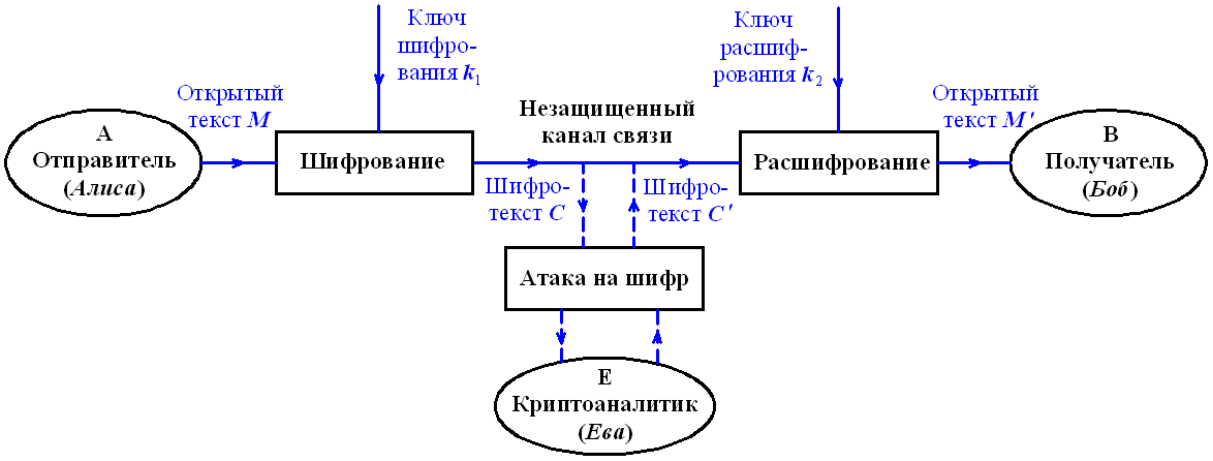


Рис. 4.1. Система передачи секретной информации

Классическими шифрами известными с древних времён являются шифры замены и шифры перестановки. Данные виды шифров представляют лишь академический интерес, поскольку не обеспечивают требуемой стойкости для современных методов и средств криптоанализа.

Изучение элементарных методов шифрования позволяет проследить эволюцию шифров и лучше понять работу современных алгоритмов шифрования.

Особенности шифров перестановки. Перестановка с фиксированным периодом.

В ***шифрах перестановки*** (*transposition ciphers*) шифрование заключается в изменении порядка следования букв открытого текста. Для реализации шифра перестановки открытый текст разбивается на блоки, и шифруется поблочно в соответствии с заданными правилами перестановки.

Наиболее простым видом шифров перестановки является ***перестановка с фиксированным периодом*** n. В данном шифре сообщение делится на группы символов длины n и каждой группе применяется одна и та же перестановка. Эта перестановка и является ключом:

$$k = \begin{pmatrix} 1 & 2 & \dots & i & \dots & n \\ k_1 & k_2 & \dots & k_i & \dots & k_n \end{pmatrix} = (k_1, k_1, \dots, k_i, \dots, k_n);$$

где k_i – номер места шифротекста, на которое попадает i-ый символ открытого текста при заданной перестановке.

Шифр простой перестановки легко поддаётся взлому атакой с выбором открытого текста. Для этого требуется навязать отправителю нужный открытый текст и получить его в зашифрованном виде. Путём сопоставления открытого текста и шифротекста можно легко получить перестановку.

❒ ***Пример* 4.1. *Шифрование и расшифрование перестановкой с фиксированным периодом.***

• ***Шифрование.***

Требуется зашифровать перестановкой с фиксированным периодом следующий открытый текст:

грузите_апельсины_бочками

В качестве ключа используется перестановка: (4, 5, 1, 3, 2).

Для шифрования запишем под открытым текстом необходимое число перестановок. Переставляя буквы исходного сообщения в соответствии с числами перестановки, получим требуемый шифротекст.

Процесс шифрования заданного открытого текста приведён в табл. 4.1.

Таблица 4.1

Шифрование перестановкой с фиксированным периодом

г	р	у	з	и	т	е	_	а	п	е	л	ь	с	и	н	ы	_	б	о	ч	к	а	м	и
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
4	5	1	3	2	4	5	1	3	2	4	5	1	3	2	4	5	1	3	2	4	5	1	3	2
З	И	Г	У	Р	А	П	Т	_	Е	С	И	Е	Ь	Л	Б	О	Н	_	Ы	М	И	Ч	А	К

Шифротекст: **ЗИГУРАПТ_ЕСИЕЬЛБОН_ЫМИЧАК**.

• ***Расшифрование.***

Требуется с помощью ключа (2, 6, 5, 3, 1, 4) расшифровать следующую криптограмму, полученную перестановкой с фиксированным периодом:

РДОИПХ_ТВЗИААП__РВЛЬЧНОО

Процесс расшифрования криптограммы приведён в табл. 4.2.

Таблица 4.2

Расшифрование перестановкой с фиксированным периодом

Р	Д	О	И	П	Х	_	Т	В	З	И	А	А	П	_	_	Р	В	Л	Ь	Ч	Н	О	О
2	6	5	3	1	4	2	6	5	3	1	4	2	6	5	3	1	4	2	6	5	3	1	4
1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
п	р	и	х	о	д	и	_	з	а	в	т	р	а	_	в	_	п	о	л	н	о	ч	ь

Получен открытый текст: **приходи_завтра_в_полночь.** ❐

Маршрутные перестановки. Шифр вертикальной перестановки.

Маршрутные перестановки представляют собой перестановочные шифры, в которых отрытый текст записывается в определённую геометрическую фигуру по некоторой траектории. Шифротекстом является последовательность, полученная при выписывании текста по другой траектории. Например, можно записывать сообщение в прямоугольную таблицу, двигаясь по горизонтали, начиная с левого верхнего угла, поочерёдно слева направо и справа налево. Списывать сообщение можно двигаясь по вертикали, начиная с верхнего правого угла, поочерёдно сверху вниз и снизу вверх.

Широкое распространение получила разновидность маршрутной перестановки, называемая ***вертикальной*** (***столбцовой***) ***перестановкой***. В этом шифре используется прямоугольная таблица, в которую сообщение записывается обычным образом (по строкам слева направо). Выписывается же сообщение по вертикали (сверху вниз), при этом столбцы выбираются в порядке, определяемом числовым ключом.

❒ ***Пример* 4.1. *Шифрование и расшифрование с помощью вертикальной перестановки.***

• ***Шифрование***.

Требуется зашифровать методом вертикальной перестановки по таблице 5 × 5 следующее сообщение:

грузите_апельсины_бочками

Ключом является перестановка: (3, 5, 2, 1, 4).

Шифруемое сообщение, записанное в таблицу, представлено в табл. 4.3

Таблица 4.3

Шифрование с помощью вертикальной перестановки

г	р	у	з	и
т	е	_	а	п
е	л	ь	с	и
н	ы	_	б	о
ч	к	а	м	и
1	2	3	4	5
3	5	2	1	4
у	и	р	г	з
_	п	е	т	а
ь	и	л	е	с
_	о	ы	н	б
а	и	к	ч	м

Шифротекст при выписывании символов из табл. 4.3 по вертикали будет:

У_Ь_АИПИОИРЕЛЫКГТЕНЧЗАСБМ

• ***Расшифрование***.

Требуется с помощью ключа (3, 1, 4, 2, 6, 5) расшифровать шифротекст, полученный вертикальной перестановкой по таблице размером 4 × 6:

ИЗ_НПИРОХАВОР_АЛДТПЬОВ_Ч.

Для расшифрования запишем по вертикали шифротекст в таблицу указанного размера. Процесс расшифрования показан в табл. 4.4.

Таблица 4.4

Расшифрование с помощью вертикальной перестановки

И	П	Х	Р	Д	О
З	И	А	_	Т	В
_	Р	В	А	П	_
Н	О	О	Л	Ь	Ч
3	1	4	2	6	5
1	2	3	4	5	6
П	Р	И	Х	О	Д
И	_	З	А	В	Т
Р	А	_	В	_	П
О	Л	Н	О	Ч	Ь

Получен открытый текст: **приходи_завтра_в_полночь.** ❐

4.2.2. Шифры замены. Шифр Виженера

Особенности шифров замены. Шифры сдвига и простой замены.

Шифры замены (подстановки) – это шифры, в которых элементы открытого текста заменяются на другие символы, но, в отличие от шифров перестановки, остаются на своих местах. Заменяемыми элементами открытого текста могут быть отдельные символы (самый распространённый случай), пары букв, тройки букв, комбинирование этих случаев.

Простейшими шифрами замены являются шифры сдвига и простой замены.

В ***шифрах сдвига*** процесс шифрования заключается в замене каждого символа на другой символ, отстоящий от исходного символа на определенное число позиций в алфавите.

Наиболее известным шифром сдвига является ***шифр Цезаря***[2], в котором каждая буква сообщения сдвигается на три буквы по алфавиту.

Основной недостаток шифров сдвига заключается в том, что существует слишком мало возможных ключей. Общее число ключей не превышает числа символов алфавита. Поэтому нужный ключ можно легко получить перебором возможных ключей.

Шифр простой замены (подстановки) подразумевает замену каждого символа сообщения на другой символ (обычно из того же алфавита). Ключ в таком шифре является просто перестановкой алфавита.

Количество возможных ключей N в шифре простой замены совпадает с числом всех возможных перестановок алфавита, то есть может быть найдено следующим образом:

$$N = m!;$$

где m – объём алфавита.

Шифр простой подстановки может быть взломан с помощью ***частотного криптоанализа***, в котором для символов шифротекста ищут частоты их появления. Кроме того, часто используют частоты встречаемости пар символов (биграмм) и троек символов (триграмм). Частоты отдельных символов, биграмм и триграмм шифротекста будут соответствовать вероятностям элементов естественного языка.

Многоалфавитные шифры замены. Шифр Виженера.

Шифры сдвига и простой замены относятся к ***шифрам моноалфавитной замены***, в которых используется только один упорядоченный набор символов, заменяющий стандартный алфавит. При криптоанализе таких шифров эффективно работает статистика языка.

Для увеличения стойкости шифров можно использовать несколько наборов символов вместо стандартного алфавита. Шифрование открытого текста будет заключаться в выборе соответствующих символов из разных наборов в определенной последо-

[2] Назван по имени древнеримского политика и полководца Гая Юлия Цезаря, который использовал данный шифр сдвига и изложил его в сочинении «Записки о Галльской войне» (I в. до н.э.)

вательности. Шифры такого типа называются ***многофавитными шифрами замены***. Примерами многоалфавитных шифров являются шифры Виженера, Бофора, Вернама.

В ***шифре Виженера***[3] ключ задается короткой последовательностью символов. Эта последовательность записывается с повторением под открытым текстом. Полученные две последовательности складываются по модулю, равному объему алфавита. Шифр Виженера можно рассматривать, как многократное применение шифра Цезаря с разными значениями сдвига.

Шифрование по Виженеру формально определяется следующим образом:

$$c_i = (t_i + k_{i \bmod d}) \bmod m; \qquad (4.3)$$

где t_i, k_i, c_i – порядковые номера символов в алфавите соответственно для открытого текста, ключа и шифротекста; m – объем алфавита.

Расшифровывание по Виженеру записывается следующим образом:

$$t_i = (c_i - k_{i \bmod d} + m) \bmod m. \qquad (4.4)$$

Для шифрования и расшифровывания по шифру Виженера можно использовать ***таблицу Виженера***, которая состоит из циклически сдвигаемых алфавитов (табл. П.3).

Первая строка таблицы Виженера служит алфавитом открытого текста, а первый столбец – алфавитом ключа. Последовательность символов открытого текста и ключевая последовательность записываются друг под другом. При этом пара символов, стоящих друг под другом, указывает, соответственно, номера строк и столбцов таблицы, на пересечении которых находится знак шифрованного текста.

Для расшифровывания с помощью таблицы Виженера ключевая последовательность и последовательность символов шифротекста записываются друг под другом. Для каждой пары символов в первом столбце таблицы Виженера ищем символ ключевой последовательности. В строке символа ключа находим соответствующий символ шифротекста. Верхний символ столбца, в

[3] Назван в честь французского дипломата Блеза де Виженера, который в 1586 г. в своем «Трактате о шифрах» описал собственный шифр многоалфавитной замены

котором расположен символ шифротекста, является символом открытого текста.

Для криптоанализа шифра Виженера может быть использован метод Казиски[4].

❒ ***Пример* 4.3. *Шифрование и расшифрование с помощью шифра Виженера.***

• ***Шифрование.***

Требуется с помощью шифра Виженера зашифровать текст:

карл_у_клары_украл_кораллы

В качестве ключа будет выступать слово **лорнет.**

Чтобы получить шифротекст с помощью таблицы Виженера, запишем поток ключей под открытым текстом. Затем находим символы шифротекста, расположенные на пересечении соответствующей строки и столбца. Процесс шифрования приведён в табл. 4.5.

Таблица 4.5

Шифрование с помощью шифра Виженера

к	а	р	л	_	у	_	к	л	а	р	ы	_	у	к	р	а	л	_	к	о	р	а	л	л	ы
л	о	р	н	е	т	л	о	р	н	е	т	л	о	р	н	е	т	л	о	р	н	е	т	л	о
Ц	О	А	Щ	Д	Е	К	Щ	Ь	Н	Х	М	К	Б	Ы	Ю	Е	Ю	К	Щ	Я	Ю	Е	Ю	Ч	И

Таким образом, получен шифротекст:

ЦОАЩДЕКЩЬНХМКБЫЮЕЮКЩЯЮЕЮЧИ

• ***Расшифрование.***

Требуется с помощью ключа **курьер** расшифровать следующую криптограмму, полученную шифром Виженера:

ЪЬБЧСЯЙЧЯМЧРМЯХИУПКЧААЦРЭЁ

Процесс расшифрования заданной криптограммы приведён в табл. 4.6.

[4] По имени прусского офицера Фридриха В. Казиски, который в 1863 г. предложил статистический метод нахождения длины ключа для шифров многоалфавитной замены

Таблица 4.6

Расшифрование с помощью шифра Виженера

к	у	р	ь	е	р	к	у	р	ь	е	р	к	у	р	ь	е	р	к	у	р	ь	е	р	к	у
Ъ	Ь	Б	Ч	С	Я	Й	Ч	Я	М	Ч	Р	М	Я	Х	И	У	П	К	Ч	А	А	Ц	Р	Э	Ё
п	и	с	ь	м	о	_	д	о	с	т	а	в	л	е	н	о	_	а	д	р	е	с	а	т	у

Таким образом, получен открытый текст:

письмо_доставлено_адресату. ❐

4.3. Порядок выполнения работы и варианты заданий

Основные этапы выполнения работы.

Данная практическая работа предполагает выполнение следующих этапов:

1. Изучить методические указания по самостоятельной работе.

2. Выполнить шифрование открытого текста (табл. 4.7) методом перестановки с фиксированным периодом, используя заданный ключ (табл. 4.8). С помощью того же ключа расшифровать текст из табл. 4.8.

3. Осуществить шифрование предложенного открытого текста (табл. 4.8) с помощью вертикальной перестановки, используя для этого ключ из табл. 4.9. С помощью заданного ключа необходимо расшифровать криптограмму из табл. 4.9. Дополнительно требуется написать программу на языке C#, выполняющую шифрование и расшифрование с использованием вертикальной перестановки.

4. Произвести шифрование заданного открытого текста (табл. 4.7) с помощью шифра Виженера. В качестве ключа использовать первые шесть – восемь букв своего ФИО. Используя заданный ключ (табл. 4.10), расшифровать закрытый текст из табл. 4.10, полученный шифром Виженера. Дополнительно разработать программу на языке C#, которая позволяет шифровать и расшифровывать с помощью шифра Виженера.

5. Оформить и защитить отчет по практической работе.

Индивидуальные варианты заданий.

Таблица 4.7

Варианты открытых текстов

№ вар.	Открытый текст (24 симв)
1, 9, 17	контракт_был_аннулирован
2, 10, 18	сегодня_встреча_отменена
3, 11, 19	документы_в_нижнем_ящике
4, 12, 20	проведена_крупная_сделка
5, 13, 21	координаты_не_определены
6, 14, 22	перевод_денег_произведен
7, 15, 23	собрание_будет_в_четверг
8, 16, 24	данные_сильно_повреждены

Таблица 4.8

Исходные данные для шифрования и расшифрования перестановкой с фиксированным периодом

№ вар.	Ключ	Шифротекст
1, 13	(5, 2, 6, 1, 4, 3)	ПЫРБ_Л_КНИЗАГ_УИАШАНД_ЗА
2, 14	(2, 7, 4, 3, 8, 1, 5, 6)	ОАУК_ППКБТЙОСОДЕ_ГОДОЯРО
3, 15	(4, 3, 5, 6, 1, 2)	ЕПХ_УСВЕЗОПР_ЛОЖШЕНАИЯИД
4, 16	(8, 3, 7, 1, 6, 4, 2, 5)	ЕЮЛК_ЧЛИ_ТДЖО_АПМВОККРОИ
5, 17	(3, 6, 2, 5, 1, 4)	САО_РТИНЦ_КЙД__ААВНАУТПК
6, 18	(6, 3, 5, 8, 2, 7, 4, 1)	МУ_СДАТИ_ВЕЕОБЫСДОЯИПКРС
7, 19	(5, 3, 1, 6, 4, 2)	АОПТДРВТЬАО__ЗРПА_НЦОЫЕЛ
8, 20	(5, 4, 7, 2, 1, 8, 3, 6)	В_САВЛМЫП_ДНАКООЕЛИЕРЕПН
9, 21	(3, 5, 2, 1, 6, 4)	ИННОЕ_ОПП_ИДЛ_АСДИООГОРВ
10, 22	(7, 5, 2, 1, 6, 3, 8, 4)	Е_ЕДНЛ_ОИОРПСИТНЛБП_ЫРИИ
11, 23	(6, 3, 1, 5, 2, 4)	ПАЭОТ_ЯАЕИРЦАР_ВПОЬЛЛСИА
12, 24	(3, 4, 7, 1, 8, 2, 6, 5)	ВЫСНОО_ЕУДКТИРИНВОН_ЫУЕЛ

Таблица 4.9

Исходные данные для шифрования и расшифрования с помощью вертикальной перестановки

№ вар.	Таблица	Ключ	Шифротекст
1, 9, 17	3 × 8	(5, 4, 7, 2, 1, 8, 3, 6)	МНЩУ_ЯНЖКО_МДЫЕТНЕКВ_ЕИИ
2, 10, 18	4 × 6	(3, 5, 2, 1, 6, 4)	ОАНЕЕКЯКРНПДПЕУСДР_АВ_АЛ
3, 11, 19	3 × 8	(7, 5, 2, 1, 6, 3, 8, 4)	НОНДЕЛОЫЕКТРИ_ЕО_ДАПЫРНЕ

4, 12, 20	4 × 6	(6, 3, 1, 5, 2, 4)	ОЕИНРДПЕПДГЗВНОЕЕ__ВЕЕРД
5, 13, 21	3 × 8	(3, 4, 7, 1, 8, 2, 6, 5)	РДТНТЕС__ОБЧЕВГИ_РБУЕАЕВ
6, 14, 22	4 × 6	(5, 3, 6, 1, 4, 2)	ЫЬВННИПДЕНРЫД_ОЕНЛОЕАС_Ж
7, 15, 23	3 × 8	(2, 7, 4, 3, 8, 1, 5, 6)	ОБЛКНАТЛРНЫИТННК_УР_ОААВ
8, 16, 24	4 × 6	(4, 3, 5, 6, 2, 1)	ОС_ЕГВАНДТОННРТАЕ_ЧЕСЯЕМ

Таблица 4.10

Исходные данные для расшифрования с помощью шифра Виженера

№ вар.	Ключ	Шифротекст
1, 13	доходный	ЗУАЭГЫ_ИУЯЮЬТЯГЬГЮЕЧЕЗЁТ
2, 14	провести	ЛВОБУАЧЩПЁЧАДАВЧСРЪКРСГД
3, 15	работать	ЮОГИЧ_ГЙВРФТ_ИЭДПУГЭЮЕ_Ц
4, 16	родина	СИПЗЭРЩЩДРМНЩНЬИРУПЬДРНД
5, 17	расход	_ОЬЗЮОР__ЦЭНФЕГЁМГФОБГСТ
6, 18	победа	В_РЙЩ_ЯЯЁЖЛОЖУМДТЖШТБТМЯ
7, 19	открывай	ЩЮЗЖГБЛОХТЭПЙРДИЩАМАГМОЦ
8, 20	компания	ЫЭЮБ_НУФУШЛЭАММАКНЬВНШЫЯ
9, 21	протесты	ШФБДДЮТЛАЯРМЙРУ_А_ЭВГХЭГ
10, 22	торгаш	БЯЯЖАЙННВСВШВНШГ_ЖАЪЁЗНТ
11, 23	приказ	СРХЙВВАЬИШОЖЯЯМХРМЯЬНШИМ
12, 24	клиент	ЩЩСДЫЧЙЫЧИЭЫЬЛФНМЦЩОЧЖЬВ

Требования к отчёту.

Отчёт по практической работе должен содержать следующие пункты:

1. Титульный лист с указанием названия работы, фамилий и инициалов выполнившего и проверившего работу.
2. Цель и задачи работы.
3. Шифрование и расшифрование перестановкой с фиксированным периодом.
4. Шифрование и расшифрование методом вертикальной перестановки.
5. Шифрование и расшифрование с помощью шифра Виженера.

4.4. Контрольные вопросы и задачи

Теоретические вопросы.

1. Что называют шифром?
2. В чём заключается различие между расшифрованием и дешифрованием?
3. Что понимают под ключом в криптографии?
4. Какие шифры называют шифрами перестановки?
5. В чём заключается шифрование перестановкой с фиксированным периодом?
6. Как выполняется шифрование и расшифрование с помощью вертикальной перестановки?
7. Какие шифры называют шифрами замены?
8. Как осуществляется шифрование в шифре Цезаря?
9. Каким образом можно определить ключ для шифра сдвига?
10. Что понимают под шифрами простой замены?
11. Как можно осуществить взлом шифра простой замены?
12. В чём заключаются особенности шифра Виженера?

Практические задачи.

1. Зашифровать сообщение методом вертикальной перестановки по таблице размером 4 × 4 с помощью заданного ключа (табл. 4.11). С помощью этого же ключа расшифровать предложенный шифротекст.

Таблица 4.11

Исходные данные для шифрования и расшифрования с помощью шифра вертикальной перестановки

Вариант	Ключ	Открытый текст	Шифротекст
Нечётный	(2, 1, 3, 4)	пишу_контрольную	АА_ЕЗЧЕШАНЕАД_РН
Чётный	(4, 3, 1, 2)	шифрую_сообщение	НОИЯЖТПСНОРТУ_ОЬ

ТЕМЫ ДЛЯ САМОСТОЯТЕЛЬНОГО ИЗУЧЕНИЯ

5. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ИНФОРМАЦИИ

5.1. Цель и задачи темы

Цель темы – изучить основные понятия теории информации.

Основные задачи:

• освоить составные части системы передачи сообщений;

• ознакомиться с основными количественными оценками неопределенности и информации, используемыми в теории информации;

• изучить связь между вероятностью и количеством информации, а также понятие частного количества информации;

• ознакомиться с понятием кода и кодирования информации, также основными характеристиками и видами кодов.

Тема рассчитана на 4 часа.

5.2. Основные теоретические сведения

Система передачи сообщений и её основные элементы.

Одним из главных объектов исследования в теории информации является ***система передачи сообщений*** (СПС), которая состоит из источника сообщений и их получателя, а также из кодеров и декодеров, разделенных каналом связи (рис. 5.1). В теории информации такая система хорошо исследована с помощью различных математических моделей.

Источник сообщений представляет собой исследуемый или наблюдаемый объект, формирующий сообщения о своем состоянии. Источником сообщений могут быть объекты неживой природы (например, космические тела), люди, технические устройства (например, ЭВМ, датчик, телефон).

В зависимости от вида формируемых сообщений, различают дискретные и непрерывные источники сообщений.

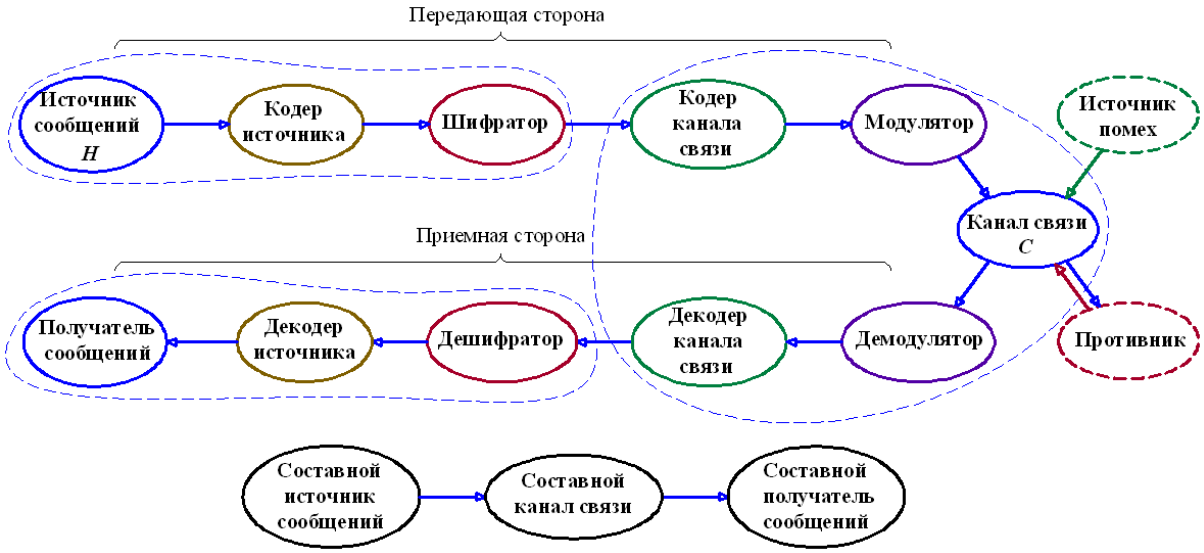


Рис. 5.1. Структура системы передачи сообщений (СПС)

Дискретным источником сообщений называют источник, который может в каждый момент времени случайным образом принять одно из конечного множества возможных состояний. Каждому состоянию источника соответствует условное обозначение в виде символа.

Множество всех символов $S = \{s_1, s_2, ..., s_m\}$, доступных источнику дискретных сообщений, называют ***алфавитом*** этого источника или ***первичным алфавитом***, а общее число символов m – ***объемом алфавита***.

Сообщения, выдаваемые источником, обычно подаются на вход кодера.

Кодер представляет собой устройство, осуществляющее ***кодирование информации***, то есть преобразование информации из формы, удобной для непосредственного ее использования, в форму, удобную для передачи, хранения или автоматической обработки. Обратный процесс называется ***декодированием***.

Кодер имеет собственный алфавит символов $C = \{c_1, c_2, ..., c_K\}$, называемый ***вторичным алфавитом*** (за первичный алфавит принимается алфавит источника сообщений). Кроме того, кодер обладает определенным алгоритмом кодирования (кодом), позволяющим символы первичного алфавита преобразовать в символы вторичного алфавита.

В системе передачи информации выделяют кодер источника и кодер канала связи.

Кодер источника служит для представления сообщений от источника в наиболее компактной форме. Это нужно, чтобы максимально эффективно использовать ресурсы канала связи либо запоминающего устройства. Кодеры источника устраняют из сообщений избыточность.

Кодер канала служит для представления сообщений в форме, обеспечивающей их защиту от помех при передаче по каналу связи, либо возможных искажений при хранении информации. Кодеры канала вводят в сообщения избыточность по специальным правилам, что позволяет обнаруживать и исправлять ошибки в сообщениях.

Модулятор – устройство, предназначенное для преобразования сообщений в сигналы, которые согласованны с физической природой канала связи (например, радиоволны специальной формы) или средой накопителя информации.

Под модуляцией понимается воздействие на один или несколько параметров передаваемого сигнала. В системах связи в качестве несущего сигнала обычно используют синусоидальные колебания или периодические последовательности импульсов.

Под ***каналом связи*** понимают техническое устройство или физическую среду, обеспечивающую поступление сообщений от источника к получателю.

Сигналы на выходе канала связи могут отличаться от переданных вследствие затухания и воздействия помех.

В качестве канала связи может выступать устройство для хранения информации. В этом случае передача информации осуществляется не в пространстве, а во времени.

Аналогично источникам сообщений каналы связи разделяют на непрерывные и дискретные.

Передаваемая через канал связи информация может быть перехвачена противником. Под противником понимается любой субъект, не имеющий права ознакомления с содержанием передаваемой информации. Кроме того, противник может умышленно менять содержание сообщений.

Для предотвращения прочтения сообщения или его замены в систему передачи сообщений вводят шифраторы.

Шифратор является устройством, которое преобразовывает сообщения в форму, обеспечивающую защиту от прочтения этих сообщений противником. Данное преобразование (шифр) должно быть обратимым. Шифры являются объектом исследования криптографии.

Остальные блоки системы, расположенные на приемной стороне, выполняют обратные операции и представляют информацию в удобном для использования виде.

Количественные оценки неопределенности и информации. Формула Хартли.

Первичными неопределяемыми понятиями теории информации являются неопределенность и информация. В теории информации вводятся количественные оценки неопределенности и информации.

Если состояние системы (источник сообщений) известно заранее, то нет смысла передавать сообщение. Сообщение приобретает смысл только тогда, когда состояние системы заранее неизвестно. В этом случае система может случайным образом оказаться в том или ином состоянии, то есть системе присуща какая-то степень неопределенности. Очевидно, что сведения, получаемые о системе, будут тем содержательнее (ценнее, информативнее), чем больше была неопределенность системы до получения этих сведений (априори).

Приобретение информации сопровождается уменьшением неопределенности. При этом количество информации I в сообщении измеряют количеством исчезнувшей неопределенности H (*энтропии*):

$$\boldsymbol{I = H_{pr} - H_{ps}}; \qquad (5.1)$$

где индекс ***pr*** означает «априори», а ***ps*** – «апостериори»; $\boldsymbol{H_{pr}}$ – априорная энтропия (до получения сообщения); $\boldsymbol{H_{ps}}$ – апостериорная энтропия (после получения сообщения).

Рассмотрим, что представляет собой количество неопределенности. При этом возникает вопрос: Что значит «большая» или «меньшая» неопределенность и как ее можно оценить?

❒ ***Пример* 5.1.** Рассмотрим две системы, которым присуща неопределенность:

1. Опыт с подбрасыванием монеты (два состояния).

2. Опыт с подбрасыванием игральной кости (шесть состояний).

Требуется выяснить, какая система обладает большей неопределенностью. Очевидно, что вторая, так как у нее больше возможных состояний, в каждом из которых она может оказаться с одинаковой вероятностью.

Таким образом, степень неопределенности зависит от числа состояний системы (возрастает с увеличением числа состояний). В действительности неопределённость зависит не только от числа состояний. ❒

❒ ***Пример* 5.2.** Пусть в качестве системы выступает некоторое техническое устройство, которое может находиться в двух состояниях:

• исправно (вероятность 0,99);

• неисправно (вероятность 0,01).

Такая система обладает очень малой степенью неопределенности: почти наверное можно предугадать (т. е. до получения сообщения), что устройство будет работать исправно. Но при соотношении 0,55 (исправно) и 0,45 (неисправно) неопределенность будет на много большей. ❒

Таким образом, количество неопределенности системы характеризуется не только числом ее возможных состояний, но и вероятностями нахождения системы в этих состояниях.

В качестве меры неопределенности системы в теории информации применяется специальная характеристика, называемая информационной энтропией или просто энтропией.

Разберемся с тем, что понимается под количеством информации. Следует различать понятия такие понятия, как «количество информации» и «объем информации». Понятие количества

информации связано со снятием неопределенности, которая существовала до получения сообщения.

Количество информации I (в сообщении) – это числовая характеристика, отражающая ту степень неопределенности, которая исчезает после получения данного сообщения. Указанную степень неопределенности в теории информации называют энтропией, но об энтропии поговорим. Чем большая неопределенность существовала до получения сообщения, тем большее количество информации мы получаем в принятом сообщении.

В 1928 г. американским инженером Робертом Хартли была предложена логарифмическая мера количества информации.

Число сообщений N, которое можно получить, комбинируя m символов алфавита по n элементов в сообщении можно определить по формуле

$$N = m^n. \qquad (5.2)$$

Хартли предложил в качестве меры количества информации принять логарифм числа возможных последовательностей символов:

$$I = \log N = \log m^n = n \cdot \log m \qquad (5.3)$$

Последняя формула называется ***формулой Хартли***. Основание логарифма зависит от выбранной единицы количества информации.

Разумной мерой информации, содержащейся в сообщении, является мера, монотонно связанная с затратами на передачу или хранение сообщения. Поскольку эта мера определяет затраты, то она должна быть неотрицательной.

Возникает вопрос: какое основание логарифма следует использовать? Ответ на этот вопрос определяется лишь соображениями удобства, поскольку все логарифмы пропорциональны друг другу:

$$\log_a x = \log_b x / \log_b a = (\log_a b) \log_b x. \qquad (5.4)$$

Выбор основания a логарифма $\log_a x$ определяет единицу измерения количества информации:

• $a = 2 \rightarrow$ бит – сокращение английских слов *binary digit* (двоичная единица);

• $a = 10 \rightarrow$ дит или хартли (в честь Р. Хартли);

• $a = e \approx 2{,}718 \rightarrow$ нат.

Удобнее всего использовать логарифмы по основанию 2, поскольку получающая единица является битом. В дальнейшем будем использовать именно это основание при записи логарифмов.

Наряду с понятием количества информации используется понятие объем информации.

Объем информации V (в сообщении) измеряется количеством символов вторичного алфавита (то есть кодовых разрядов), содержащихся в данном сообщении.

$$\boldsymbol{V} = n\,k. \qquad (5.4)$$

Например, в коде ASCII для кодирования текстового символа (буквы, цифры, знака препинания и др.) используется 8 двоичных разрядов (бит), образующие вместе единицу объема, называемую байт.

В отличие от объема количество информации определяется относительно первичного алфавита и зависит от его вероятностных характеристик.

Количество информации и вероятность. Частное количество информации.

В теории информации для оценки количества информации используется вероятностный подход.

Определим, как связаны между собой количество информации I и вероятность p некоторого события.

Пусть задан алфавит источника сообщений, состоящий из символов $s_1, s_2, \ldots, s_m$, которые имеют следующие вероятности появления в сообщении: $p_1, p_2, \ldots, p_m$.

Сколько информации поступает при приеме одного из этих символов? Если, например, $p_1=1$ (соответственно для остальных символов вероятность появления равна нулю), то никакой неопределенности нет, и никакой информации мы не получим, поскольку заранее известно каким будет сообщение. Если вероятности появления символов различаются, то появление маловероятного символа является более неожиданным и при этом получается большее количество информации, чем при появлении высоковероятного символа.

• Таким образом, количество информации $I(s)$ от наступления события s обратно вероятности $p(s)$ этого события:

$$I(s) \sim f\left(\frac{1}{p(s)}\right).$$

• Представляется естественным, что количество информации от двух разных независимых символов равно сумме количеств информации от каждого из них в отдельности (удивление аддитивно):

$$I_{1,2} = I_1 + I_2.$$

• Кроме того, вероятность составного события $p(s_1, s_2)$ равна произведению вероятностей p_1 и p_2 двух независимых событий:

$p_{1,2} = p_1 \cdot p_2$ или

$1/p_{1,2} = 1/(p_1 \cdot p_2)$.

Если прологарифмировать последнее выражение, то получим:

$$\log_2[1/p_{1,2}] = \log_2[1/(p_1 \cdot p_2)] = \log_2(1/p_1) + \log_a(1/p_2).$$

Отсюда можно сделать вывод, что:

$I_1 = \log_2(1/p_1)$, $I_2 = \log_2(1/p_2)$;

$I_{1,2} = I_1 + I_2 = \log_2[1/(p_1 \cdot p_2)] = -\log_2(p_1 \cdot p_2)$.

Последняя формула показывает, что произведение вероятностей соответствует сумме количеств информации. Такая зависимость согласуется с представлениями о том, чем должна быть информация.

Частное (собственное) ***количество информации***:

$$I(s_1) = -\log_2 p(s_1), \text{ бит} \qquad (5.6)$$

Частное количество информации характеризует «информативность» или «степень неожиданности» конкретного сообщения.

❒ ***Пример*** **5.3.** Пусть требуется узнать, сдал или не сдал экзамен некоторый студент, и определить количество информации от наступления события.

Примем следующие вероятности этих двух событий:

P(сдал) = 7 / 8;

P(не сдал) = 1 / 8.

Видно, что студент является довольно сильным.

Если кто-нибудь сообщит, что студент сдал экзамен, то адресат вправе сказать: «Ну и что? Я и без этого знал, что он сдаст». Количество информации в этом сообщении будет:

$\boldsymbol{I}$(сдал) = $\log_2(8/7)$ = 0,193 бит.

Если сообщат, что данный студент экзамен не сдал, то получатель скажет: «Неужели?» и сильно обогатится знаниями. Количество информации здесь будет:

I(не сдал) = $\log_2 8$ = 3 бит. ❒

Из определения частного количества информации и свойств логарифма непосредственно вытекают следующие свойства частного количества информации:

- ***Неотрицательность***: $I(s_i) \geq 0$;
- ***Монотонность***: если $s_1, s_2 \in S$, $p(s_1) \geq p(s_2)$, то $I(s_1) \leq I(s_2)$;
- ***Аддитивность***: $I(s_1, s_2) = I(s_1) + I(s_2)$.

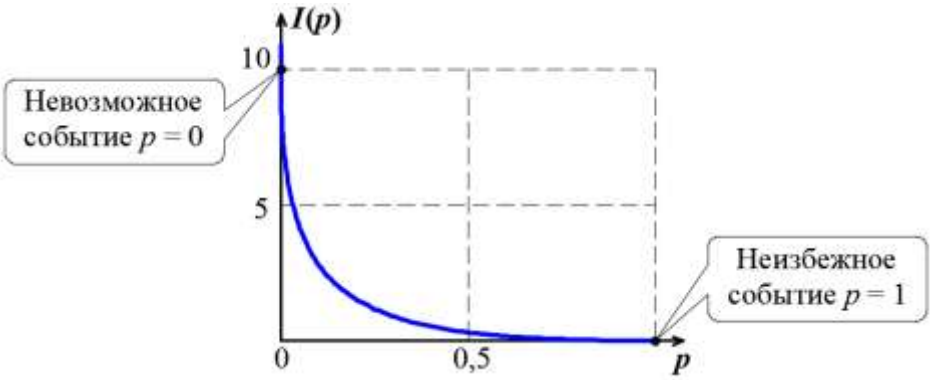


Рис. 5.2. Зависимость частного количества информации $I(p)$ от вероятности

Понятие кода. Характеристики и виды кодов.

В настоящее время отсутствует общепринятая трактовка, что считать кодом.

В самом общем случае кодом можно считать язык, который представляет собой совокупность звуковых сигналов и графиче-

ских символов, а также является средством общения людей владеющих этим языком. В такой трактовке кодом можно назвать любой язык, письменность, общепринятые обозначения. Например, к кодам можно отнести дорожные знаки.

В качестве рабочего определения кода можно использовать следующее:

Код – правило сопоставления каждому конкретному сообщению (символу первичного алфавита) строго определённой комбинации символов вторичного алфавита. Отдельная комбинация таких символов называется ***кодовым словом***. Иногда используют термины «кодовая комбинация», «кодовая последовательность» или «кодограмма».

Основными целями кодирования информации являются:

• преобразование информации в форму, пригодную для её автоматической обработки с помощью технических средств;

• эффективное использование канала связи; уменьшение стоимости передачи и хранения; уменьшение избыточности;

• защита от помех в канале связи; повышение помехоустойчивости и достоверности передачи сообщений;

• криптографическая защита информации; сокрытие смысла передаваемой информации от непосвящённых лиц.

К основным характеристикам кодов относятся:

• ***Основание кода*** – объем алфавита K кодера, то есть число различных символов кода. Например, если $K = 2, 3, 4, \ldots$, то код называют бинарным (двоичным), триарным (троичным), тетрарным и т. д.

• ***Длина кодового слова*** (разрядность) – число символов L в кодовом слове.

Коды, все слова которых имеют одинаковую длину, называются ***равномерными*** (блоковыми). Соответственно коды, в которых данное условие не выполняется, называют ***неравномерными***. Классическим примером равномерного кода является пятизначный двоичный код Бодо, используемый в телеграфной связи. К неравномерным кодам относится код Морзе.

Неравномерные коды требуют либо специальных разделительных знаков, учитывающих конец одного кодового слова и начало другого, либо должны строиться так, чтобы никакое кодо-

вое слово не являлась передней частью (префиксом) другого кодового слова (***свойство префикса***).

Коды, обладающие свойством префикса называют ***префиксными*** кодами. К префиксным кодам относятся коды Шеннона-Фано и Хаффмана. Равномерные коды являются префиксными.

• ***Общее число кодовых слов*** *N* (мощность кода).

Если все кодовые слова используются для кодирования сообщений, то такой код называется ***полным***.

Для равномерного кода общее число кодовых слов определяется следующим образом:

$$N = K^L. \qquad (5.6)$$

• ***Взвешенность кода*** – соответствие символов кода весовым коэффициентам системы счисления.

Если все кодовые комбинации кода соответствуют числам выбранной системы счисления, то такие коды ***называются взвешенными*** (арифметическими). В противном случае коды называют ***невзвешенными***.

Взвешенные коды широко используются для передачи и обработки числовой информации. Примером взвешенного кода является натуральный двоичный код (НДК), в котором кодовые комбинации соответствуют последовательности натуральных чисел. Примером невзвешенного кода является двоичный код Грэя.

По назначению коды можно разделить на следующие группы:

• ***Коды представления данных***, которые обеспечивают запись данных в требуемом для автоматической обработки формате.

• ***Эффективные*** (оптимальные) ***коды***, позволяющие сжимать передаваемые сообщения и тем самым более эффективно использовать канал связи.

• ***Помехоустойчивые коды***, которые обеспечивают обнаружение и исправление ошибок в полученных сообщениях.

• ***Шифры***, служащие для защиты передаваемой информации от прочтения противником.

Классификация кодов представлена на рис. 5.3.

Рис. 5.3. Классификация кодов

5.3. Контрольные вопросы

1. Что в теории информации понимают под системой передачи сообщений?
2. Какие источники сообщений называют дискретными?
3. Что такое кодирование и декодирование информации?
4. Какие функции выполняют кодеры источника сообщений и канала связи?
5. Какую часть системы передачи сообщений называют каналом связи?
6. Как определяется количество информации в дискретном сообщении по формуле Хартли?
7. В чём заключается различие между объёмом и количеством информации?
8. Что характеризует и как определяются частное количество информации?
9. Что понимают под кодом?
10. Для решения каких задач применяют кодирование информации?
11. Что относят к основным характеристикам кодов?
12. Какие выделяют виды кодов по назначению?

6. ИНФОРМАЦИОННЫЕ ХАРАКТЕРИСТИКИ ДИСКРЕТНЫХ ИСТОЧНИКОВ СООБЩЕНИЙ

6.1. Цель и задачи темы

Цель темы – изучить основные информационные характеристики источников сообщений.

Основные задачи:

• ознакомиться с понятиями статистических связанных источников сообщений и условной энтропии, а также рассмотреть свойства условной энтропии;

• изучить понятие объединения источников сообщений и энтропии объединения; рассмотреть свойства энтропии объединения;

• ознакомиться с понятием взаимной информации, а также эффективностью, избыточностью и производительностью источника сообщений.

Тема рассчитана на 4 часа.

6.2. Основные теоретические сведения

Статистически связанные источники сообщений. Условная энтропия и её свойства.

При определении энтропии часто необходимо учитывать статистические связи, которые в большинстве случаев имеют место как между состояниями двух или нескольких источников сообщений, объединенных в рамках одной системы. Кроме того статистические связи существуют между состояниями, последовательно выбираемыми одним источником (связи между символами в сообщениях). Например, в текстах на русском языке вероятность появления мягкого знака после гласной буквы равна нулю. В этом случае речь ведут об условной энтропии.

При рассмотрении взаимодействующих систем состояние одной из них влияет на состояние другой, как состояние моря и скорость ветра влияют на скорость корабля. В таких случаях энтропия не может быть определена на основе безусловных вероятностей.

Пусть заданы два дискретных источника сообщений, характеризуемые ансамблями A и B (для краткости они будут называться «источники A и B»):

$$A=\begin{pmatrix} a_1 & a_2 & \dots & a_{m_A} \\ p(a_1) & p(a_2) & \dots & p(a_{m_A}) \end{pmatrix}; \quad B=\begin{pmatrix} b_1 & b_2 & \dots & b_{m_B} \\ p(b_1) & p(b_2) & \dots & p(b_{m_B}) \end{pmatrix}.$$

Между состояниями двух дискретных источников A и B сообщений могут существовать статистические связи. В этом случае вероятности $p(b_j|a_i)$ перехода одного источника B (зависимого) в определенное состояние зависит от того, в каком состоянии находится источник A (независимый) (рис. 6.1).

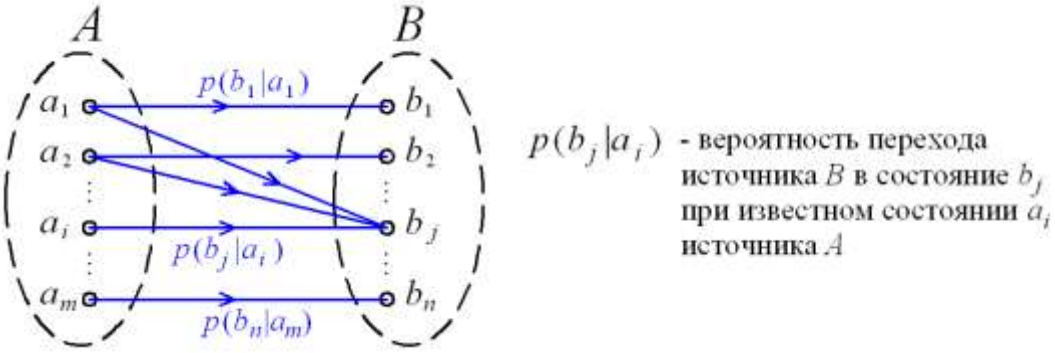


Рис. 6.1. Граф переходов состояний источника сообщений B относительно состояний источника A

Для задания статистической связи между состояниями источников используется ***матрица переходных вероятностей***:

$$P_{mn}=\begin{bmatrix} p(b_1|a_1) & p(b_2|a_1) & \dots & p(b_n|a_1) \\ p(b_1|a_2) & p(b_2|a_2) & \dots & p(b_n|a_2) \\ \dots & \dots & \dots & \dots \\ p(b_1|a_m) & p(b_2|a_m) & \dots & p(b_n|a_m) \end{bmatrix}.$$

Сумма элементов всех элементов любой строки матрицы P_{mn} должна быть равна 1, поскольку текущее состояние источника A должно обязательно перейти в какое-либо состояние источника B:

$$\sum_{j=1}^{m} p(b_j \mid a_i) = 1 \quad (i = 1, 2, \ldots, m).$$

Если существует статистическая связь между состояниями разных источников, а также состояниями одного источника, то энтропия не может быть определена только на основе безусловных вероятностей.

Понятие условной энтропии основывается на понятии условной вероятности, которое дает общую характеристику связи между событиями.

Если заданы два события ***a*** и ***b***, то вероятность события ***b*** при условии, что наступило событие ***a***, будет условной вероятностью $p(\boldsymbol{b}|\boldsymbol{a})$, которую можно найти по формуле

$$p(\boldsymbol{b}|\boldsymbol{a}) = p(\boldsymbol{a},\boldsymbol{b}) / p(\boldsymbol{a}).$$

где $p(\boldsymbol{a},\boldsymbol{b})$ – вероятность того, что произойдет как событие A_1, так и событие A_2 (совместная вероятность). Приведенная формула имеет смысл лишь при $p(\boldsymbol{a}) > 0$.

Условная и совместная вероятности обладают следующими свойствами:

1. При отсутствии связи между событиями ***a*** и ***b*** (независимые события):

$$p(\boldsymbol{b}|\boldsymbol{a}) = p(\boldsymbol{b});$$

$$p(\boldsymbol{a},\boldsymbol{b}) = p(\boldsymbol{a})p(\boldsymbol{b}).$$

2. При жесткой связи между событиями ***a*** и ***b*** (полностью зависимые события):

$$p(\boldsymbol{b}|\boldsymbol{a}) = 1;$$

$$p(\boldsymbol{a},\boldsymbol{b}) = p(\boldsymbol{a}).$$

Для оценки количества информации в статистически зависимых событиях вводят понятие условной энтропии.

Для любого состояния a_i можно найти распределение условных вероятностей $p(b_j|a_i)$ на множестве состояний B и для каждого b_j можно подсчитать частное количество информации

$$I(b_j|a_i) = -\log_2 p(b_j|a_i), \qquad (6.1)$$

которое называется ***частным условным количеством информации*** от получения символа $\boldsymbol{b}_j$ при известном $\boldsymbol{a}_i$. Данная величина является случайной, поскольку зависит от a_i *и* b_j.

Усредним количество информации $I(b_j|a_i)$ по алфавиту B.

При известном символе a_i из источника A энтропия источника B будет:

$$H(B \mid a_i) = -\sum_{j=1}^{n} p(b_j \mid a_i) \log_2 (b_j \mid a_i). \qquad (6.2)$$

Энтропия $H(B|a_i)$ называется ***частной условной энтропией*** источника B относительно состояния a_i источника A.

Вновь введенная энтропия является случайной величиной, поскольку она зависит от случайной переменной a_i. Чтобы получить неслучайную информационную характеристику для пары ансамблей, необходимо выполнить усреднение по всем значениям a_i.

Условная энтропия источника B относительно источника A может быть определена путем усреднения частных условных энтропий $H(B|a_i)$ по всему источнику A следующим образом:

$$\begin{aligned} H(B \mid A) &= \sum_{i=1}^{m} p(a_i) H(B \mid a_i) = \\ &= -\sum_{i=1}^{m} \sum_{j=1}^{n} p(a_i)\, p(b_j \mid a_i) \log_2 p(b_j \mid a_i). \end{aligned} \qquad (6.3)$$

Условная энтропия $H(B|A)$ источника сообщений B относительно источника A характеризует степень неопределенности состояния источника B при известном состоянии из источника A.

Поскольку $p(a_i) \cdot p(b_j|a_i) = p(a_i,b_j)$, то формулу для условной энтропии можно записать следующим образом:

$$H(B \mid A) = -\sum_{i=1}^{m} \sum_{j=1}^{n} p(a_i, b_j) \log_2 p(b_j \mid a_i). \qquad (6.4)$$

Условная энтропия обладает следующими свойствами:

1. Имеет место неравенство:

$$H(B|A) \le H(B); \qquad (6.5)$$

то есть условная энтропия не может превосходить безусловной. Это согласуется с интуитивным представлением о том, что

знание об источнике сообщений A, может только уменьшить неопределенность в сообщении от источника B, а если они независимы, то оставит ее неизменной. То есть степень неопределенности системы не может увеличиться от того, что состояние какой-то другой системы стало известным.

2. Если источники сообщений A и B статистически независимы, то условная энтропия $H(B|A)$ источника B относительно источника A равна энтропии (безусловной) $H(B)$.

$$H(B|A) = H(B);\ H(A|B) = H(A).$$

Данное свойство можно доказать из свойства условной вероятности$\cdot p(b_j|a_i) = p(b_i)$ для независимых событий.

3. Если источники сообщений B и A являются статистически жестко связанными, то есть появление сообщения от одного из них непременно подразумевает появление сообщения от другого, то условная энтропия источника B относительно источника A равна нулю:

$$H(B|A) = 0.$$

Справедливость данного утверждения вытекает из следующего свойства условных вероятностей: при полной статистической зависимости $p(b_j \mid a_i) = 1$. При таких значениях все слагаемые в формуле для условной энтропии обращаются в ноль.

Понятие условной энтропии в теории информации используется при определении взаимозависимости между символами кодируемого алфавита, для определения потерь при передаче информации по каналам связи, при вычислении энтропии объединения.

Объединение дискретных источников сообщений. Энтропия объединения и её свойства.

На практике часто приходится определять энтропию для сложной системы, полученной объединением двух или более простых систем.

Под ***объединением*** двух дискретных источников сообщений A и B с возможными состояниями $a_1, a_2, ..., a_m; b_1, b_2, ..., b_n$ понимается сложный источник (A,B), состояния которого представля-

ют все возможные комбинации состояний (a_i, b_j) источников A и B.

Очевидно, число возможных состояний системы (A,B) будет $m \times n$.

$$(A,B) = \begin{pmatrix} (a_1,b_1) & (a_2,b_1) & \dots & (a_i,b_j) & \dots & (a_m,b_n) \\ p(a_1,b_1) & p(a_2,b_1) & \dots & p(a_i,b_j) & \dots & p(a_m,b_n) \end{pmatrix}.$$

Обозначим через $p(a_i, b_j)$ вероятность того, что система (A,B) будет в состоянии (a_i, b_j).

На основе свойств условной и совместной вероятностей можно записать

$$p(a_i, b_j) = p(a_i) \cdot p(b_j \mid a_i);$$
$$p(a_i, b_j) = p(b_j) \cdot p(a_i \mid b_j).$$

Для источников сообщений A и B ***энтропия объединения*** (взаимная энтропия) представляет собой сумму вида:

$$H(A,B) = -\sum_{i=1}^{m} \sum_{j=1}^{n} p(a_i,b_j) \log_2 p(a_i,b_j); \qquad (6.6)$$

где $p(a_i,b_j)$ – вероятность совместного появления в сообщениях символов a_i и b_j из источников A и B.

Энтропия объединения $H(A,B)$ источников сообщений A и B характеризует неопределенность того, что система (A,B) находится в состоянии (a_i, b_j).

Энтропия объединения обладает следующими свойствами:

1. Если две системы A и B объединяются в одну, то энтропия объединенной системы (A,B) равна энтропии одной из ее составных частей плюс условная энтропия второй части относительно первой:

$$H(A,B) = H(A) + H(B|A) = H(B) + H(A|B).$$

Формула (6.7) доказывается из формулы для совместной вероятности $p(a_i,b_j) = p(a_i) \cdot p(b_j|a_i) = p(b_j) \cdot p(a_i|b_j)$ при подстановке в формулу для энтропии объединения.

2. Если источники сообщений A и B статистически независимы, то

$$H(A,B) = H(A) + H(B).$$

Данное свойство вытекает из свойства условной энтропии, которая в рассматриваемом случае будет $H(B|A) = H(B)$, $H(A|B) = H(A)$.

В этом случае энтропия объединения максимальна.

В общем случае (при наличии статистической связи)

$$H(A,B) \le H(A) + H(B).$$

Соотношение следует из того, что условная энтропия не может превосходить безусловной: $H(A|B) \le H(B)$.

Совместная неопределенность снижается, так состояние одного источника позволяет заранее предполагать состояние другого источника. Снижение неопределенности означает обмен информацией между источниками.

3. Если источники сообщений A и B являются полностью статистически зависимыми, то

$$H(A,B) = H(A) = H(B).$$

Данное свойство также вытекает из свойства условной энтропии, которая равна нулю при полной статистической связи.

Энтропия объединения и условная энтропия дискретных источников A и B при наличии статистической связи между их состояниями связаны между собой соотношениями:

$$H(A,B) = H(A) + H(B|A) = H(B) + H(A|B); \qquad (6.7)$$

$$H(B|A) = H(A,B) - H(A);$$

$$H(A|B) = H(A,B) - H(B).$$

Взаимная информация и её свойства. Эффективность, избыточность и производительность источника сообщений.

При определении количества информации о системе A, предполагается, что наблюдение ведется непосредственно за самой системой A. На практике часто бывает так, что система A непосредственно недоступна для наблюдения, и выясняется состояние не самой системы A, а некоторой другой системы B, связанной с нею.

В том случае, когда интересующая система A и наблюдаемая B различны, возникает вопрос: какое количество информации о системе A дает наблюдение за системой B? Это количество ин-

формацию можно естественно определить как уменьшение энтропии системы A в результате получения сведений о состоянии системы B:

$$I(A,B) = H(A) - H(A|B), \text{ (бит/символ)}$$

До получения сведений о системе A энтропия системы A была $H(A)$ (априорная энтропия); после получения сообщения энтропия стала $H(A|B)$. Уничтоженная сообщением энтропии и есть количество информации $I(A,B)$, называемое взаимной информацией.

Взаимной информацией, содержащейся в источниках A и B, называется величина $I(A,B)$, которая характеризует среднее количество информации, получаемое от одного символа источника A при одном переданном символе источника B.

Взаимную информацию можно определить следующим образом:

$$I(A,B) = H(B) - H(B|A) = H(A) - H(A|B). \qquad (6.8)$$

Подставляя выражение (6.7) в формулу (6.8), получим:

$$I(A,B) = H(A) + H(B) - H(A,B). \qquad (6.9)$$

То есть взаимная информация, содержащаяся в двух системах, равна сумме энтропий систем минус энтропия объединенной системы.

На основе полученных зависимостей можно вывести общее выражение для взаимной информации:

$$H(A) = -\sum_{i=1}^{m} p(a_i)\log_2 p(a_i);$$

$$H(B) = -\sum_{j=1}^{n} p(b_j)\log_2 p(b_j);$$

$$H(A,B) = -\sum_{i=1}^{m}\sum_{j=1}^{n} p(a_i,b_j)\log_2 p(a_i,b_j).$$

После подстановки выражений для энтропий $H(A)$, $H(B)$ $H(A,B)$ в формуле (6.9) и проведения преобразований получим следующую формулу для нахождения взаимной информации:

$$I(A,B) = \sum_{i=1}^{m}\sum_{j=1}^{n} p(a_i,b_j)\log_2 \frac{p(a_i,b_j)}{p(a_i)\,p(b_j)}. \qquad (6.10)$$

Свойства средней взаимной информации:

1. Симметричность $I(A;B) = I(B,A)$.

2. Неотрицательность $I(A,B) \geq 0$.

3. Если источники A и B независимы, то $I(A,B) = 0$.

4. При жесткой статистической связи между источниками A и B

$$I(A,B) = H(A) = H(A).$$

Связь между понятиями условной энтропии, энтропии объединения и взаимной информации показана на рис. 6.2.

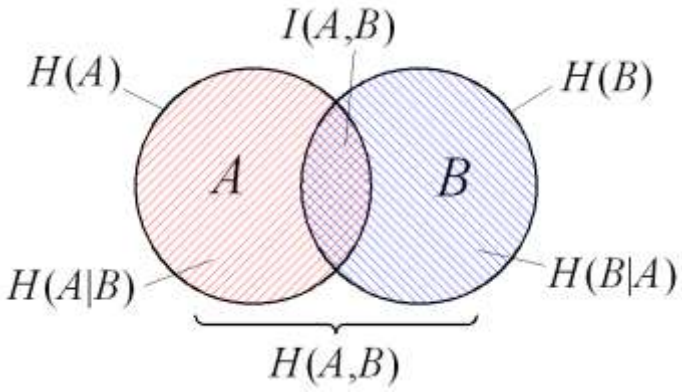


Рис. 6.2. Связь между условной энтропией, энтропией объединения и взаимной информацией

Важными информационными характеристиками источников сообщений являются эффективность и избыточность.

Если энтропия H источника сообщений не равна максимальной энтропии $H_{\max} = \log_2 m$, то это прежде всего означает, что сообщения данного источника могли бы нести большее количество информации.

Эффективность $E(S)$ источника сообщений S называется величина, определяеемая следующим образом:

$$E(S) = \frac{H(S)}{H_{\max}}; \qquad (6.11)$$

где $H_{\max} = \log_2 m$ – максимально возможная для данного источника сообщений энтропия, которая достигается при равной вероятности появления символов.

Избыточностью $R(S)$ источника сообщений S, называется величина, которая показывает относительную недогруженность информацией на символ алфавита источника. Данная характеристика может быть определена следующим образом:

$$R(S) = 1 - \frac{H(S)}{H_{\max}} = 1 - E(S); \qquad (6.12)$$

Кроме общей информационной избыточности существуют частные избыточности, основными из которых являются:

• Избыточность, вызванная статистической связью между символами s_1, s_2, ..., s_m:

$$R_p = 1 - \frac{H(S)}{H_1(S)}; \qquad (6.13)$$

где $H_1(S) = -\sum_{i=1}^{m} p(s_i)\log_2 p(s_i)$ – энтропия источника сообщений при отсутствии статистической связи между символами.

• Избыточность, вызванная неэкстремальным распределением символов s_1, s_2, ..., s_m (неравновероятным появлением символов в сообщениях):

$$R_\varphi = 1 - \frac{H_1(S)}{H_{\max}}. \qquad (6.14)$$

Общая информационная избыточность связана с частными избыточностями следующим образом:

$$R(S) = R_p + R_\varphi - R_p R_\varphi. \qquad (6.15)$$

Если частные избыточности малы, то принимают $R_p R_\varphi = 0$ и

$$R = R_p + R_\varphi.$$

Информационная избыточность не всегда является вредной. При передаче сообщений в условиях действия помех избыточность может быть использована для повышения помехоустойчивости передаваемых сообщений. Например, в сообщении «Весна пришла» можно исключить две последние буквы и передать «Весна приш», но из-за помех может получиться искажение, например, «Весна прошла». Для гарантии надо передать «Весна пришла, но еще не кончилась».

Под ***производительностью*** $P(S)$ источника сообщений понимают количество информации, вырабатываемое источником в единицу времени. Эту характеристику источника называют также скоростью создания сообщений. Производительность источника сообщений равна энтропии источника, приходящейся на единицу времени:

$$P(S) = \frac{H(S)}{\bar{\tau}}; \quad \text{(бит/с)} \qquad (6.16)$$

где $\bar{\tau}$ – средняя длительность выдачи источником одного символа.

Если τ_i – длительность выдачи символа s_i, то значение $\bar{\tau}$ может быть найдено следующим образом:

$$\bar{\tau} = \sum_{i=1}^{m} p(s_i)\tau(s_i).$$

6.3. Контрольные вопросы

1. Что характеризует и как определяется условная энтропия одного дискретного источника сообщений относительно другого?
2. В каком случае условная энтропия равна нулю?
3. Что характеризует и как определяется энтропия объединения двух дискретных источников сообщений?
4. Чему равна энтропия объединения источников сообщений при их полной статистической зависимости?
5. Как связана энтропия объединения и условная энтропия двух статистически зависимых источников сообщений?
6. Что характеризует взаимная информация источников сообщений?
7. Что характеризует информационная избыточность источника сообщений?
8. Какие выделяют частные виды информационной избыточности и как они связаны с общей избыточностью источника сообщений?

7. ИНФОРМАЦИОННЫЕ ХАРАКТЕРИСТИКИ ДИСКРЕТНЫХ КАНАЛОВ СВЯЗИ

7.1. Цель и задачи темы

Цель темы – изучить информационные характеристики дискретных каналов связи.

Основные задачи:

• ознакомиться с основными видами помех в канале связи, а также с определением информационных потерь при передаче информации через дискретный канал связи с помехами;

• изучить понятие пропускной способности канала связи и методы определения пропускной способности для дискретного канала связи без памяти.

Тема рассчитана на 4 часа.

7.2. Основные теоретические сведения

Помехи в канале связи. Потери информации при передаче через канал с помехами.

Помехи являются мешающими передаче сигналов факторами и приводят к ошибкам в принятых сообщениях. Степень воздействия помехи определяется соотношением мощностей сигнала и помехи.

По характеру воздействия на сигнал различают аддитивные и мультипликативные помехи.

При аддитивных помехах передаваемый сигнал $\boldsymbol{x(t)}$ суммируется с помехой $\boldsymbol{e_a(t)}$ (рис. 7.1):

$$\boldsymbol{y(t) = x(t) + e_a(t).}$$

При мультипликативной помехе производится умножение:

$$\boldsymbol{y(t) = x(t) \times e_m(t).}$$

Наиболее распространенной моделью непрерывного канала связи с аддитивной помехой является гауссовый канал. Помеха в таком канале представляет непрерывный случайный процесс с нормальной плотностью вероятности. Данный вид помех называется аддитивным белым гауссовским шумом.

Рис. 7.1. Принцип работы дискретного канала с аддитивными ошибками

Каналы связи, в которых присутствует только мультипликативная помеха, называются каналами с релеевскими замираниями. Плотность величины помехи в таких каналах определяется релеевским распределением.

В канале связи без помех количество информации, получаемой с выхода канала связи, совпадает с количеством информации, подаваемой на его вход. То есть в этом случае передача осуществляется без потерь.

При наличии помех в канале связи нарушается соответствие между средним количеством информации, получаемой с выхода канала в единицу времени, и количеством информации на его входе за тот же период времени.

Если канал полностью зашумлен, то никакой передачи информации не происходит (вся передаваемая информация теряется).

Важным является вопрос: сколько информации будет потеряно при передаче через канал связи при заданном уровне помех?

Количество переданной через канал связи информации определяется взаимной информацией:

$$I(X,Y) = H(X) - H(X|Y).$$

Диаграмма потоков информации через канал связи показана на рис. 7.2.

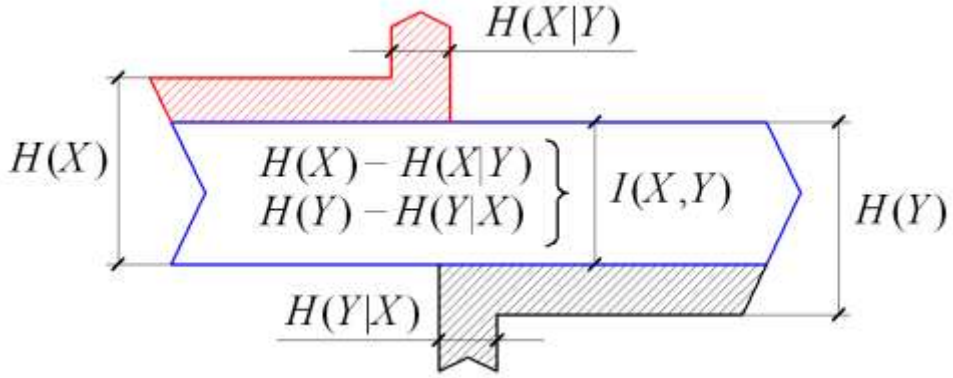


Рис. 7.2. Схема информационного потока через канал связи с помехами

Энтропия $H(X)$ характеризует неопределенность сообщений на входе канала связи.

Энтропия $H(X|Y)$ характеризует неопределенность состояния входа X канала связи при известном состоянии выхода Y, то есть оставшаяся неопределенность приемника. Данная величина определяет информационные потери при передаче одного символа через канал связи.

Энтропия $H(X|Y)$ характеризует неопределенность принятых символов при известных посланных символах. Данная величина определяет среднее количество посторонней «шумовой» информации при передаче одного символа через канал связи.

При отсутствии помех каждому принятому символу y_j соответствует один единственный переданный символ x_i и наоборот (жесткая связь между символами).

В этом случае $H(X|Y) = 0$ и вся передаваемая информация происходит через канал без ошибок, то есть

$I(X,Y) = H(X) – H(X|Y) = H(X)$.

В этом случае матрица переходных вероятностей дискретного канала без памяти представляет собой единичную матрицу $P = E$.

При очень высоком уровне помех (канал полностью зашумлен) любой из принятых символов y_j может соответствовать любому переданному символу x_i (статистическая связь между символами отсутствует). В этом случае $H(X|Y) = H(X)$ и никакой передачи информации не происходит (вся информация теряется), то есть $I(X,Y) = H(X) – H(X|Y) = 0$.

Информационные потери при передаче L символов по каналу связи с помехами можно найти из формулы

$$\Delta I = L \cdot H(X|Y). \qquad (7.1)$$

Отсюда среднее количество принятой информации может быть определено следующим образом:

$$I = L \cdot H(Y) - \Delta I = L[H(Y) - H(X|Y)]. \qquad (7.2)$$

Скорость передачи информации. Пропускная способность дискретного канала связи.

Для описания возможностей канала по передаче информации используют понятие скорости передачи (технической и информационной).

Под ***технической скоростью передачи*** V_T (скорость манипуляции) понимают число элементарных сигналов (символов) передаваемых по каналу в единицу времени. Техническая скорость зависит от свойств линии связи и быстродействия аппаратуры канала и определяется следующим образом:

$$V_T = 1 / \tau_{ср}, \text{ (бод)}; \qquad (7.3)$$

где $\tau_{ср}$ – среднее время передачи одного символа через канал связи.

Единицей измерения технической скорости является бод – скорость, при которой за секунду передается один символ.

Под ***информационной скоростью*** V (скоростью передачи информации) понимают среднее количество информации, которое передается по каналу связи в единицу времени. Информационная скорость зависит от таких характеристик данного канала связи, как объем алфавита используемых символов, технической скорости их передачи, статистических свойств помех в среде передачи, распределения вероятностей символов и их статистической взаимосвязи.

Количество информации о входных символах канала, содержащееся в выходных символах определяется средней взаимной информацией $I(X,Y)$.

При известной технической скорости V_T скорость передачи информации по каналу связи определяется соотношением:

$$V = V_T \cdot I(X,Y), \text{ (бит/с)}; \qquad (7.4)$$

где $I(X,Y)$ – средняя взаимная информация, переносимая одним символом.

Важно знать, до какого предела и каким путем можно повысить скорость передачи информации по конкретному каналу связи. Предельные возможности канала по передаче информации характеризуются его пропускной способностью.

Пропускной способностью C дискретного канала связи называется максимальное значение скорости передачи информации по этому каналу, которой можно достигнуть при самых совершенных способах передачи и приема:

$$C = \max\{V\} = \max\{V_T \cdot I(X,Y)\}, \text{ (бит/с)}; \qquad (7.5)$$

При отсутствии помех имеет место взаимно однозначное соответствие между символами входного X и выходного Y алфавитов. В этом случае взаимная информация будет $I(X,Y) = H(X) = H(X)$.

Максимум энтропии может быть достигнут при равной вероятности символов: $H_{\max} = \log_2 m$.

Отсюда пропускная способность канала связи при отсутствии помех будет

$$C = V_T \cdot \log_2 m. \qquad (7.6)$$

Таким образом, для увеличения скорости передачи информации по дискретному каналу без помех и приближения ее к пропускной способности канала передаваемые символы должны быть равновероятными, а статистические связи между ними должны отсутствовать. Этого можно достичь, если кодировать входные последовательности символов блоками такой длины, при которой вероятности их появления становятся близкими по значению.

Увеличение объема алфавита приводит к повышению пропускной способности канала (рис. 7.3), но при этом усложняется его техническая реализация.

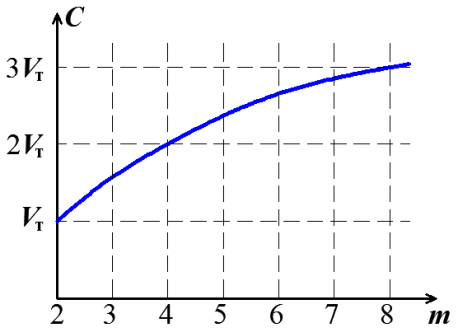


Рис. 7.3. Зависимость пропускной способности канала связи от объёма алфавита

Скорость передачи информации по каналу связи с помехами определяется формулой:

$$V = V_T \cdot I(X,Y) =$$

$$= V_T \sum_{i=1}^{m}\sum_{j=1}^{n} p(x_i, y_j)\log_2 \frac{p(x_i, y_j)}{p(x_i)\,p(y_j)}, \text{ (бит/с).} \qquad (7.7)$$

Считая величину V_T предельно допустимой при заданных технических характеристиках канала, величину $I(X,Y)$ можно максимизировать, изменяя статистические свойства последовательностей символов на входе канала посредством кодера канала связи.

$$C = \max_{\bar{p}(x)}\{V_T \cdot I(X,Y)\}; \qquad (7.8)$$

где $\bar{p}(x)$ – распределение вероятностей входных символов.

Пропускная способность двоичного симметричного канала связи определяется следующим образом:

$$C = V_T\cdot[1 + p\log_2 p + (1-p)\log_2(1-p)] = V_T\cdot[1 - H(p)].$$

7.3. Контрольные вопросы

1. Какие выделяют виды помех в канале связи по характеру воздействия на сигнал?
2. Что называют белым гауссовским шумом?
3. Какие каналы связи называют каналами с релеевским замиранием?

4. Что характеризует величины $H(X|Y)$ и $H(Y|X)$ для канала связи с помехами?

5. Как определяются информационные потери при передаче информации по каналу связи с помехами?

6. Что называют технической скорость передачи информации?

7. От каких характеристик канала связи зависит информационная скорость?

8. Что понимают под пропускной способностью канала связи?

9. Как определяется пропускная способность канала связи при отсутствии помех?

10. Каким образом рассчитывается пропускная способность дискретного канала с помехами?

8. СТАТИСТИЧЕСКИЕ МЕТОДЫ СЖАТИЯ ИНФОРМАЦИИ

8.1. Цель и задачи темы

Цель работы – изучить особенности статистических методов сжатия информации.

Основные задачи:

• ознакомиться с методом Хаффмана для источников без памяти;

• изучить метод арифметического кодирования.

Тема рассчитана на 4 часа.

8.2. Основные теоретические сведения

Метод Хаффмана для источников без памяти.

В 1952 г. американский ученый Дэвид Хаффман предложил метод эффективного (оптимального) кодирования для источников без памяти, который был назван методом Хаффмана (позже этот метод стал называться статическим или каноническим методом Хаффмана – HUFF).

Метод Хаффмана основывается на следующих утверждениях:

1. В эффективном коде длины L_i кодовых слов не убывают, то есть $L_1 \le L_2 \le \ldots \le L_m$.

2. Существует эффективной префиксный код, в котором двум наименее вероятным символам s_{m-1} и s_m соответствуют кодовые слова, имеющие одинаковую длину и различающиеся лишь в последнем разряде.

Для двоичного кода метод Хаффмана сводится к следующему:

1. Символы алфавита выписываются в столбец в порядке убывания вероятностей.

2. Два последних символа с наименьшими вероятностями объединяются в один вспомогательный символ, которому приписывается суммарная вероятность.

3. Вероятности символов, участвующих в объединении и полученная суммарная вероятность вновь располагаются в порядке убывания вероятностей в дополнительном столбце, а два последних символа объединяются.

4. Процесс продолжается до тех пор, пока не будет получена единственный вспомогательный символ с суммарной вероятностью, равной 1.

Рассмотрим пример построения кода Хаффмана для алфавита из пяти символов (табл. 8.1). Результат показан на рис. 8.1 в виде кодового дерева соответствующего коду Хаффмана.

Таблица 8.1.

Получение эффективного кода по методу Хаффмана

s_i	1	2	3	4	5
s_3	0,4 →	0,4 →	0,4	0,6 (1) →	1,0
s_1	0,25 →	0,25	0,35 (1)	0,4 (0)	
s_5	0,2 →	0,2 (1)	0,25 (0)		
s_2	0,1 (1)	0,15 (0)			
s_4	0,05 (0)				

Для получения кодовой комбинации, соответствующей данной букве необходимо проследить путь перехода по строкам и столбцам таблицы. Этот процесс можно представить как построение кодового дерева, корень которого – вспомогательная буква с суммарной вероятностью 1 (рис 8.1), причем ветви дерева с большей вероятностью присваивается 1, а с меньшей – 0.

Код Хаффмана, как и код Шеннона-Фано, также является префиксным кодом.

Коды Хаффмана играют важную роль в кодировании изображений, звука и видео. Они являются составной частью стандартов JPEG, MPEG и H.261.

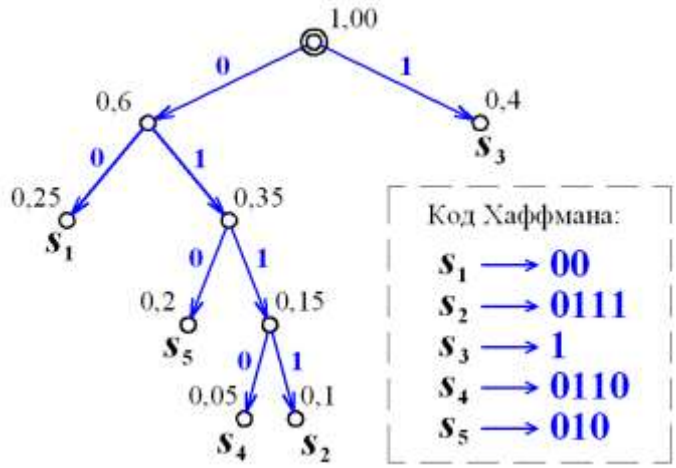


Рис. 8.1. Пример кодового дерева для кода Хаффмана

Метод арифметического кодирования.

Алгоритм кодирования Хаффмана, в лучшем случае, не может передавать на каждый символ сообщения менее одного бита информации. Потребовалось разработать такой метод кодирования, который позволял бы кодировать символы менее чем одним битом. Одним из лучших среди таких методов является арифметическое кодирование, предложенное в 70-х годах XX века.

Арифметическое кодирование представляет собой метод сжатия информации, в котором кодируемое сообщение представляется в виде двоичного дробного числа от 0 до 1. То есть в отличии от методов Шеннона-Фано и Хаффмана кодируется не каждый символ алфавита в отдельности, а кодируется сообщение целиком. Причем, чем длиннее сообщение, тем длиннее получающееся в результате кодирования число. Из этого числа можно однозначно восстановить исходную последовательность символов.

Арифметическим кодом считаются знаки, полученные после запятой.

Пусть задан дискретный источник сообщений $[S, p(s_i)]$ с алфавитом $S=\{s_1, \ldots, s_m\}$ и распределением вероятностей $p(s_i)$. На выходе источника задана последовательность символов

$$\bar{s} = (s_{i_1}, s_{i_2}, \ldots, s_{i_k}, \ldots, s_{i_M});$$

где i – порядковый номер символа в алфавите; k – порядковый номер символа в последовательности (двойная нумерация)

Для заданной последовательности $\bar{s}$ требуется получить кодовое слово:

$$\bar{x} = (x_{j_1}, x_{j_2}, ..., x_{j_L});$$

где $x_j = 0$ или 1.

Определим величины, называемые ***кумулятивными вероятностями*** $q(s_i)$ символов s_i называются величины:

$$q(s_1) = 0; \quad q(s_2) = p(s_1); \quad \ldots \quad q(s_m) = \sum_{i=1}^{m-1} p_i \, .$$

В рекурсивной форме данные вероятности можно записать следующим образом:

$$q(s_i) = q(s_{i-1}) + p(s_{i-1}).$$

Определим величины F_k и G_k, представляющие собой нижнюю границу и длину интервала вероятностей, который соответствует последовательности $\bar{s}_{i_k}$:

$$F_k = F_{k-1} + q(s_i) \cdot G_{k-1};$$

$$G_k = p(s_i) \cdot G_{k-1}.$$

При $k = 0$ (начальный шаг алгоритма) принимают F=0, G=1.

Длина кодового слова $\bar{x}$, обеспечивающая однозначное декодирование последовательности символов определяется следующим образом

$$L = \lceil -\log_2 G_M \rceil + 1.$$

где G_M – ширина интервала вероятностей, соответствующего последнему шагу алгоритма (k=M); $\lceil ... \rceil$ – округление числа, стоящего в скобках, до целого в большую сторону.

Искомое кодовое слово $\bar{x}$ определяется как L знаков после запятой в двоичной записи числа $F_M + G_M/2$, где F_M и G_M – нижняя граница интервала вероятностей и его длина, соответствующие последнему (M-му) символу последовательности:

$$\text{bin}[F_M + G_M/2] = 0, x_{j_1} x_{j_2} ... x_{j_L} \rightarrow \bar{x} = (x_{j_1}, x_{j_2}, ..., x_{j_L})$$

Таким образом, метод арифметического кодирования заключается в последовательном нахождении величин $q(s_i)$, F_k, G_k, для i = 2, 3, … , m и k = 1, 2, … , M, пока не будут определены

значения F_M и G_M. На основе последних значений и определяется искомое кодовое слово.

Для пояснения метода арифметического кодирования можно прибегнуть к его графической интерпретации. В этом случае интервал [0, 1) вероятностей появления символов рассматривается в виде отрезка длиной 1. При этом символам на данном отрезке будут соответствовать непересекающиеся полуинтервалы длинами, равными вероятностям появления символов (рис 8.2). Кумулятивные вероятности соответствуют началам этих полуинтервалов. Эти точки идентифицируют сообщения источника.

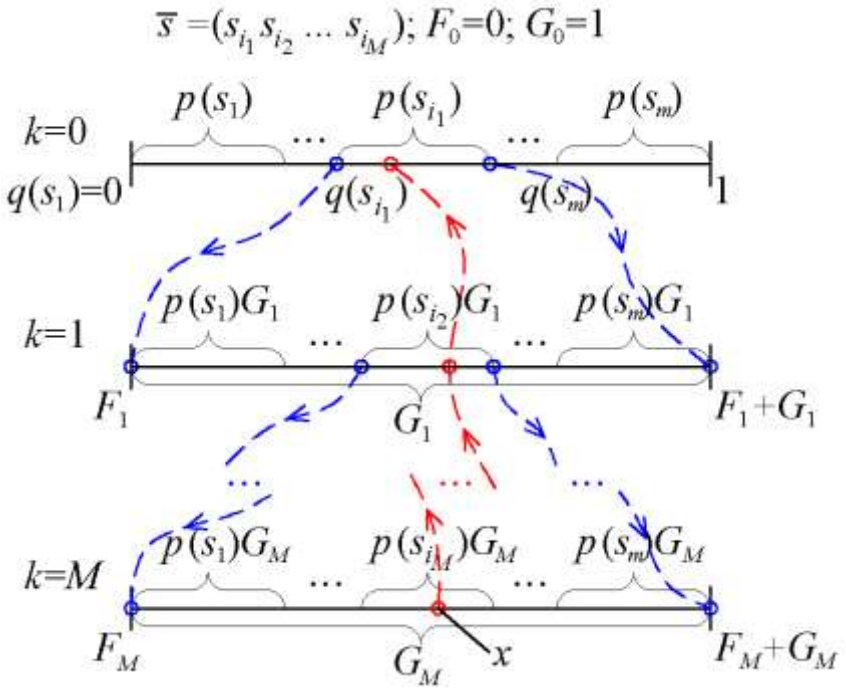


Рис. 8.2. Графическая интерпретация арифметического кодирования

На каждом шаге алгоритма производится пересчет границ всего отрезка, в котором будет расположено число, соответствующее кодовому слову. Число $F(\bar{s}_{i_k})$ в этом случае является начальной точкой отрезка на шаге k, а число $G(\bar{s}_{i_k})$ является длиной данного отрезка.

Рассмотрим пример. Пусть задан дискретный источник сообщений с алфавитом $S = \{s_1, s_2, s_3\}$ и распределением вероятностей $p(s_1)=0{,}1$, $p(s_2)=0{,}6$, $p(s_3)=0{,}3$. Произведем арифметическое

кодирование последовательности $\bar{s} = (s_2 s_3 s_2 s_1 s_2)$ длиной M=5. Процесс арифметического кодирования последовательности $\bar{s}$ представлен в таблице 8.2.

Длина кодового слова будет

$L = \lceil -\log_2 0{,}00648 \rceil + 1 = \lceil 7{,}2698 \rceil + 1 = 9$, т. е. после запятой в кодовом слове должно быть не меньше 9 знаков.

Кодовое слово

x = bin(0,53908 + 0,00648/2) = bin(0,54232) =

$= 0{,}100010101_2 \rightarrow \bar{x}$ = 100010101.

Таблица 8.2.

Арифметическое кодирование последовательности

Шаг k	s_i	$\bar{s}_{i_k}$	$p(s_i)$	$q(s_i)$	$F(\bar{s}_{i_k})$	$G(\bar{s}_{i_k})$
0	–	–	–	–	0	1
1	s_2	s_2	0,6	0,1	0,1	0,6
2	s_3	$s_2 s_3$	0,3	0,7	0,52	0,18
3	s_2	$s_2 s_3 s_2$	0,6	0,1	0,538	0,108
4	s_1	$s_2 s_3 s_2 s_1$	0,1	0	0,538	0,0108
5	s_2	$s_2 s_3 s_2 s_1 s_2$	0,6	0,1	0,53908	0,00648

Суммарная сложность арифметического кодирования пропорциональна $n(n + 1)/2$.

Таким образом, арифметическое кодирование может быть использовано тогда, когда степень сжатия важнее, чем затраты на сжатие информации.

Декодеру арифметического кода должны быть известны следующие сведения:

• алфавит $S=\{s_1, \ldots, s_m\}$ и распределение вероятностей $p(s_i)$;

• кумулятивные вероятности $q(s_i)$;

• длина декодируемой последовательности M;

• кодовое слово $\bar{x}$.

На нулевом шаге (k=0) принимаем $q(s_{M+1}) = 1$, $F_k = 0$, $G_k = 1$.

Последовательно (от i = 1 до m) для символов s_i на каждом шаге производят проверку выполнения условия:

$$F_k + q(s_i) \cdot G_k < x.$$

Если для символа s_i данное условие не выполнятся, то в качестве символа декодируемой последовательности на данном шаге принимают s_{i-1}. В обратном случае (условие выполняется для всех i) принимают символ s_m с наибольшей кумулятивной вероятностью $q(s_m)$.

Для найденного символа определяют величины F_k и G_k:

$$F_k = F_{k-1} + q(s_i)\cdot G_{k-1};$$

$$G_k = p(s_i)G_{k-1}.$$

Алгоритм продолжают до тех пор, пока не будет найден последний символ ($k = M$) декодируемой последовательности $\bar{s}$.

Рассмотрим декодирование последовательности из предыдущего примера.

Таблица 8.3.

Декодирование арифметического кода

k	F_k	G_k	Гипотеза s_i	$q(s_i)$	Проверка $F_k+q_i\cdot G_k < x$	Решение s_i	$p(s_i)$
0	$100010101_2 \rightarrow x = 0{,}541_{10}$						
1	0	1	s_1	0	$0 < x$	s_2	0,6
			s_2	0,1	**0,1** $< x$		
			s_3	0,7	$0{,}7 > x$		
2	0,1	0,6	s_1	0	$0{,}1 < x$	s_3	0,3
			s_2	0,1	$0{,}16 < x$		
			s_3	0,7	**0,52** $< x$		
3	0,52	0,18	s_1	0	$0{,}52 < x$	s_2	0,6
			s_2	0,1	**0,538** $< x$		
			s_3	0,7	$0{,}646 < x$		
4	0,538	0,108	s_1	0	**0,538** $< x$	s_1	0,1
			s_2	0,1	$0{,}5488 > x$		
5	0,538	0,0108	s_1	0	$0{,}538 < x$	s_2	0,6
			s_2	0,1	**0,53908** $< x$		
			s_3	0,7	$0{,}5450 > x$		

Для арифметического кодирования, как и для кодирования методом Хаффмана, существуют адаптивные алгоритмы.

При адаптивном арифметическом кодировании кодеру при поступлении на его вход очередного сообщения не доступна информация о сообщениях, которые появятся в будущем. Поэтому способ кодирования текущего сообщения зависит только от того, какими были предыдущие сообщения.

Вместо вероятностей появления символов s_1, …, s_m при адаптивном арифметическом кодировании могут быть использованы оценочные вероятности:

$$\tilde{p}_M(s_i) = \frac{\tau_M(s_i)+1}{M+m};$$

где $\tau_M(s_i)$ – число появлений символа s_i в последовательности длины M; m – объем алфавита.

Существуют также другие подходы к определению оценочных вероятностей при адаптивном кодировании. Например, подход, основанный на введении в алфавит дополнительной буквы, которую называют *esc*-символом.

Тем буквам алфавита, которые уже встречались в последовательности из M предшествовавших букв, приписываются вероятности

$$\tilde{p}_M(s_i) = \frac{\tau_M(s_i)}{M+1}, \qquad \tau_M(s_i) > 0.$$

Если в момент времени $M + 1$ появилась буква, которой ранее ни разу не было, то передается *esc*-символ, которому приписывается вероятность

$$\tilde{p}_M(esc) = \frac{\tau_M(s_i)}{M+1},$$

а затем передается сама буква. При этом всем буквам, которые не встречались, приписываются одинаковые вероятности.

8.3. Контрольные вопросы

1. С помощью какой операции в методе Хаффмана обеспечивается получение вспомогательных символов?
2. Как получают кодовые слова по методу Хаффмана?

3. В какую форму преобразуется последовательность символов в методе арифметического кодирования?

4. Что понимают под кумулятивными вероятностями символов?

5. Как в методе арифметического кодирования определяются значения величин F_k и$\cdot G_k$?

6. Что представляют собой величины $p(s_i)$, $q(s_i)$, F_k,$\cdot G_k$ в графической интерпретации арифметического кодирования?

7. Как находят значение числа, соответствующего арифметическому коду?

8. В чем заключается декодирование арифметического кода?

9. ПОСТРОЕНИЕ И ДЕКОДИРОВАНИЕ ЛИНЕЙНЫХ БЛОКОВЫХ КОДОВ

9.1. Цель и задачи темы

Цель темы – изучить особенности построения и декодирования линейных блоковых кодов.

Основные задачи темы:

• рассмотреть математическое описание линейных блоковых кодов;

• ознакомиться с порождающими и проверочными матрицами линейных блоковых кодов;

• изучить особенности декодирования линейных кодов с помощью стандартной таблицы.

Тема рассчитана на 4 часа.

9.2. Основные теоретические сведения

Математическое описание линейных блоковых кодов.

Самый большой класс блоковых корректирующих кодов составляют ***линейные коды***, у которых значения проверочных символов определяются при проведении линейных операций над информационными символами.

К ***линейным операциям*** относятся сложение элементов некоторого множества и умножение их на число.

То есть, если исходное информационное слово $\bar{x} = (x_1, x_2, \ldots, x_m)$, то при кодировании его линейным кодом будет получено слово $\bar{y} = (y_1, y_2, \ldots, y_n)$ длиной $n = m + k$, где m символов y_j будут являться информационными символами x_i, а k символов – дополнительными проверочными символами p_j.

Значения проверочных символов определяются через информационные на основе системы проверочных равенств:

$$p_j = \sum_{i=1}^{m} g_{ij} x_i \; ; \; (j = 1, 2, \ldots, k) \qquad (9.1)$$

где g_{ij} – коэффициенты, принимающие значения согласно правилам формирования конкретного линейного кода.

Число проверочных символов k и номера конкретных информационных символов, входящих в каждое из равенств, определяется тем, какие и сколько ошибок должен исправлять или обнаруживать данный линейный блоковый код.

Для случая двоичных линейных блоковых кодов каждый проверочный символ выбирают таким, чтобы его сумма с определенными информационными символами была равна нулю.

Математическим аппаратом для построения линейных кодов служит линейная алгебра. Поэтому понадобится рассмотреть некоторые понятия из данного раздела математики (линейное пространство, линейно-зависимые вектора, базис линейного пространства, ортогональность векторов, матрицы, операции над матрицами и др.).

Множество A является ***полем***, если для любых элементов a_i, a_j, a_k из A определены операции сложения «+» и умножения «×», а также выполняются следующие аксиомы:

Для сложения.

A1. Замкнутость: $\boldsymbol{a_i + a_j \in A}$.

A2. Коммутативность: $\boldsymbol{a_i + a_j = a_j + a_i}$.

A3. Ассоциативность: $\boldsymbol{(a_i + a_j) + a_k = a_i + (a_j + a_k)}$.

A4. Существует единственный нулевой элемент $\boldsymbol{0 \in A}$, такой, что: $\boldsymbol{0 + a_i = a_i}$.

A5. Для каждого $\boldsymbol{a_i \in A}$ существует противоположный элемент $\boldsymbol{-a_i \in A}$, такой, что $\boldsymbol{a_i + (-a_i) = 0}$.

Для умножения.

A6. Замкнутость: $\boldsymbol{a_i \times a_j \in A}$.

A7. Коммутативность: $\boldsymbol{a_i \times a_j = a_j \times a_i}$.

A8. Ассоциативность: $\boldsymbol{(a_i \times a_j) \times a_k = a_i \times (a_j \times a_k)}$.

A9. Существует единственный единичный элемент $\boldsymbol{1 \in A}$, такой, что $\boldsymbol{1 \times a_i = a_i}$.

A10. Для каждого $\boldsymbol{a_i \in A}$ существует обратный элемент $\boldsymbol{a_i^{-1} \in A}$, такой, что $\boldsymbol{a_i \times a_i^{-1} = 1}$.

Для сложения и умножения.

A11. Дистрибутивность $\boldsymbol{a_i \times (a_j + a_k) = a_i \times a_j + a_i \times a_k}$.

Примерами полей являются множества рациональных, действительных и комплексных чисел.

Так как в каналах связи множество передаваемых сигналов (символов) всегда конечно, то для теории информации интерес представляют поля с конечным числом элементов.

Поля с конечным числом элементов называют ***конечными полями*** или ***полями Галуа*** и обозначают через $GF(q)$, где q – число элементов конечного поля. При работе с кодами q – основание кода.

Поле не может содержать менее двух элементов, поскольку в нем должны быть по крайней мере нейтральный элемент относительно операции сложения (0) и нейтральный элемент относительно операции умножения (1).

Поле, включающее только 0 и 1, называется ***двоичным полем Галуа*** и обозначается $GF(2)$. Обратным элементом к 1 по сложению и умножения является 1. Правила сложения и умножения в поле $GF(2)$ (арифметика по модулю 2) следующие:

$\oplus$	**0**	**1**
0	0	1
1	1	0

$\otimes$	**0**	**1**
0	0	0
1	0	1

Линейным (векторным) ***пространством*** V над числовым полем F называется множество элементов $\bar{x}$, $\bar{y}$, $\bar{z}$, … , для которых возможны операции сложения и умножения на число из F. Элементы $\bar{x}$, $\bar{y}$, $\bar{z}$, ... $\in V$ называются ***векторами***. Каждое линейное пространство содержит нулевой вектор $\bar{0} = 0 \cdot \bar{x}$.

Важным свойством векторов линейного пространства является линейная зависимость.

Векторы $\bar{x}_1$, $\bar{x}_2$, … , $\bar{x}_m$, линейного пространства V называются ***линейно зависимыми над полем*** F, если в этом поле существуют числа α_1, α_2, … α_m, из которых хотя бы одно отлично от нуля, такие, что

$$\alpha_1 \bar{x}_1 + \alpha_2 \bar{x}_2 + \ldots + \alpha_m \bar{x}_m = \bar{0}.$$

Если данное равенство выполняется только при $\alpha_1 = \alpha_2 = \ldots = \alpha_m = 0$, то векторы $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m$ называются ***линейно независимыми***.

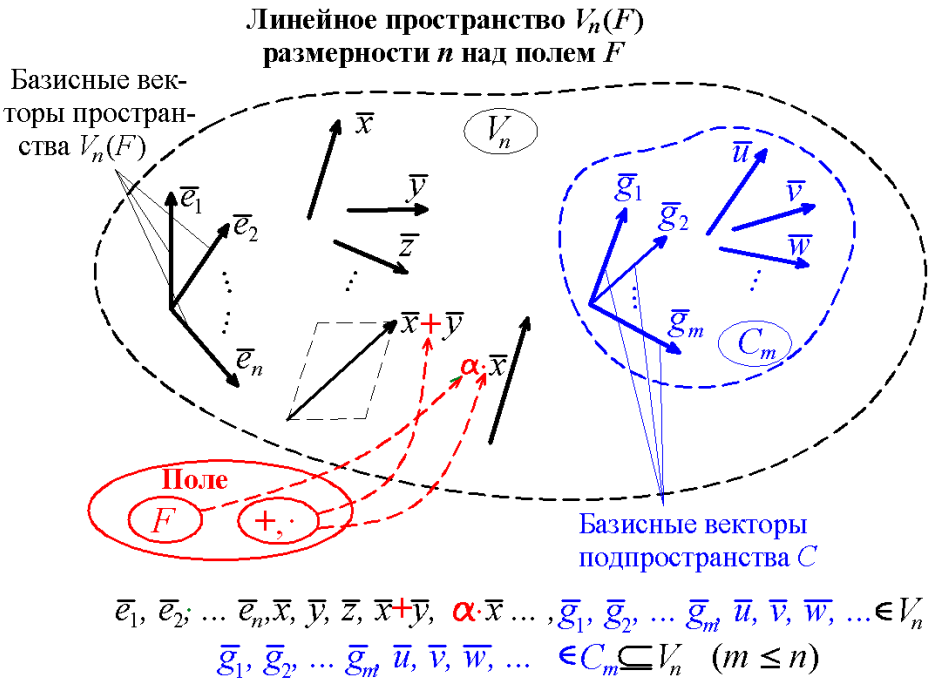


Рис. 9.1. Геометрическая интерпретация понятий линейного пространства и подпространства над полем

Наибольшее число линейно независимых векторов в линейном пространстве V характеризует размерность (число измерений) этого пространства. Линейное пространство называется n-мерным, если наибольшее число линейно независимых векторов в нем равно n (n-мерное пространство обозначается через V_n).

Система из n линейно независимых векторов n-мерного пространства V называется ***базисом*** в V.

Любой вектор $\bar{x}$ из n-мерного пространства V может быть представлен как линейная комбинация базисных векторов $\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_n$:

$$\bar{x} = x_1\bar{e}_1 + x_2\bar{e}_2 + \ldots + x_n\bar{e}_n,$$

где числа $x_1, x_2, \ldots, x_n$ называются координатами вектора $\bar{x}$ в базисе $\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_n$, а слагаемые $x_1\bar{e}_1, x_2\bar{e}_2, \ldots, x_n\bar{e}_n$, называются компонентами вектора $\bar{x}$. Через свои координаты вектор $\bar{x}$ записывается как $\bar{x} = (x_1, x_2, \ldots, x_n)$.

Множество векторов C, образующих линейное пространство относительно уже введенных в V операций сложения и умножения на число из F, называется ***подпространством*** C линейного пространства V ($C \subseteq V$).

Введем строгое определение линейного блокового кода как подпространства линейного пространства V_n над конечным полем $GF(K)$, где K – основание кода, то есть число символов в кодовом алфавите.

Пусть V_n – линейное n-мерное пространство над полем $GF(K)$, тогда подпространство $C \subseteq V_n$ называется ***линейным блоковым кодом*** с основанием K и разрядности n.

Кодовые слова линейного кода C называют ***кодовыми векторами*** и обозначают через $\bar{x}$, $\bar{y}$, $\bar{z}$, $\ldots \in C$. Кодовый вектор $\bar{0}$, состоящий из одних нулей, называется ***нулевым кодовым вектором***. Число ненулевых компонент кодового вектора $\bar{x}$ называется его ***весом*** и обозначается через $w(\bar{x})$.

Расстояние Хэмминга $d(\bar{x}, \bar{y})$ между кодовыми векторами $\bar{x}$ и $\bar{y}$ равно весу разности этих векторов:

$$\bar{z} = \bar{x} - \bar{y};\ \bar{x} \neq \bar{y};\ d(\bar{x}, \bar{y}) = w(\bar{z}). \qquad (9.2)$$

Для двоичных кодовых векторов расстояние Хэмминга можно найти как вес вектора, полученного путем суммирования исходных кодовых векторов по модулю 2.

Кодовое расстояние линейного кода C равно минимальному весу его ненулевых кодовых векторов:

$$d(C) = \min\ w(\bar{x});\ \bar{x} \in C;\ \bar{x} \neq \bar{0}. \qquad (9.3)$$

Для линейного кода C размерности n с числом информационных символов m в кодовых векторах и кодовым расстоянием d применяют обозначение (n, m, d)-код. Если не уточняют кодовое расстояние, то линейный код обозначают, как (n, m)-код.

Порождающие матрицы линейных блоковых кодов.

Линейный код можно задать путем перечисления его кодовых векторов, число которых будет равно K^m, где K – основание кода, m – число информационных символов. При больших значениях m такой способ является неудобным. Поэтому для задания

линейных кодов применяют более компактный способ с помощью порождающих матриц, которые составляются из базисных векторов линейного подпространства, соответствующего коду.

Пусть задан линейный блоковый (*n*, *m*, *d*)-код *C*, тогда матрица $G_{m,n}$, составленная из базисных векторов $\bar{g}_1$, $\bar{g}_2$, … , $\bar{g}_m$ подпространства $C \subseteq V_n$, называется ***порождающей*** (*производящей, образующей*) матрицей линейного кода *C*:

$$G_{m,n} = \begin{bmatrix} \bar{g}_1 \\ \bar{g}_2 \\ \dots \\ \bar{g}_m \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1n} \\ g_{21} & g_{22} & \dots & g_{2n} \\ \dots & \dots & \dots & \dots \\ g_{m1} & g_{m2} & \dots & g_{mn} \end{bmatrix}; \qquad (9.4)$$

где g_{ij} – компоненты базисных векторов ($i = \overline{1,m}$; $j = \overline{1,n}$) подпространства *C*.

То есть строками порождающей матрицы $G_{m,n}$ являются любые *m* линейно независимых *n*-мерных кодовых векторов $\bar{g}_i = [g_{i1}, g_{i2}, \dots, g_{ij}, \dots, g_{in}]$, отстоящие друг от друга не менее чем на заданное кодовое расстояние *d*. **(/d)**

Поскольку линейное подпространство можно задать различными базисами (набором базисных векторов), то один и тот же линейный код можно задать различными порождающими матрицами.

Зная порождающую матрицу линейного блокового кода, легко найти кодовый вектор $\bar{y}$, соответствующую любой информационной последовательности $\bar{x}$ из *m* информационных символов.

Кодовый вектор $\bar{y} \in C$ можно получить из информационного вектора $\bar{x}$ путем умножения $\bar{x}$ на порождающую матрицу *G* кода *C*:

$$\bar{x}_m \times G_{m,n} = \bar{y}_n. \qquad (9.5)$$

Пространство матрицы останется неизменным при выполнении элементарных преобразований над ее строками.

К элементарным преобразованием над строками матрицы относятся:

- перестановка любых двух строк;
- умножение любой строки на ненулевой элемент поля;

• сложение какой-либо строки с произведением другой строки на ненулевой элемент поля.

Если порождающая матрица G_2 получена из порождающей матрицы G_1 с помощью элементарных преобразований над строками, то обе матрицы порождают один и тот же код.

Перестановка двух любых столбцов порождающей матрицы приводит к порождающей матрице эквивалентного кода.

Линейные блоковые коды, отличающиеся перестановкой столбцов в порождающей матрице, называются ***эквивалентными кодами***. Корректирующая способность эквивалентных кодов одинакова.

Две порождающие матрицы, называются ***комбинаторно-эквивалентными***, если одна может быть получена из другой путем элементарных преобразований строк и перестановкой столбцов.

Гораздо более удобными для построения являются порождающие матрицы в приведенной форме, которые является комбинаторно-эквивалентными матрицами. В приведенной форме порождающая матрица является блочной матрицей, состоящей из двух матриц: единичной матрицы размерности m и дописываемой справа матрицы-дополнения размерности $m \times k$, которая состоит из проверочных символов базисных векторов.

Для любой порождающей матрицы можно получить комбинаторно-эквивалентную матрицу, называемую ***приведенной*** (систематической) ***формой***, которая имеет следующий вид:

$$G_{m,n} = \left[E_m \mid P_{m,k}\right] = \left[\begin{array}{cccc|ccccc} 1 & 0 & \ldots & 0 & p_{11} & p_{12} & \ldots & p_{1k} \\ 0 & 1 & \ldots & 0 & p_{21} & p_{22} & \ldots & p_{2k} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & p_{m1} & p_{m2} & \ldots & p_{mk} \end{array}\right]; \qquad (9.6)$$

где E_m – единичная матрица; $P_{m,k}$ – матрица-дополнение, состоящая из проверочных символов p_{ij} базисных векторов.

Линейные блоковые коды, которые соответствуют порождающей матрице в приведенной форме, называют ***систематическими кодами***. В систематических кодах первые m компонент каждого кодового вектора являются информационными символами, а последние $k = n - m$ – проверочными символами.

В теории помехоустойчивых кодов доказывается следующая теорема:

Каждый линейный блоковый (n, m)-код является эквивалентным систематическому коду.

То есть линейный блоковый код, получаемый на основе некоторой порождающей матрицы, будет обладать той же корректирующей способностью, что и код (систематический), получаемый на основе этой же матрицы в приведенной форме.

Так как матрица-дополнение содержит всю информацию о правилах построения кода, то линейный блоковый код с заданными свойствами можно получить путем построения соответствующей матрицы-дополнения. Правила построения матрицы дополнения для получения линейного блокового кода с заданными свойствами будут рассмотрены в пятой практической работе. При этом руководствуются одним из двух взаимно противоречивых требований:

• получаемый линейный блоковый код должен быть совершенным (строгое определение совершенных коды будет дано далее, а пока можно сказать, что совершенными называют коды, которые обнаруживают и исправляют максимальное число ошибок при минимальном числе проверочных символов);

• аппаратура для кодирования/декодирования линейного блокового кода должна быть максимально простой.

Проверочные матрицы линейных блоковых кодов.

Другой формой задания линейных блоковых кодов являются проверочные матрицы, которые основаны на рассмотрении ортогонального линейного подпространства.

Для векторов $\bar{x} = (x_1, x_2, \ldots, x_n)$, $\bar{y} = (y_1, y_2, \ldots, y_n)$ линейного пространства V_n может быть введена операция скалярного произведения:

$$(\bar{x}, \bar{y}) = \sum_{i=1}^{n} x_i y_i .$$

Если $(\bar{x}, \bar{y}) = 0$, то векторы $\bar{x}$ и $\bar{y}$ называются ***ортогональными***. В аналитической геометрии данное отношение между век-

торами свидетельствует, что векторы $\bar{x}$ и $\bar{y}$ взаимно перпендикулярны (рис. 9.2).

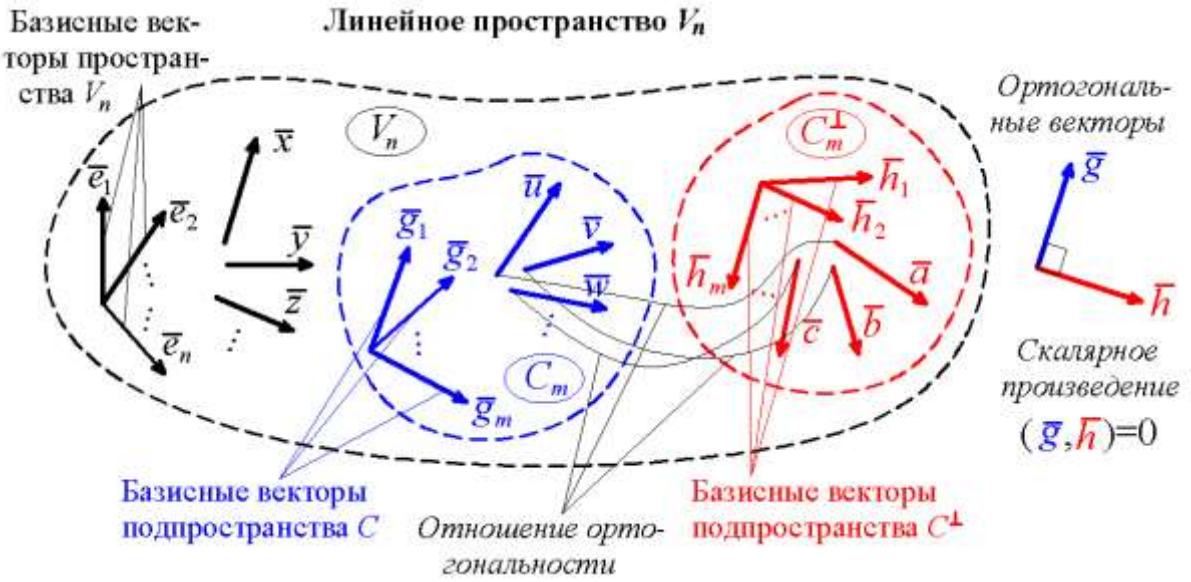


Рис. 9.2. Геометрическая интерпретация ортогональности векторов и ортогонального подпространства

Пусть $C \subseteq V$ – линейное подпространство, тогда множество всех векторов ортогональных к векторам из C называется ортогональным линейным подпространством, обозначаемым через $C^{\perp}$.

Порождающая матрица H для линейного подпространства $C^{\perp}$ называется ***проверочной матрицей*** линейного блокового кода $C \subseteq V_n$. То есть проверочная матрица состоит из базисных векторов линейного подпространства $C^{\perp}$.

$$H_{m,n} = \begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \dots \\ \bar{h}_m \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \dots & \dots & \dots & \dots \\ h_{m1} & h_{m2} & \dots & h_{mn} \end{bmatrix}; \qquad (9.7)$$

где h_{ij} – компоненты базисных векторов $(i = \overline{1,m};\ \ j = \overline{1,n})$ подпространства $C^{\perp}$. **(/d)**

Проверочная матрица обладает очень важным свойством, которое позволяет определить принадлежит ли принятый кодо-

вый вектор данному коду или нет, то есть определить, содержатся ли в нем ошибки или нет.

Транспонированной матрицей по отношению к матрице $A=[a_{ij}]$ $(i=\overline{1,m};\ \ j=\overline{1,n})$ называется матрица $A^{\mathsf{T}}=[a_{ij}{}^{\mathsf{T}}]$, элементы которой удовлетворяют условию $a_{ij}{}^{\mathsf{T}} = a_{ji}$.

При транспонировании происходит замена ролями строк и столбцов матрицы (первая строка становится первым столбцом, вторая – вторым и т. д.), что в общем случае ($m \neq n$) приводит к изменению размера матрицы:

$$\text{Если } A=\begin{matrix} \\ m \end{matrix}\overset{n}{\left[\quad\right]}, \text{ то } A^{\mathsf{T}}=\begin{matrix} \\ n \end{matrix}\overset{m}{\left[\quad\right]}.$$

Если умножить любой кодовый вектор данного кода на транспонированную проверочную матрицу этого же кода, то будет получен нулевой вектор (т. е. вектор, состоящий из одних нулей). Можно сформулировать следующую теорему.

Если H – проверочная матрица линейного блокового кода C, то кодовый вектор $\bar{y} = [y_1 y_2 \ldots y_n] \in C$, тогда и только тогда, когда

$$\bar{y} H^{\mathsf{T}} = \bar{0}. \qquad (9.8)$$

В координатной записи это произведение имеет следующий вид:

$$\sum_{j=1}^{n} y_j h_{ij} = 0; \qquad (i=\overline{1,m}) \qquad (9.9)$$

Последние уравнения называются ***обобщенными проверками на четность***.

Если задана порождающая матрица линейного блокового кода, то возникает задача построения соответствующей проверочной матрицы. Проще всего это сделать с помощью порождающих матриц в приведенной форме, поскольку существует следующая теорема:

Для линейного блокового кода с порождающей матрицей в приведенной форме $G_{m,n} = \left[E_m \middle| P_{m,k}\right]$ проверочной матрицей будет являться матрица:

$$H_{k,n} = \left[- P_{m,k}^{T} \middle| E_k \right] = \left[\begin{array}{cccc|cccc} p_{11} & p_{12} & \dots & p_{1m} & -1 & 0 & \dots & 0 \\ p_{21} & p_{22} & \dots & p_{2m} & 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ p_{k1} & p_{k2} & \dots & p_{km} & 0 & 0 & \dots & -1 \end{array} \right]. \qquad (9.10)$$

Для двоичных кодов знак «–» в проверочной матрице не ставится, поскольку в поле $GF(2)$ не существует обратных элементов.

Для проверочной матрицы, построенной на основе порождающей, выполняется условие:

$$G \times H^{\mathsf{T}} = \emptyset; \qquad (9.11)$$

где $\emptyset$ – нулевая матрица (состоит из одних нулей) размером $n \times k$.

Декодирование линейных блоковых кодов.

Для декодирования линейных блоковых кодов используются синдромы и стандартные таблицы.

Декодирование линейного блокового кода C заключается в получении переданного кодового вектора $\bar{y} \in C$ ближайшего к принятому кодовому вектору $\bar{z} = \bar{y} + \bar{e}$, где $\bar{e}$ – вектор ошибок, созданный каналом связи в процессе передачи кодового вектора $\bar{y}$.

Для нахождения вектора ошибок используют ***синдром***, который представляет собой вектор, определяемый следующим образом:

$$\bar{s} = \bar{z} \times H^{\mathsf{T}}; \qquad (9.12)$$

где H – проверочная матрица линейного блокового кода.

Покажем, что синдром является индикатором присутствия ошибок в принятом кодовому векторе:

$$\bar{s} = \bar{z} H^{\mathsf{T}} = (\bar{y} + \bar{e}) H^{\mathsf{T}} = \langle \ \bar{y} H^{\mathsf{T}} = \bar{0} \ \rangle = \bar{e} H^{\mathsf{T}}.$$

Таким образом, вычисление синдрома можно рассматривать как линейное преобразование вектора ошибок.

Рис. 9.3. Схема работа декодера линейных блоковых кодов

Ошибка будет обнаружена, если хотя бы одна из компонент $\bar{s}$ не равна нулю. Каждому принятому кодовому слову $\bar{z}$, не принадлежащему коду C, то есть $\bar{z} \notin C$, соответствует отличный от нуля синдром: $\bar{s} \neq \bar{0}$.

На основе определения синдрома записывается система проверочных уравнений:

$$\sum_{j=1}^{n} y_j h_{ij} = 0; \qquad (i=\overline{1,m})$$

В общем случае для декодирования линейных блоковых кодов используют ***стандартные таблицы*** (таблицы соответствия смежных классов), содержащие все возможные значения принятых из канала векторов $\bar{z}$ и организованные таким образом, чтобы мог быть найден ближайший к $\bar{z}$ переданный кодовый вектор $\bar{y}$ (табл. 9.1).

Таблица 9.1

Стандартная таблица для декодирования линейного блокового кода

$\bar{s}_j$	$e_j \quad y_i$	$\bar{y}_1$	$\bar{y}_2$	…	$\bar{y}_N$
$\bar{s}_1$	$\bar{e}_1$	$\bar{y}_1 + \bar{e}_1$	$\bar{y}_2 + \bar{e}_1$	…	$\bar{y}_N + \bar{e}_1$
$\bar{s}_2$	$\bar{e}_2$	$\bar{y}_1 + \bar{e}_2$	$\bar{y}_2 + \bar{e}_2$	…	$\bar{y}_N + \bar{e}_2$
…	…	…	…	…	…
$\bar{s}_K$	$\bar{e}_K$	$\bar{y}_1 + \bar{e}_K$	$\bar{y}_2 + \bar{e}_K$	…	$\bar{y}_N + \bar{e}_K$

В первой строке таблицы располагаются все кодовые векторы $\bar{y}_i$, число которых составляет $N = 2^m$. В первом столбце второй строки размещается вектор ошибки $\bar{e}_1$, вес которого равен 1.

В первой строке таблицы располагаются все кодовые векторы $\bar{y}_i$, число которых составляет $N = 2^m$. В первом столбце второй строки размещается вектор ошибки $\bar{e}_1$, вес которого равен 1.

Остальные ячейки второй строки заполняются векторами, полученными в результате суммирования по модулю 2 вектора $\bar{e}_1$ с вектором $\bar{y}_i$, расположенным в соответствующем столбце первой строки. В первом столбце третьей строки записывается вектор $\bar{e}_2$, вес которого также равен 1, однако, если вектор $\bar{e}_1$ содержит единицу в первом разряде, то $\bar{e}_2$ – во втором. В остальные ячейки третьей строки записывают суммы $\bar{x}_i$ и $\bar{e}_2$.

Аналогично поступают до тех пор, пока не будут просуммированы с векторами $\bar{y}_i$ все векторы $\bar{e}_j$, весом 1, с единицами в каждом из n разрядов. Затем суммируются по модулю 2 векторы $\bar{e}_j$ весом 2, с последовательным перекрытием всех возможных разрядов. Вес вектора $\bar{e}_j$ определяет число исправляемых ошибок. Число векторов $\bar{e}_j$ определяется возможным числом неповторяющихся синдромов и равно $K = 2^k - 1$ (нулевая комбинация говорит об отсутствии ошибки). Условие неповторяемости синдрома позволяет по его виду определять соответствующий ему вектор $\bar{e}_j$.

По виду синдрома $\bar{s}_j$ принятый кодовый вектор может быть отнесен к определенной строке таблицы соответствия (смежному классу). Принятый кодовый вектор сравнивается с векторами, записанными в данную строку и в случае совпадения в каком-либо из столбцов выбирается вектор (истинный), расположенный в первой строке данного столбца.

9.3. Контрольные вопросы

1. Какие математические операции называют линейными?
2. Что понимают под линейными блоковыми кодами?
3. Как определяют расстояние Хэмминга и кодовое расстояние для линейных блоковых кодов?
4. Как с помощью порождающей матрицей линейного блокового кода осуществляется кодирование информационных слов?

5. Какими соображениями руководствуются при построении матрицы-дополнения для порождающей матрицы линейного блокового кода?

6. Как с помощью проверочной матрицы линейного блокового кода можно определить принадлежность кодового вектора данному коду?

7. Что понимают под синдромом при декодировании линейных блоковых кодов?

8. Каким образом строится стандартная таблица декодирования линейного блокового кода?

9. Как по стандартной таблице декодирования выполняется исправление ошибок в принятых кодовых векторах?

10. ДЕКОДИРОВАНИЕ ЦИКЛИЧЕСКИХ КОДОВ

10.1. Цель и задачи темы

Цель темы – изучить основные особенности декодирования циклических кодов.

- рассмотреть проверочные матрицы и полиномы циклических кодов;
- изучить особенности работы декодера Меггита.

Тема рассчитана на 4 часа.

10.2. Основные теоретические сведения

Проверочные матрицы и полиномы циклических кодов.

Поскольку циклические коды являются линейными блоковыми кодами, то для их описания можно использовать не только порождающие матрицы, но и проверочные матрицы.

Все строки проверочной матрицы $H_{k,n}$ циклического (n,m)-кода могут быть получены циклическим вправо сдвигом одного кодового вектора $\overline{h}_m = [h_m, h_{m-1}, \ldots, h_1, h_0]$:

$$H_{k,n} = \begin{bmatrix} h_m & h_{m-1} & \ldots & h_1 & h_0 & 0 & \ldots & 0 & 0 \\ 0 & h_m & \ldots & h_2 & h_1 & h_0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & h_m & h_{m-1} & \ldots & h_1 & h_0 \end{bmatrix}.$$

Как и для линейных блоковых кодов, проверочную матрицу можно записать в приведённой форме:

$$H_{k,n} = \left[-P_{m,k}^T \middle| E_k \right] = \left[\begin{array}{cccc|cccc} p_{11} & p_{12} & \ldots & p_{1m} & -1 & 0 & \ldots & 0 \\ p_{21} & p_{22} & \ldots & p_{2m} & 0 & -1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ p_{k1} & p_{k2} & \ldots & p_{km} & 0 & 0 & \ldots & -1 \end{array} \right].$$

где E_m – единичная матрица; $P_{m,k}$ – матрица-дополнение, состоящая из проверочных символов p_{ij}.

Полином $H_m(x)$, соответствующий проверочной матрице циклического кода, называется ***проверочным полиномом*** циклического кода.

Порождающий и проверочный полиномы циклического кода связаны соотношением:

$$G_k(x)H_m(x) = x^n - 1. \qquad (10.1)$$

Если известен порождающий полином, то проверочный полином определяется следующим образом:

$$H_m(x) = (1 - x^n)/G_k(x) = h_m x^m + h_{m-1}x^{m-1} + \ldots + h_1 x + h_0. \qquad (10.2)$$

Для произвольного кодового полинома $C_{n-1}(x)$ циклического (n,m)-кода с проверочным полиномом $H_m(x)$ справедливо равенство:

$$\begin{aligned} &[C_{n-1}(x)\cdot H_m(x)] \bmod (x^n - 1) = \\ &= [F_{m-1}(x)\cdot G_k(x)\cdot H_m(x)] \bmod (x^n - 1) = \\ &= F_{m-1}(x)\cdot (x^n - 1) \bmod (x^n - 1) = 0. \end{aligned} \qquad (10.3)$$

Синдромы циклических кодов. Декодер Меггита.

Кодовый полином циклического кода, должен делиться без остатка на порождающий полином этого кода или при умножении на проверочный полином данного кода давать ноль. Это свойство позволяет обнаружить ошибку. По виду остатка можно определить полином ошибки.

На выходе канала связи полином с ошибкой $D_{n-1}(x)$ можно записать следующим образом:

$$D_{n-1}(x) = C_{n-1}(x) + E_{n-1}(x); \qquad (10.4)$$

$C_{n-1}(x)$ – кодовый полином (без ошибок); $E_{n-1}(x)$ – полином ошибок, соответствующий вектору ошибок $\bar{e}$ канала связи и содержащий единицы только в тех разрядах, которые искажены.

В циклическом коде опознавателями ошибок (синдромом) являются остатки от деления полиномов ошибок $E_{n-1}(x)$ на порождающий полином кода $G_k(x)$.

Синдромом (синдромным полиномом) ***циклического кода*** называется полином $S_{k-1}(x)$, являющийся остатком от деления

принятого полинома $D_{n-1}(x)$ на порождающий полином $G_k(x)$ данного кода:

$$S_{k-1}(x) = D_{n-1}(x) \bmod G_k(x) = < [C_{n-1}(x) + E_{n-1}(x)] \bmod G_k(x) > =$$

$$= E_{n-1}(x) \bmod G_k(x).$$

Таким образом, синдромный полином $S(x)$ может использоваться для определения полинома ошибок $E(x)$. Декодирование циклического кода с исправлением ошибок сводится к поиску полинома ошибок $E_{n-1}(x)$ по известному синдрому $S_{k-1}(x)$.

Декодер Меггита используется для декодирования систематических циклических кодов. Сложность декодера Меггита с ростом числа исправляемых ошибок растёт экспоненциально. Поэтому декодер Меггита используется для коррекции небольшого числа ошибок (от 1 до 3).

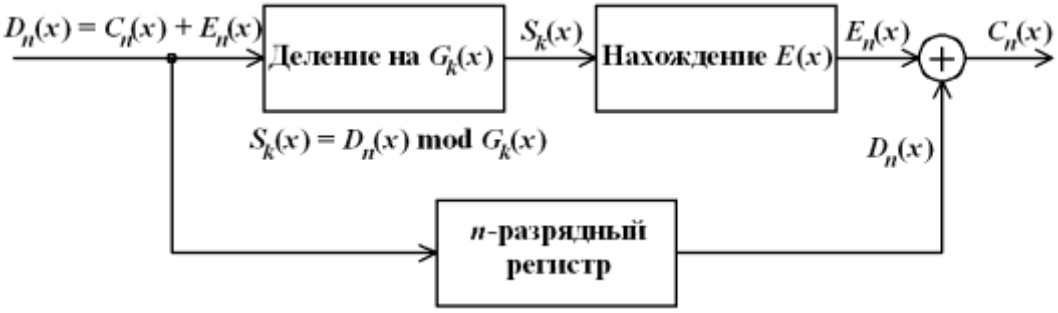


Рис. 10.1. Общая структура декодера Меггита

Идея исправления ошибок основывается на том, что ошибочный кодовый полином после определенного числа циклических сдвигов подгоняется под остаток таким образом, что в сумме с остатком он дает исправленный полином. Остаток при этом представляет собой разницу между искажёнными и правильными символами.

Подгоняют ошибочный полином до тех пор, пока число единиц в остатке w не будет меньше или равно максимальному числу ошибок t, исправляемых данным кодом.

Процесс исправления одиночной ошибки с помощью двоичного циклического кода включает следующие шаги:

1. Делят принятый кодовый полином $D_{n-1}(x)$ на порождающий полином $G_k(x)$:

$$D_{n-1}(x) / G_k(x) = Q_{m-1}(x) + S_{k-1}(x) / G_k(x).$$

где $Q_{m-1}(x)$ – частное; $S_{k-1}(x)$ – остаток.

Если $S_{k-1}(x) = 0$, то принятый полином не содержит ошибки. Если $S_{k-1}(x) \neq 0$, то в полиноме присутствует ошибка и требуется перейти к пункту 2.

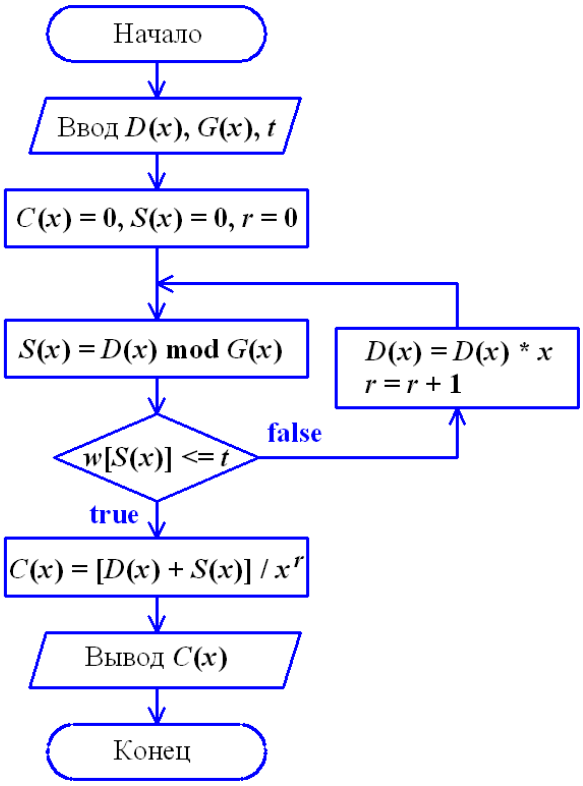


Рис. 10.2. Алгоритм работы декодера Меггита для двоичных циклических кодов

2. Определяют вес w полученного остатка $S_{k-1}(x)$ путем подсчета количества единиц в остатке. Если $w \leq t$, где t – максимальное число исправляемых кодом ошибок (корректирующая способность), то принятый полином $D_{n-1}(x)$ складывается по модулю 2 с полученным остатком $S_{k-1}(x)$. Сумма даст исправленный полином. Если $w > t$, то переходят к пункту 3.

3. Производят циклический сдвиг принятого полинома $F(x)$ на один разряд влево. Комбинацию, полученную в результате циклического сдвига, делят на порождающий полином $G(x)$.

4. Повторяют пункты 2 и 3 до тех пор, пока не будет выполнено условие $w \leq t$. Запоминают число сдвигов r полинома.

5. Полином, полученный в результате последнего циклического сдвига, складывают с остатком от деления этого полнома на порождающий полином $G_k(x)$.

6. Производят циклический сдвиг полинома, полученного в результате суммирования последнего делимого с последним остатком вправо на r разрядов. В результате будет получена исправленная кодовая комбинация $C_{m-1}(x)$.

Пусть в передаваемом кодовом полиноме произошла следующая одиночная ошибка: 1**0**00101. Выполним декодирование полинома с исправлением ошибки.

Разделим принятый полином на $G_3(x)$ в полиномиальной и векторной формах:

$$\begin{array}{l} x^6 + x^2 + 1 \\ \underline{x^6 + x^5 + x^3} \\ x^5 + x^3 + x^2 + 1 \\ \underline{x^5 + x^4 + x^2} \\ x^4 + x^3 + 1 \\ \underline{x^4 + x^3 + x} \\ x + 1 \end{array} \left| \begin{array}{l} \underline{x^3 + x^2 + 1} \\ x^3 + x^2 + x \end{array} \right. \qquad \begin{array}{l} 1000101 \\ \underline{1101} \\ 1011 \\ \underline{1101} \\ 1100 \\ \underline{1101} \\ 11 \end{array} \left| \begin{array}{l} \underline{1101} \\ 1110 \end{array} \right.$$

Остаток $S(x) = x + 1 \Leftrightarrow 011$; вес остатка $w = 2$.

Поскольку $w > t$, то сдвинем влево на один разряд полученный полином, умножив его на x по модулю $x^7 - 1$:

$$[x(x^6 + x^2 + 1)] \bmod (x^7 - 1) = x^3 + x + 1 \Leftrightarrow 0001011.$$

Разделим полином $x^3 + x + 1$ на $G(x)$.

$$\begin{array}{l} x^3 + x + 1 \\ \underline{x^3 + x^2 + 1} \\ x^2 + x \end{array} \left| \begin{array}{l} \underline{x^3 + x^2 + 1} \\ 1 \end{array} \right. \qquad \begin{array}{l} 0001011 \\ \underline{1101} \\ 110 \end{array} \left| \begin{array}{l} \underline{1101} \\ 0001 \end{array} \right.$$

Остаток $S(x) = x^2 + x \Leftrightarrow 011$; вес остатка $w = 2$.

Поскольку $w > t$, то снова сдвигаем полином влево на один разряд:

$$x(x^3 + x + 1) \bmod (x^7 - 1) = x^4 + x^2 + x \Leftrightarrow 0010110.$$

Разделим полином $x^4 + x^2 + x$ на $G(x)$.

$$\begin{array}{l|l} x^4 + x^2 + x & \underline{x^3 + x^2 + 1} \\ \underline{x^4 + x^3 + x} & x + 1 \end{array} \qquad \begin{array}{l|l} 0010110 & \underline{1101} \\ \underline{1101\ \ } & 0011 \end{array}$$

$$\begin{array}{l} x^3 + x^2 \\ \underline{x^3 + x^2 + 1} \\ 1 \end{array} \qquad \begin{array}{r} 1100 \\ \underline{1101} \\ 1 \end{array}$$

Остаток $S(x) = 1 \Leftrightarrow 001$; вес остатка $w = 1$.

Поскольку $w = t$, то сложим полученный полином по модулю 2 с остатком:

$$\begin{array}{r} 0010110 \\ \underline{\quad\ 001} \\ 0010111 \end{array} \qquad \Leftrightarrow \quad x^4 + x^2 + x + 1.$$

Так как число сдвигов влево $r = 2$, то выполним сдвиг суммы на два разряда вправо, разделив её на x^2 по модулю $x^7 - 1$:

$[(x^4 + x^2 + x + 1) / x^2] \bmod (x^7 - 1) = x^6 + x^5 + x^2 + 1 \Leftrightarrow 1100101.$

Результат совпадает с кодовым полиномом **1100101**. **(/e)**

Характерной особенностью циклических кодов является способность к распознаванию ***пакета ошибок***, под которым понимают группирование ошибок в одной ограниченной области кодового слова.

10.3. Контрольные вопросы

1. Что представляет собой проверочная матрицы циклического кода?
2. Что понимают под проверочным полиномом циклического кода?
3. Как можно найти проверочный полином при известном порождающем полиноме?
4. Как определяется синдром циклического кода?
5. При каком условии считают, что полученный кодовый полином не содержит ошибок?

6. В чём заключаются основные особенности декодера Меггита?

7. Из каких шагов состоит процесс исправления одиночной ошибки в декодере Меггита?

11. МОДЕЛИ И КЛАССИФИКАЦИИ ШИФРОВ. ТЕОРЕТИЧЕСКАЯ СТОЙКОСТЬ ШИФРОВ

11.1. Цель и задачи темы

Цель темы – изучить модели и классификации шифров, а также ознакомиться с понятием теоретической стойкости шифров.

Основные задачи темы:

• рассмотреть формальные модели шифров и модели открытых текстов;

• изучить классификацию симметричных шифров и современных алгоритмов шифрования;

• ознакомиться с понятиями энтропии открытого текста и ключа, а также с понятием расстояния единственности;

• рассмотреть правило Керкгоффа и особенности абсолютно стойкого шифра.

Тема рассчитана на 4 часа.

11.2. Основные теоретические сведения

Формальные модели шифров. Модели открытых текстов.

Введём строгое формальное определение шифра. Для этого определим составные части шифра.

Пусть X, Y, K – непустые конечные множества, называемые соответственно:

• $X = \{x\}$ – пространство открытых текстов;

• $Y = \{y\}$ – пространство шифротекстов;

• $K = \{k\}$ – пространство ключей.

Для указанных множеств можно ввести распределение вероятностей, то есть каждому элементу множества приписать вероятность его появления.

Каждый элемент пространства ключей $k \in K$ является парой $k = (k_з, k_р)$, где $k_з$ – ключ зашифрования, $k_р$ – ключ расшифрования (в общем случае $k_з \neq k_р$).

Под ***функцией*** (правилом) ***шифрования*** (зашифрования) E на ключе $k \in K$ понимают отображение следующего вида:

$$E_k: X \to Y. \tag{11.1}$$

Обозначим через $E_k(X) \subseteq Y$ множество $\{E_k(x): x \in X\}$, тогда ***функцией расшифрования*** D на ключе $k \in K$ называется отображение:

$$D_k: E_k(X) \to X. \tag{11.2}$$

То есть при расшифровании участвуют только те элементы множества Y, которые были получены при шифровании.

При шифровании используется ключ $k_з$, а при расшифровании – ключ $k_р$.

Шифр (криптографическая система) – совокупность множеств возможных открытых текстов (то, что шифруется), возможных ключей (то, с помощью чего шифруют), возможных шифрованных текстов, а также функций зашифрования и расшифрования.

Математически шифр записывается следующим образом (***алгебраическая модель шифра***):

$$\Sigma_A = (X, K, Y, E, D). \tag{11.3}$$

Шифр обладает следующими свойствами:

• однозначность расшифрования, то есть для любых $x \in X$ и $k \in K$ выполняется равенство:

$$D_k(E_k(x)) = x. \tag{11.4}$$

• любой элемент шифротекста $y \in Y$ может быть представлен в виде $E_k(x)$ для подходящих элементов $x \in X$ и $k \in K$, то есть:

$$Y = \bigcup_{k \in K} E_k(x). \tag{11.5}$$

Помимо алгебраической модели используется ***вероятностная модель шифра*** (*введена К. Шенноном*):

$$\Sigma_P = (X, K, Y, E, D, P(X), P(K)). \tag{11.6}$$

где $P(X)$, $P(K)$ – распределения вероятностей на множествах X и K (открытых текстов и ключей).

Вероятностные модели шифров используются в криптоанализе для взлома шифров.

Если $P(K)$ определяется свойствами генератора ключей, то $P(X)$ определяется частотными характеристиками самих открытых текстов. Характер таких текстов может быть различным, но открытые тексты обладают многими закономерностями, которые наследуются шифрованными текстами. Данный факт является определяющим фактором, влияющим на надёжность шифрования.

Основой для построения моделей открытого текста является теоретико-информационный подход, в котором для описания открытого текста используются модели источников сообщений.

Классификация симметричных шифров. Композиция шифров.

Первыми известными шифрами являются симметричные шифры, в которых ключи шифрования и расшифрования совпадают.

Симметричные шифры (шифры с закрытым ключом) используют один и тот же ключ, как для шифрования, так и для расшифрования информации. На практике ключи шифрования и расшифрования могут различаться, но при наличии одного ключа можно легко вычислить другой.

Классическими симметричными шифрами известными с древних времён являются шифры замены и шифры перестановки. В шифрах перестановки каждый символ остается самим собой, но меняет свое местоположение. В шифрах замены каждый символ меняется на другой символ, но остается на своем месте. Также можно выделить сочетания подобных шифров – композиционные шифры.

По типу преобразования осуществляемого с открытым текстом при шифровании выделяют следующие группы шифров:

• ***Шифры замены*** (***подстановки*** – *substitution*) – шифры, в которых элементы открытого текста заменяются некоторыми их эквивалентами в шифротексте.

Элементами открытого текста могут быть отдельные символы (самый распространённый случай), пары символов (биграммы), тройки символов (триграммы), k-граммы в общем случае и их сочетания.

• ***Шифры перестановки*** (*transposition*) – шифры, в которых символы открытого текста меняют порядок следования в шифротексте. То есть в шифрах перестановки буквы открытого текста не замещаются на другие, а меняются местами друг с другом.

• ***Композиционные (каскадные) шифры*** (*composition*) – шифры, в которых для преобразования открытого текста используется последовательное применение двух и более шифров замены и/или перестановки.

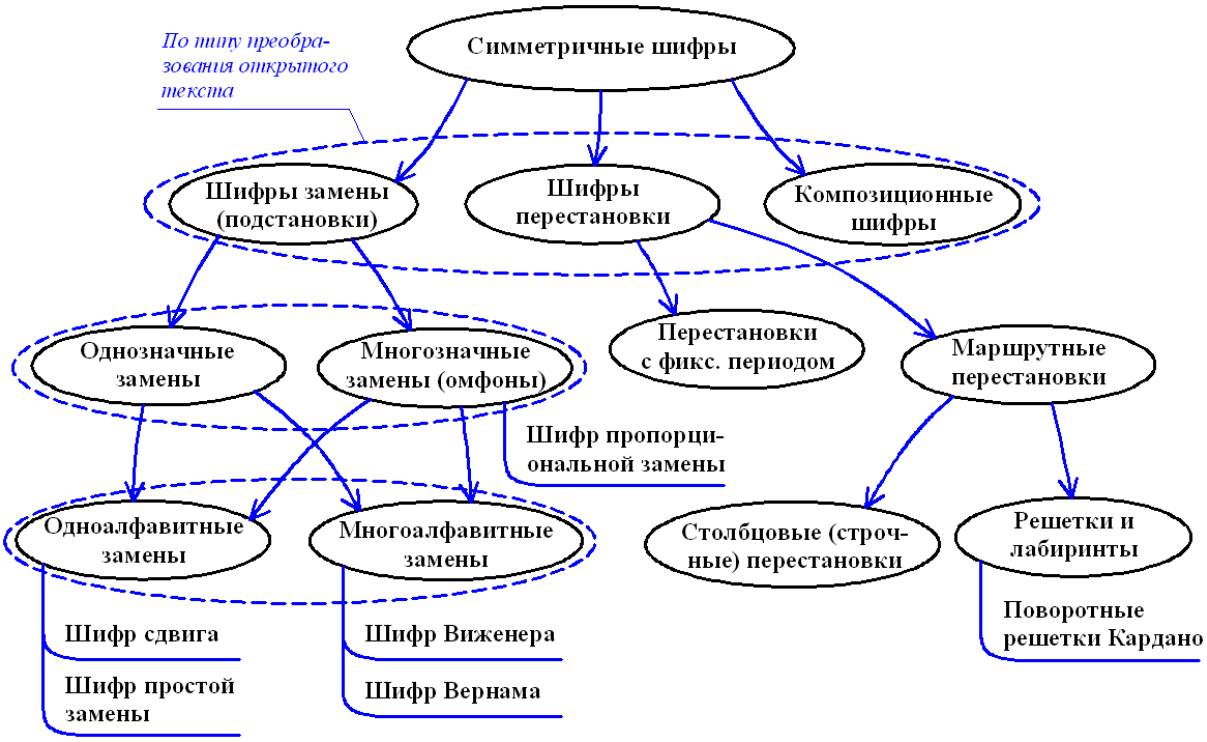


Рис. 11.1. Классификация симметричных шифров

По числу символов различают следующие виды шифров:

• ***Однозначные замены***, в которых элемент открытого текста заменяется только на один возможный элемент шифротекста.

• ***Многозначные замены*** (омфоны), в которых символ открытого текста может быть заменен одним из нескольких возможных символов.

Примером омфонического шифра является шифр пропорциональной замены, в котором каждой букве ставится в соответствие несколько эквивалентов, число которых пропорционально частоте встречаемости буквы в открытом тексте. Испоьзовние

омфонов позволяет скрыть статистику букв открытого сообщения.

По числу символов шифротекста, соответствующих символу открытого текста различают следующие виды шифров:

• ***Одноалфавитные*** (моноалфавитные) ***замены*** – шифры, в которых символы открытого текста меняются на символы того же алфавита по определенному правилу, зависящему от ключа.

Различают следующие разновидности одноалфавитных шифров замены:

- ***шифр сдвига***, в котором каждый символ заменяется на другой символ, отстоящий от исходного на определенное число k позиций в алфавите (например, шифр Цезаря, в котором $k = 3$);

- ***шифр простой замены***, в котором каждый символ открытого текста заменяется на некоторый, фиксированный при данном ключе символ того же алфавита.

• ***Многоалфавитные*** (полиалфавитные) ***замены*** – это шифры, в которых символы открытого текста меняются на символы того же или другого алфавита по определенному правилу, зависящему от ключа и от положения текста в символе. (примеры: шифр Виженера, шифр Бофора, одноразовый блокнот).

Первым известным шифром многоалфавитной замены является диск Альберти, описанный итальянским архитектором Леоном Альберти в 1466 г. В этом шифре и использовался шифровальный диск, представляющий собой пару шифровальных дисков разного диаметра (рис. 4.X). Больший из них – неподвижный, его окружность разделена на 24 равных сектора, в котором в алфавитном порядке вписано 20 латинских букв и 4 цифры (от 1 до 4). Из 24-буквенного алфавита удалены 4 буквы, без которых можно обойтись (например, в русском языке Ъ, Ё, Й). Меньший диск – подвижный, по его окружности, разбитой на 24 сектора записывались все буквы перемешанного латинского алфавита.

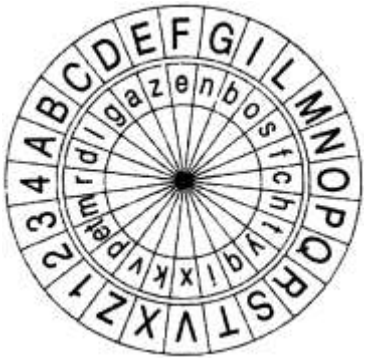


Рис. 11.2. Шифровальный диск Альберти

Примером шифра многоалфавитной замены также может служить шифр Вижинера. Этот шифр, был описан французским дипломатом Блезом Виженером в «Трактате о шифрах», вышедшем в 1585 году.

В этом методе для шифрования используется таблица, представляющая собой квадратную матрицу с числом элементов mxm, где m – количество символов в алфавите. В первой строке матрицы записывают буквы в порядке очередности их в исходном алфавите, во второй – ту же последовательность букв, но с циклическим сдвигом влево на одну позицию, в третьей – со сдвигом на две позиции и т. д.

Частным случаем многоалфавитных шифров являются шифры, использующие ***метод гаммирования*** – наложение по определенному правилу гаммы шифра на открытый текст.

Под ***гаммой шифр***а понимается псевдослучайная последовательность, вырабатываемая по заданному алгоритму, для шифрования и расшифровывания.

Процесс расшифрования сводится к повторной генерации гаммы шифра при известном ключе и наложению такой же гаммы на зашифрованные данные.

Маршрутные перестановки основаны на некоторой геометрической фигуре. Фрагмент открытого текста записывается в такую фигуру по некоторой траектории. Шифрованным текстом является последовательность, полученная при выписывании текста по другой траектории.

Разновидностью маршрутной перестановки является столбцовая (вертикальная) перестановка, в которой открытый текст

записывается обычным образом (по строкам слева направо). Сообщение выписывается по столбцам (сверху вниз), при этом столбцы выбираются в порядке, определяемом числовым ключом.

Более сложные маршрутные перестановки (решётки и лабиринты) могут использовать другие геометрические фигуры и более «хитрые» маршруты, как, например, при обходе шахматной доски ходом коня, пути в некотором лабиринте и т. п.

Примером являются решётки Джероламо Кардано. В качестве средства для шифрования использовался лист из твёрдого материала, в котором через неправильные интервалы сделаны прямоугольные выреза высотой в одну строку и различной длины. Накладывая эту решётку на лист бумаги, можно было записать в вырезы секретное сообщение. После этого, сняв решётку, нужно было заполнить оставшиеся свободные места на листе бумаги текстом, маскирующим секретное сообщение.

Последовательное применение двух или более перестановок называется составной перестановкой.

Шифры замены и перестановки представляют лишь академический интерес, поскольку не обеспечивают требуемой стойкости для современных методов и средств криптоанализа. Изучение элементарных методов шифрования позволяет проследить эволюцию шифров и лучше понять работу современных алгоритмов шифрования.

С целью повышения надёжности шифрования шифрованный текст, полученный применением некоторого шифра, может быть ещё раз зашифрован с помощью другого шифра. Такие сочетания шифров образуют класс шифров, называемых ***композиционными*** (каскадными) ***шифрами***.

Классификация современных алгоритмов шифрования.

В зависимости от числа ключей, применяемых в конкретном методе, выделяют следующие группы методов:

• ***бесключевые*** – не используют в работе ни каких ключей;

• ***одноключевые*** – работают с одним ключевым параметром (секретный ключ);

• ***двухключевые*** – использую два ключевых параметра (секретный и открытый ключи).

Классификация методов шифрования (криптографических алгоритмов) показана на рис. 11.3.

Хэширование представляет собой преобразование информации, при котором из данных неограниченного размера вычисляется сообщение фиксированной длины (хэш-значение), однозначно соответствующее исходным данным. Хэширование может выполняться как с использованием некоторого секретного ключа, так и без него.

Кроме того, хэширование применяется в методе электронной цифровой подписи, а также в системах аутентификации пользователей.

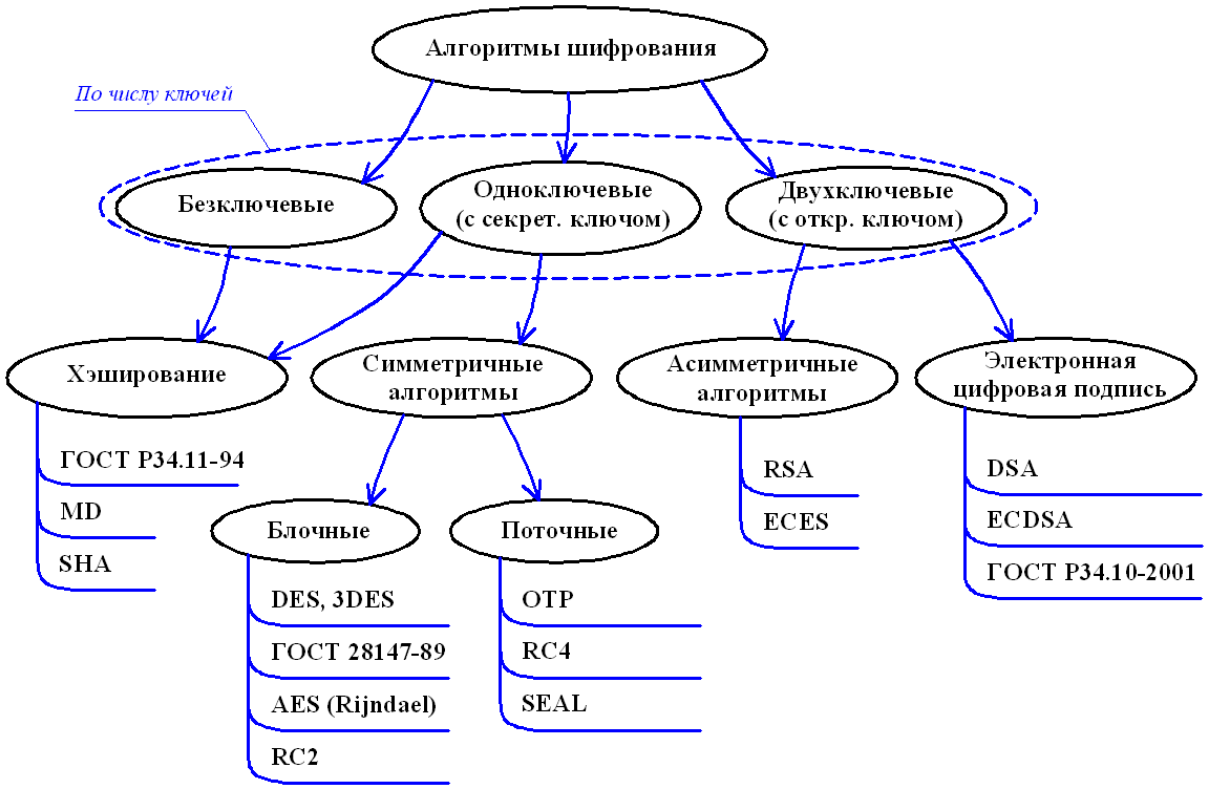


Рис. 11.3. Классификация современных алгоритмов шифрования

Хэш-функции – это математические или иные функции, принимающие на входе строку переменной длины (прообраз) и преобразующие её в выходную строку фиксированной длины

(обычно меньше прообраза), называемую значением хэш-функции.

Симметричные алгоритмы (*шифрование с закрытым ключом*) использует один и тот же ключ как для зашифровывания, так и для расшифровывания информации. На практике ключи зашифровывания и расшифровывания могут различаться, но при наличии одного ключа можно легко вычислить другой.

Симметричные алгоритмы шифрования подразделяется на два вида:

• ***Блочные алгоритмы шифрования,*** в которых информация разбивается на блоки фиксированной длины (типовым размером блока является 64 или 128 бит).

• ***Поточные алгоритмы шифрования,*** в которых шифрование производится без разделения на блоки. Алгоритмы поточного шифрования шифруют данные побитно или посимвольно.

В некоторых классификациях поточное шифрование рассматривается как частный случай блочного – шифрование блоков единичной длины.

Асимметричные алгоритмы шифрования (алгоритмы шифрования с открытым ключом) характеризуются применением двух типов ключей: открытого – для зашифровывания информации, а также секретного – для её расшифровывания.

Секретный и открытый ключи связаны соотношением, которое обеспечивает легкость вычисления открытого ключа из секретного и невозможность (за ограниченное время при реальных ресурсах) вычисления секретного ключа из открытого.

Электронная цифровая подпись (ЭЦП) представляет собой двухключевой метод шифрования, в котором в отличие от ассиметричного шифрования открытый ключ используется для проверки ЭЦП. ЭЦП применяется для подтверждения целостности и авторства данных. При соблюдении правил безопасного хранения секретного ключа никто, кроме его владельца, не в состоянии вычислить верную ЭЦП какого-либо электронного документа

Для разрабатываемых в настоящее время криптографических систем сформулированы следующие общепринятые требования:

• зашифрованное сообщение должно поддаваться чтению только при наличии ключа;

• знание алгоритма шифрования не должно снижать криптостойкости шифра;

• любой ключ из множества возможных должен обеспечивать надежную защиту информации;

• алгоритм шифрования должен допускать как программную, так и аппаратную реализацию.

Не для всех алгоритмов шифрования перечисленные требования выполняются полностью. В частности, требование отсутствия слабых ключей (ключей, которые позволяют злоумышленнику легче вскрыть зашифрованное сообщение) не выполняется для некоторых «старых» блочных шифров. Однако все вновь разрабатываемые системы шифрования удовлетворяют перечисленным требованиям.

Энтропия открытого текста и ключа. Расстояние единственности.

При шифровании осуществляется два статистических выбора – выбор открытого текста и выбор ключа. Неопределенность выбора характеризуется энтропией.

Пусть $X = \{x_i\}$ – множество открытых текстов x_i ($i = 1,\ldots, n$), а $p(x_i)$ – вероятность появления i-го открытого текста. Тогда ***априорной энтропией открытого текста*** называется величина:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i). \qquad (11.7)$$

Под ***энтропией ключа*** (энтропией криптографической системы) понимают величину

$$H(K) = -\sum_{i=1}^{m} p(k_i) \log_2 p(k_i). \qquad (11.7)$$

В случае равновероятного распределения элементов в пространстве ключей $H(K) = \log_2 m$.

Для энтропии пространств X, Y, K можно выделить следующие свойства:

1. При известном шифротексте и ключе отрытый текст полностью известен:

$$H(X|K,Y) = 0.$$

В противном случае расшифровывание будет некорректным.

2. При известном открытом тексте и ключе, можно получить шифротекст:

$$H(Y|X,K) = 0.$$

Данное свойство не будет выполняться в системах с открытым ключом.

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

$$H(X,K,Y) = H(K,Y) + H(X|K,Y) = H(K,Y)$$

Таким образом, получим формулу

$$H(K,Y) = H(K) + H(X). \qquad (11.8)$$

Условная энтропия $H(K|Y)$ характеризует то количество неопределенности относительно ключа, которое остается после прочтения шифротекста. На основе последнего выражения получим:

$$H(K|Y) = H(K,Y) - H(Y) = H(K) + H(X) - H(Y). \qquad (11.9)$$

Следовательно, неопределенность в знании ключа при известном шифротексте равна сумме неопределенностей относительно открытого текста и ключа за вычетом неопределенности в отношении шифротекста.

Избыточность естественного языка приводит к тому, что для взлома простых шифров требуется шифротекст небольшой длины.

Отдельный шифротекст выдает ненулевое количество информации об истинном ключе шифра, поскольку он исключает некоторое подмножество неподходящих ключей.

Ложным ключом шифра называется возможный, но отличный от истинного ключ.

Например, шифротекст WNAJW, полученный с помощью шифра сдвига для английского языка, порождает два открытых текста RIVER и ARENA, соответствующих ключам $k = 5$ и $k = 22$. При этом один из ключей является истинным, а другой –

ложным. Аналогичная ситуация может иметь место для любого другого шифра.

Теорема.

Пусть k_L – число ложных ключей при длине шифротекста L, тогда для любого рассматриваемого шифра с равновероятными ключами при достаточно большой длине L шифротекста имеет место неравенство:

$$k_L \geq \frac{|K|}{m^{L \cdot R}} - 1. \qquad (11.10)$$

где $|K|$ – число всех ключей; m – объём алфавита открытого текста; R – избыточность открытого текста.

Из полученного неравенства следует, что

$$\frac{|K|}{k_L + 1} \leq m^{L \cdot R}.$$

Отсюда при $k_L = 0$ получим $|K| \leq m^{L \cdot R}$ и, следовательно,

$$L \geq \frac{\log_2 |K|}{R \cdot \log_2 m}. \qquad (11.11)$$

Минимально возможное значение L в этом неравенстве обозначается L_0 и называется расстоянием единственности.

Расстоянием единственности шифра называется длина L_0 шифротекста, начиная с которой число ложных ключей становится равным нулю. Другими словами, это средний объем шифротекста, необходимый для точного вскрытия ключа атакующим, имеющим неограниченные вычислительные ресурсы.

Из неравенства (11.11) получим:

$$L_0 = \left\lceil \frac{\log_2 |K|}{R \cdot \log_2 m} \right\rceil. \qquad (11.12)$$

Например, для шифра простой замены, применяемого к открытому тексту на английском языке $L_0 = 38$. Это значит, что для английского языка по шифротексту длиной около 40 символов можно однозначно определить открытый текст.

Правило Керкгоффа. Абсолютно стойкий шифр.

Стойкостью шифра называется характеристика шифра, определяющая его способность противостоять криптографическим атакам.

При оценке стойкости шифра в криптографии используют правило Керкгоффа. В книге «Военная криптография», изданной в 1883 в., он сформулировал следующее важное правило:

Правило Керкгоффа: стойкость шифра должна зависеть только от секретности ключа; знание функций шифрования и расшифрования не должно снижать стойкости шифра.

В этом правиле учитывается самый плохой сценарий развития событий, в котором противник обладает полной информацией о шифре и имеет в своём распоряжении некоторое количество пар шифрованных и открытых текстов, связанных с ключом k. То есть оценка стойкости шифра должна проводиться при условии, что о данном шифре известно всё, кроме используемого ключа.

Рано или поздно те или иные сведения об используемом шифре становятся известными. В военное время могут быть захвачены узлы связи с шифротехникой. Могут потерпеть аварию и попасть в руки противника самолёт или судно, оборудованные шифросредствами. Нельзя исключать предательства шифровальщика и т. п.

Тем не менее шифры, используемые специальными службами, хорошо охраняются. Это обусловлено необходимостью дополнительного запаса прочности, поскольку создание шифров с доказуемой стойкостью является очень сложной проблемой.

Очень важным результатом для развития криптографии был результат о существовании и единственности абсолютно стойкого шифра. Единственным таким шифром является лента однократного использования (одноразовый блокнот), в которой открытый текст объединяется с полностью случайным ключом такой же длины. Этот результат был доказан Шенноном с помощью разработанного им теоретико-информационного метода исследования шифров.

Шифр называется ***абсолютно стойким*** (совершенно секретным в терминологии Шеннона), если анализ зашифрованного текста не может дать никакой информации об открытом тексте,

кроме, возможно, его длины. То есть абсолютная стойкость означает, что шифротекст не несет в себе сведений об открытом тексте.

Шенноном было доказано ***необходимое условие абсолютной стойкости шифра***: для того, чтобы шифр был абсолютно стойким, необходимо, чтобы энтропия ключа была не меньше энтропии открытого текста:

$$H(K) \geq H(X). \qquad (11.13)$$

Из этого условия и правила Керкхоффа следует, что для того, чтобы шифр был абсолютно стойким, необходимо, чтобы размер используемого ключа был не меньше размера открытых текстов.

Если шифр не является абсолютно стойким, то знание шифротекста предоставляет некоторую информацию относительно соответствующего ему открытого текста.

Тогда количество информации об открытом тексте, которую можно извлечь из шифротекста, определяется величиной:

$$I(X,Y) = H(X) - H(X \mid Y).$$

Шифры, для которых $I(X,Y) = 0$ или $H(X) = H(X|Y)$ можно считать абсолютно стойкими.

Требования к абсолютно стойкому шифру:

- полная случайность (равновероятность) ключа; в частности это означает, что ключ нельзя выработать с помощью какого-либо детерминированного устройства;
- равенство длины ключа и длины открытого текста;
- однократность использования ключа.

В случае нарушения хотя бы одного из этих условий шифр перестает быть абсолютно стойким и появляются принципиальные возможности его вскрытия.

Данные условия делают абсолютно стойкий шифр очень дорогим и непрактичным. Каждый текст должен иметь собственный, единственный и неповторимый ключ. Возникают большие проблемы создания, регистрации, распространения и отмены ключей.

Абсолютно стойкие шифры имеют очень ограниченную область применения, например, в сетях для передачи особо важной

государственной информации, где объем передаваемой информации является небольшим.

Одним из возможных вариантов абсолютно стойкого шифра является ***шифр Вернама*** (***одноразовый блокнот*** или одноразовая лента), запатентованный американским инженером Гильбертом Вернамом в 1917 г.

Получение шифротекста $\bar{y} = (y_1,..., y_n)$ по этому методу выполняется как сложение по модулю два открытого текста $\bar{x} = (x_1,..., x_n)$ с ключом $\bar{k} = (k_1,..., k_n)$:

$$y_i = x_i \oplus k_i; \qquad i = 1, 2, \ldots, n.$$

При этом ключ должен обладать тремя важными свойствами: быть истинно случайным; совпадать по размеру с заданным открытым текстом; применяться только один раз.

Для абсолютно стойкого шифра расстояние единственности $L_0 = \infty$. Для других шифров расстояние единственности может быть небольшим.

11.3. Контрольные вопросы

1. Какие выделяют составные части шифра при рассмотрении его формальной модели?
2. Что понимают под функциями шифрования и зашифрования?
3. Каково формальное определение шифра (криптографической системы)?
4. Какими свойствами должен обладать любой шифр?
5. Для чего применяется вероятностная модель шифра?
6. Какие шифры называют симметричными?
7. На какие группы разделят симметричные шифры по типу преобразования открытого текста?
8. В чём заключается различие между однозначными и многозначными заменами?
9. Какие выделяют шифры по числу символов шифротекста, соответствующих символу открытого текста?
10. Как работает шифр Виженера?
11. Что понимают под гаммой шифра?

12. В чём заключается особенность маршрутных перестановок?

13. На какие виды разделяют алгоритмы шифрования по числу ключей?

14. Что понимают под хэшированием?

15. На какие виды разделяют симметричные алгоритмы шифрования?

16. Какие алгоритмы шифрования называют асимметричными?

17. Что понимают под электронной цифровой подписью?

18. Как определяется априорная энтропия открытого текста?

19. Что называют энтропией ключа?

20. Какие ключи называют ложными?

21. Что понимают под расстоянием единственности?

22. В чём заключается правило Керкгоффа?

23. Какой шифр называют абсолютно стойким?

24. Какие требования предъявляются к абсолютно стойкому шифру?

12. БЛОЧНЫЕ АЛГОРИТМЫ ШИФРОВАНИЯ. ОСОБЕННОСТИ АЛГОРИТМА DES

12.1. Цель и задачи темы

Цель темы – изучить основные особенности блочных алгоритмов шифрования на примере алгоритма DES.

Основные задачи:

• рассмотреть особенности блочных алгоритмов шифрования, включая конструкцию Фейстеля;

• изучить основные особенности алгоритма шифрования DES;

• ознакомиться с режимами работы алгоритма DES.

Тема рассчитана на 4 часа.

12.2. Основные теоретические сведения

Блочные алгоритмы шифрования. Сеть Фейстеля.

Несмотря на активное развитие асимметричной криптографии, симметричные шифры остаются надёжным средством, на которое опираются все современные алгоритмы шифрования.

При передаче секретной информации её защищают с помощью криптографических систем смешанного типа. С помощью двухключевых асимметричных алгоритмов решается задача распределения ключей для симметричных шифров. А затем все передаваемые данные шифруются с помощью симметричного шифра.

Симметричные алгоритмы шифрования подразделяется на два вида:

• ***Блочные алгоритмы шифрования***, которые делят открытый текст на отдельные блоки, как правило, одинакового размера и оперируют с каждым из них с целью получения последовательности блоков шифрованного текста.

• ***Поточные алгоритмы шифрования***, которые производят обработку данных посимвольно или отдельными байтами без разделения на блоки. Поточные шифры, в отличие от блочных, обладают памятью предыдущего состояния шифра.

Преимущества блочных алгоритмов шифрования перед поточными алгоритмами:

• более высокая стойкость, вызванная большим объёмом алфавита, с которым работает блочный шифр;

• большинство блочных алгоритмов могут работать в режиме поточного шифра; поточные шифры такой возможностью не обладают.

Недостатки блочных шифров по сравнению с поточными:

• более сложная процедура оценки стойкости шифра;

• более высокая сложность аппаратной и программной реализации.

Для обеспечения высокой стойкости блочные шифры должны обладать свойствами рассеивания и перемешивания.

Рассеиванием (*diffusion*) называется влияние любого знака открытого текста на знаки шифротекста. Рассеивание позволяет скрыть влияние статистических свойств открытого текста на свойства шифротекста. Это свойство делает восстановление ключа трудной задачей.

При использовании рассеивания малейшее изменение открытого текста должно вызывать значительные изменения шифротекста. Такой шифр даст криптоаналитику гораздо меньше возможностей исследовать взаимосвязи между участками открытого и зашифрованного текстов. Само же свойство обычно называют лавинным эффектом.

Под ***перемешиванием*** (*confusion*) понимают использование таких преобразований, которые обеспечивают полную зависимость шифротекста от ключа. Перемешивание усложняет восстановление взаимосвязи статистических свойств открытого и шифрованного текстов при неизвестном ключе.

Для построения блочных алгоритмов шифрования, реализующих принципы рассеивания и перемешивания, часто используется конструкция, называемая сетью Фейстеля (*Feistel's network*).

Сеть Фейстеля представлена на рис. 12.1, где k_1, k_2, … – ключи шифрования, а f – криптографическая функция.

На каждом цикле одна из частей открытого текста преобразуется с помощью функции f_i и подключа k_i. Каждый такой шаг называется ***раундом шифрования***.

В сети Фейстеля блок M_{ji} открытого текста разбивается на две равные части (полублоки) – правую R (*Right*) и левую L (*Left*):

$$M_{ji} = (L_i, R_i).$$

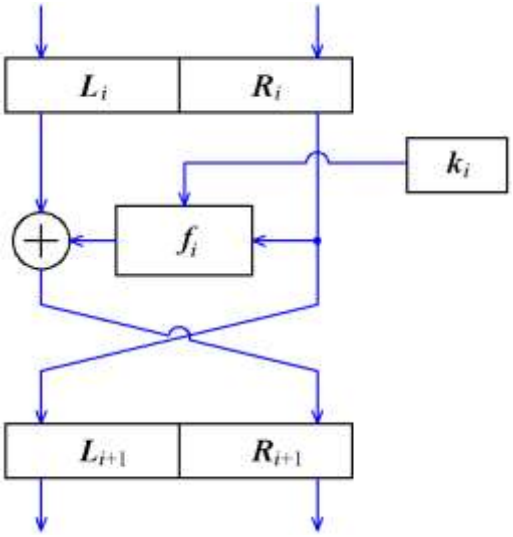


Рис. 12.1. Сеть Фейстеля с одним раундом

Раундовая функция шифрования:

$$f_i(M_{ji}) = (R_i, L_i \oplus f_i(R_i)).$$

Раундовая функция расшифрования:

$$f_i^{-1}(C_{ji}) = (R_i, L \oplus f_i(R_i)).$$

С помощью функции шифрования выполняется шифрование правой части. Результат шифрования суммируется по модулю 2 с левой частью. Затем левая и правая части меняются местами. На этом один раунд (цикл) шифрования заканчивается.

Процедура расшифровывания аналогична процедуре шифрования, но ключи k_i выбираются в обратном порядке.

Композиционный блочный шифр, использующий раундовые функции такого вида, называется композиционным шифром, построенным в соответствии с сетью Фейстеля.

Конструкция Фейстеля лежит в основе блочных алгоритмов шифрования DES, ГОСТ 28147-89, Blowfish и др.

Особенности алгоритма DES.

Разновидностью шифра Фейстеля является созданный в 1974 г. шифр DES (Data Encription Standart) и предложенный в качестве стандарта шифрования в государственных и частных организациях США.

Стандарт шифрования данных DES опубликован Национальным бюро стандартов США в 1977 г. В 1980 г. этот алгоритм был принят Национальным институтом стандартов и технологий США (НИСТ) в качестве стандарта шифрования данных для защиты важной, но несекретной информации в государственных и коммерческих организациях США.

Шифр DES имеет длину блока исходных данных равную 64 битам и ключ длиной 56 бит.

Процесс шифрования состоит из начальной перестановки битов входного блока, шестнадцати циклов шифрования и конечной перестановке битов (рис. 12.2).

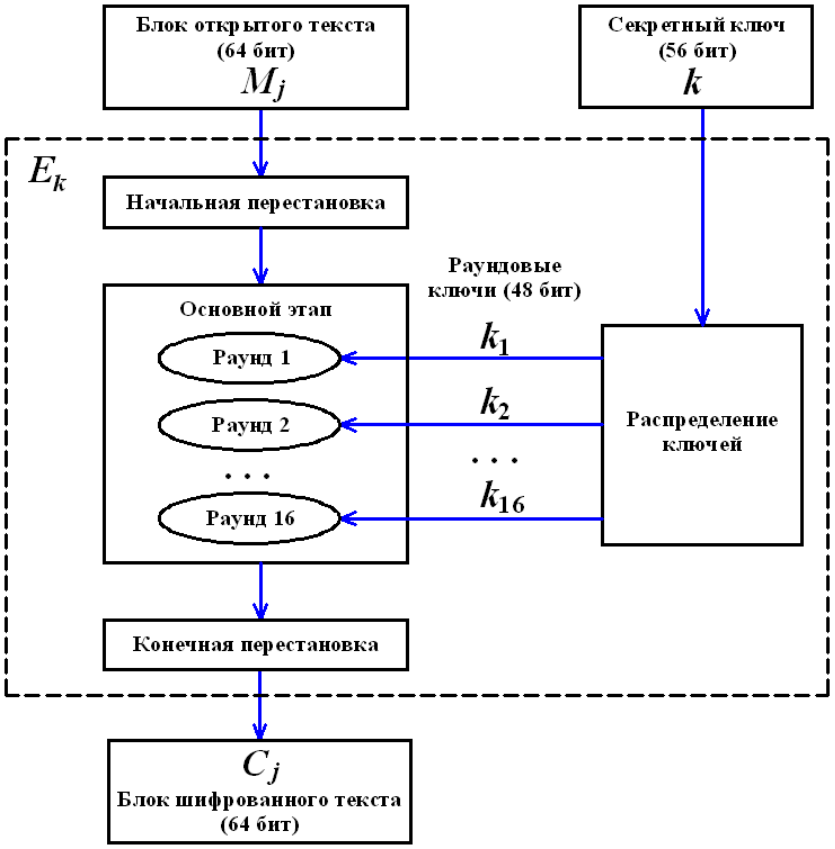


Рис. 12.2. Общая схема шифрования в алгоритме DES

Аргументами функции шифрования f являются 32-битовый вектор R_i и 48-битовый ключ k_i, который является результатом преобразования 56-битового ключа k.

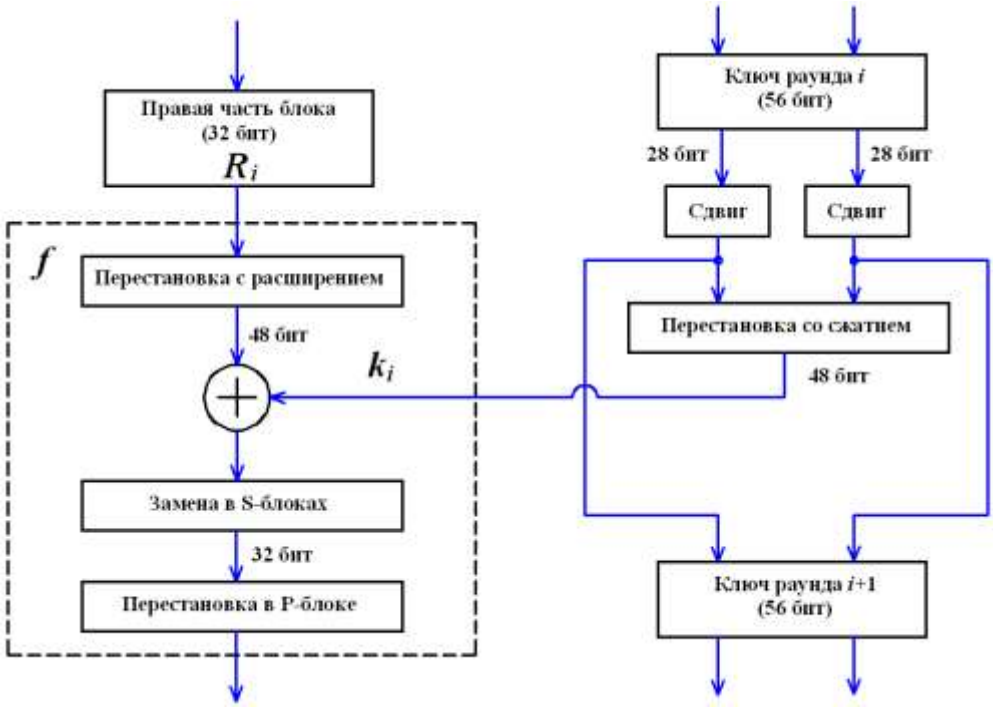


Рис.12.3. Раундовая функция шифрования и разворачивание ключа в алгоритме DES

В каждом раунде 56-битный ключ делится на две 28-битовые половинки. Затем половинки сдвигаются влево на один или два бита в зависимости от номера раунда. После сдвига определенным образом выбирается 48 из 56 битов. Так как при этом не только выбирается подмножество битов, но и изменяется их порядок, то эта операция называется «перестановка со сжатием». Ее результатом является набор из 48 битов. В среднем каждый бит исходного 56-битного ключа используется в 14 из 16 подключей, хотя не все биты используются одинаковое количество раз.

Режимы работы алгоритма DES. Понятие об алгоритмах 3DES и AES.

Один и тот же криптографический алгоритм может применяться для шифрования в различных режимах (простая замена,

гаммирование, и др.). Каждый режим шифрования имеет свои достоинства и недостатки, поэтому выбор режима зависит от конкретной ситуации.

DES поддерживает следующие режимы шифрования:

• электронная шифровальная книга (Electronic Codebook – ECB);

• сцепление шифрованных блоков (Cipher Block Chaining – CBC);

• шифрованная обратная связь (Cipher Feedback – CFB)

• обратная связь по выходу (Ouput Feedback – OFB);

• проскальзывание шифрованного текста (Cipher Text Stealing – CTS).

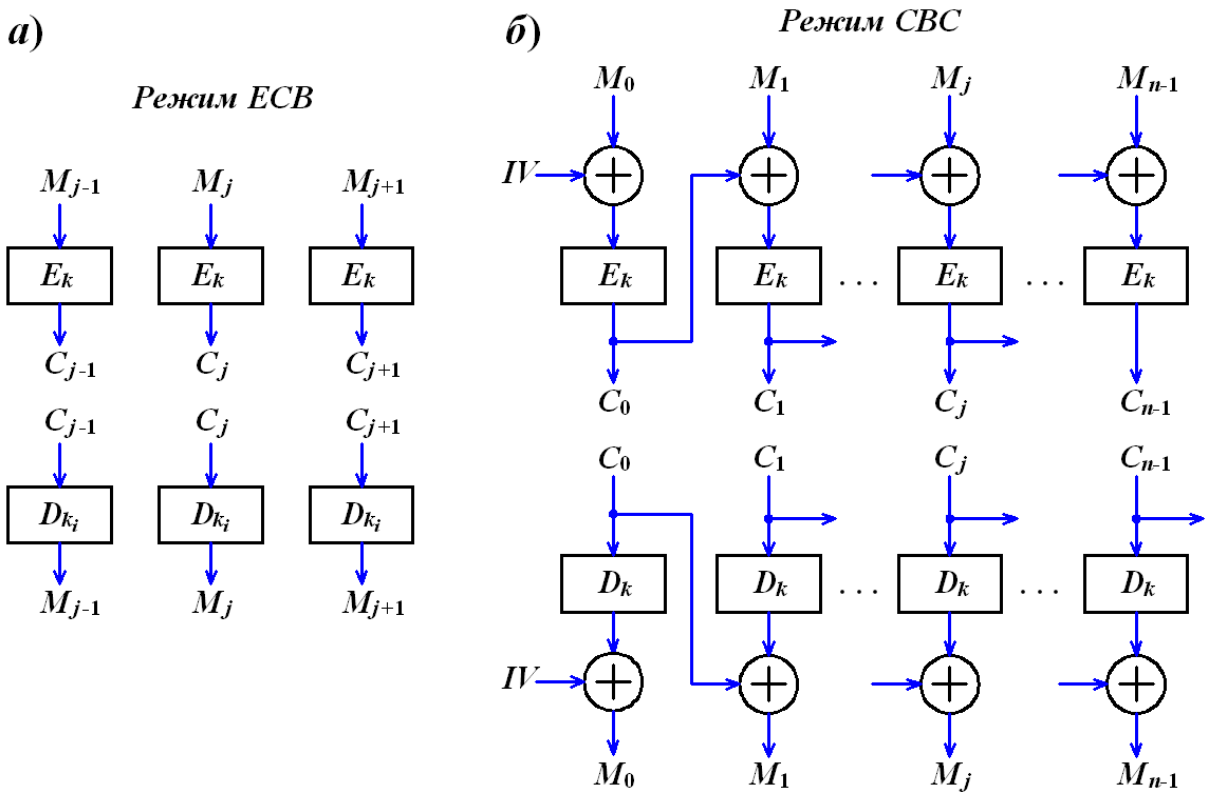


Рис. 12.4. Схема работы с блоками в DES: *а*) – режим ECB; *б*) – режим CBC

Шифрование в режиме ECB:

$$C_j = E_k(M_j).$$

Расшифрование в режиме ECB:

$$M_j = D_k(C_j).$$

Шифрование в режиме CBC:

$$C_0 = E_k(M_0 \oplus IV);$$

$$C_j = E_k(M_j \oplus C_{j-1}).$$

Расшифрование в режиме CBC:

$$M_0 = D_k(C_0) \oplus IV;$$

$$M_j = D_k(C_j) \oplus C_{j-1}.$$

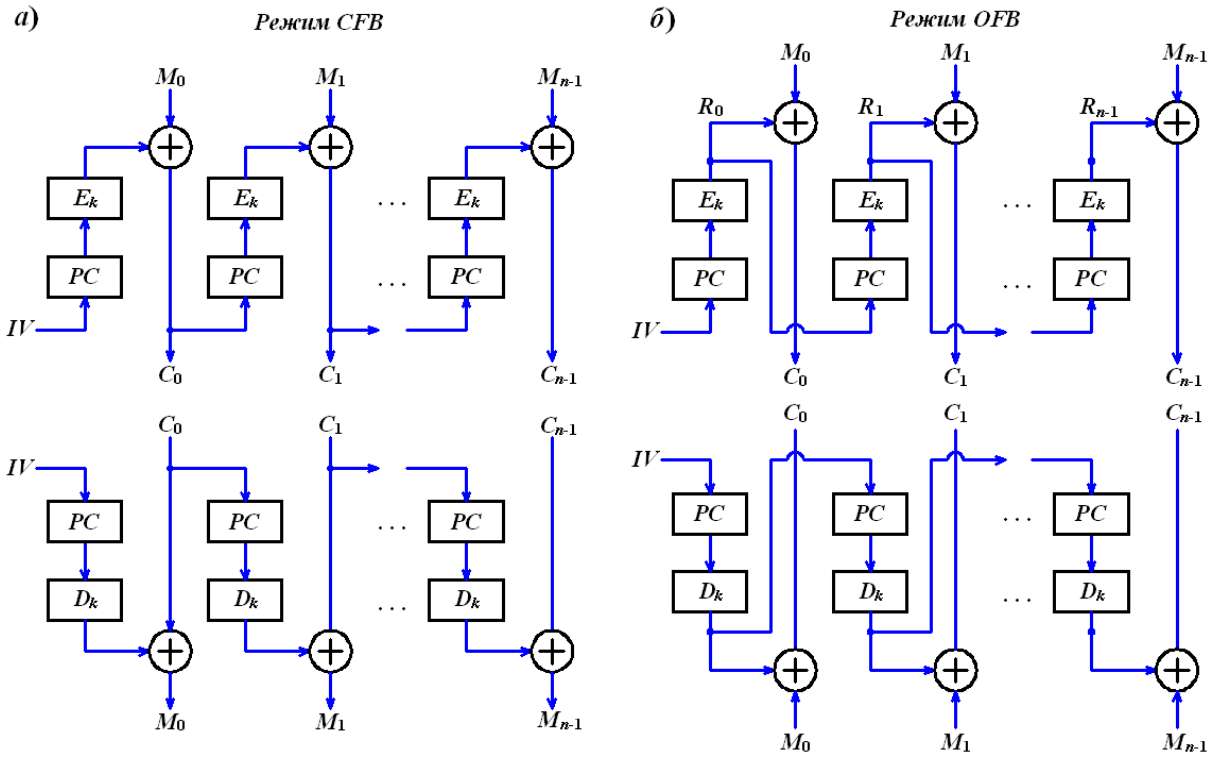


Рис. 12.5. Режимы работы DES: *а*) – режим CFB; *б*) – режим OFB

Шифрование в режиме CFB:

$$C_0 = M_0 \oplus E_k(IV);$$

$$C_j = M_j \oplus E_k(C_{j-1}).$$

Расшифрование в режиме CFB:

$$M_0 = C_0 \oplus D_k(IV);$$
$$M_j = C_j \oplus D_k(C_{j-1}).$$

Шифрование в режиме OFB:

$$C_0 = M_0 \oplus E_k(IV);$$
$$C_j = M_j \oplus E_k(R_{j-1}).$$

Расшифрование в режиме OFB:

$$M_0 = C_0 \oplus D_k(IV);$$
$$M_j = C_j \oplus D_k(R_{j-1}).$$

В качестве более надёжной альтернативы DES используется алгоритм ***«тройной» DES*** – 3DES (*Triple DES, TDES*).

3DES основывается на трёхкратном применении к каждому 64-битовому блоку открытого текста алгоритма DES с тремя разными ключами. При этом длина ключа увеличивается втрое, но также втрое увеличивается время, затрачиваемое на обработку.

Каждый блок открытого текста шифруется первым ключом, полученный результат расшифровывается вторым ключом, и, наконец, блок шифруют третьим ключом. Расшифрование осуществляется в обратном порядке.

Шифрование тройным DES:

$$C = E_{k3}(D_{k2}(E_{k1}(M))).$$

где k_1, k_2, k_3 – 56-битные ключи.

Расшифрование тройным DES:

$$M = D_{k1}(E_{k2}(D_{k3}(C))).$$

Официальным приемником DES, известным под названием AES (*Advanced Encryption Standard* – усовершенствованный стандарт шифрования) стал алгоритм Rijndael (произносится «райн дал»), разработанный бельгийскими криптографами Винсентом Райменом (*Vincent Rijmen*) и Джоан Деймен (*Joan Daemen*).

В отличие от DES, где размер ключа фиксирован и равняется 56 бит, в AES могут использоваться ключи размером 128, 192 и 256 бит. Кроме того, вместо фиксированного 64-битового блока

данных в DES, алгоритм AES может работать с блоками данных размерами 128, 192 и 256 бит.

12.3. Контрольные вопросы

1. В чём заключается отличие блочных алгоритмов шифрования от поточных?
2. Какими преимуществами и недостатками обладают блочные алгоритмы шифрования?
3. Что понимают под рассеиванием и перемешиванием в блочных шифрах?
4. Как работает сеть Фейстеля?
5. Из каких этапов состоит шифрование в алгоритме DES?
6. Как работает раундовая функция шифрования в алгоритме DES?
7. Каким образом происходит разворачивание ключа в алгоритме DES?
8. Какие режимы работы поддерживаются блочными алгоритмами шифрования?
9. Каковы особенности работы блочного шифра в режиме электронной шифровальной книги (ECB)?
10. В чём заключаются особенности режима сцепления шифрованных блоков (CBC)?
11. Как работает блочный алгоритм в режиме шифрованной обратной связи?
12. Что понимают под вектором инициализации (IV)?

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Печатные издания.

1. Аграновский А. В. Практическая криптография: алгоритмы и их программирование / А. В. Аграновский, Р. А. Хади. – М.: СОЛОН-Пресс, 2009. – 256 с.

2. Основы криптографии: учеб. пособие / А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин, А. В. Черемушкин. – 2-е изд., испр. и доп. – М.: Гелиос АРВ, 2002. – 480 с.

3. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М.: ДИАЛОГ–МИФИ, 2003. – 284 с.

4. Вернер М. Основы кодирования: учебник для ВУЗов. – М.: Техносфера, 2006. – 288 с.

5. Дмитриев В. И. Прикладная теория информации. – М.: Высш. шк., 1989. – 332 с.

6. Духин А. А. Теория информации. – М.: Гелиос АРВ, 2007. – 248 с.

7. Кудряшов Б. Д. Теория информации. – СПб.: Питер, 2009. – 188 с.

8. Лидовский В. В. Теория информации: учеб. пособие. – М.: Компания Спутник +, 2004. – 111 с.

9. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение: пер. с англ. – М.: Техносфера, 2005. – 320 с.

10. Орлов В.А., Филипов Л.И. Теория информации в упражнениях и задачах: учеб. пособие для втузов. – М.: Высш. школа, 1976. – 136 с.

11. Осипян В.О., Осипян К.В. Криптография в задачах и упражнениях. – М.: Гелиос АРВ, 2004. – 144 с.

12. Панин В. В. Введение в теорию кодирования: учеб. пособие для студентов вузов. – 2-е изд., испр. – М.: БИНОМ. Лаборатория знаний, 2011. – 322 с.

13. Панин В. В. Основы теории информации: учеб. пособие для студентов вузов. – 3-е изд., испр. – М.: БИНОМ. Лаборатория знаний, 2009.

14. Першин В. Т. Основы современной радиоэлектроники: учебное пособие / В. Т. Першин. – Ростов н/Д : Феникс, 2009. – 541 с.

15. Сэломон Д. Сжатие данных, изображения и звука. – М.: Техносфера, 2004. – 368 с.

16. Торстейнсон П. Криптография и безопасность в технологии .NET / П. Торстейнсон, Г. А. Ганеш; пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2013. – 497 с.

17. Хэмминг Р.В. Теория кодирования и теория информации: пер. с англ. – М.: Радио и связь, 1983. – 176 с.

18. Цымбал В. П. Задачник по теории информации и кодирования. Изд. 2-е – М: ЛЕНАНД, 2014. – 280 с.

19. Цымбал В. П. Теория информации и кодирование: учебник. – 4-е изд., перераб. и доп. – Киев: Вища школа, 1992. – 263 с.

Интернет-ресурсы.

20. www.intuit.ru/department/calculate/infotheory/ – учебный курс «Основы теории информации и криптографии», автор: В. В. Лидовский

21. Статья «Теория информации» в энциклопедии «Кругосвет»

22. www.compression.ru – сайт, посвященный технологиям сжатия данных

23. http://kunegin.narod.ru/ref3/code/index.htm – реферат по проблеме кодирования сообщений с исправлением ошибок

24. Статья «Обнаружение и исправление ошибок» в Википедии

25. Статья «Криптография» в Википедии

ПРИЛОЖЕНИЕ

П.1. Понятие события и его вероятности. Теоремы сложения и умножения вероятностей

Понятие события и вероятности события.

Событием (или «случайным событием») называется всякий факт, который в результате опыта может произойти или не произойти.

Вероятностью события называется численная мера степени объективной возможности этого события.

Вероятность события A обозначается через $p(A)$ или p.

Достоверным называется событие U, которое в результате опыта непременно должно произойти:

$$p(U) = 1.$$

Невозможным называется событие V, которое в результате опыта не может произойти:

$$p(V) = 0.$$

Вероятность любого события A заключена между нулем и единицей:

$$0 < p(A) < 1.$$

Полной группой событий называется несколько событий таких, что в результате опыта непременно должно произойти хотя бы одно из них.

Примеры событий, образующих полную группу:

• выпадение герба и цифры при бросании монеты;

• появление 1, 2, 3, 4, 5, 6 очков при бросании игральной кости;

• попадание и промах при выстреле и др.

Несколько событий в данном опыте называются ***несовместными***, если никакие два из них не могут появиться вместе.

Несколько событий в данном опыте называются ***равновозможными***, если есть основание считать, что ни одно из этих событий не является объективно более возможным, чем другое.

Если несколько событий образуют полную группу, несовместны и равновозможны, то они называются ***случаями***.

Случай называется ***благоприятным событию***, если появление этого случая влечет за собой появление события.

Если результаты опыта сводятся к схеме случаев, то вероятность события А вычисляется по формуле

$$p(A) = \frac{n(A)}{N};$$

где N — общее число случаев; $n(A)$ – число случаев, благоприятных событию A.

Суммой двух событий A и B называется событие C, состоящее в появлении хотя бы одного из событий A или B.

Произведением двух событий A и B называется событие C, состоящее в совместном появлении события A и события B.

Теоремы сложения и умножения вероятностей.

Вероятность суммы двух несовместных событий равна сумме вероятностей этих событий:

$$p(A + B) = p(A) + p(B).$$

В случае, когда события A и B совместны, вероятность их суммы выражается формулой

$$p(A + B) = p(A) + p(B) – p(AB),$$

где AB – произведение событий A и B.

Событие $\overline{A}$ называется ***противоположным*** событию A, если оно состоит в непоявлении события A.

Сумма вероятностей противоположных событий равна единице:

$$p(A) + p(\overline{A}) = 1.$$

Событие A называется ***независимым от события*** B, если вероятность события A не зависит от того, произошло событие B или нет.

Событие A называется ***зависимым от события*** B, если вероятность события A меняется в зависимости от того, произошло событие B или нет.

Условной вероятностью события A при наличии B называется вероятность события A, вычисленная при условии, что событие B произошло. Эта вероятность обозначается через $p(A|B)$.

Условие независимости события A от события B можно записать в виде

$$p(A|B) = p(A).$$

Вероятность произведения двух событий равна вероятности одного из них, умноженной на условную вероятность другого при наличии первого:

$$p(AB) = p(A)p(B|A)$$

или

$$p(AB) = p(B)p(A|B).$$

Для независимых событий A и B

$$p(AB) = p(A)p(B).$$

Для нескольких событий A_1, A_2, … , A_n теорема умножения вероятностей будет иметь следующий вид:

$$p(A_1A_2\ldots A_n) = p(A_1)p(A_2|A_1)p(A_3|A_1A_2)\ldots p(A_n|A_1A_2\ldots A_{n-1}).$$

П.2. Данные для расчета энтропии и количества информации

Таблица П.1

Значения $-p\log_2 p$

p	$-p\log_2 p$	p	$-p\log_2 p$	p	$-p\log_2 p$	p	$-p\log_2 p$	p	$-p\log_2 p$
0	—	0,20	0,4644	0,40	0,5288	0,60	0,4422	0,80	0,2575
0,01	0,0664	0,21	0,4728	0,41	0,5274	0,61	0,4350	0,81	0,2462
0,02	0,1129	0,22	0,4806	0,42	0,5256	0,62	0,4276	0,82	0,2348
0,03	0,1518	0,23	0,4877	0,43	0,5236	0,63	0,4199	0,83	0,2231
0,04	0,1858	0,24	0,4941	0,44	0,5211	0,64	0,4121	0,84	0,2113
0,05	0,2161	0,25	0,5000	0,45	0,5184	0,65	0,4040	0,85	0,1993
0,06	0,2435	0,26	0,5053	0,46	0,5153	0,66	0,3956	0,86	0,1871
0,07	0,2686	0,27	0,5100	0,47	0,512	0,67	0,3871	0,87	0,1748
0,08	0,2915	0,28	0,5142	0,48	0,5083	0,68	0,3783	0,88	0,1623

p	$-p\log_2 p$	p	$-p\log_2 p$	p	$-p\log_2 p$	p	$-p\log_2 p$	p	$-p\log_2 p$
0,09	0,3127	0,29	0,5179	0,49	0,5043	0,69	0,3694	0,89	0,1496
0,10	0,3322	0,30	0,5211	0,50	0,5000	0,70	0,3602	0,90	0,1368
0,11	0,3503	0,31	0,5238	0,51	0,4954	0,71	0,3508	0,91	0,1238
0,12	0,3671	0,32	0,5260	0,52	0,4906	0,72	0,3412	0,92	0,1107
0,13	0,3826	0,33	0,5278	0,53	0,4854	0,73	0,3314	0,93	0,0974
0,14	0,3971	0,34	0,5292	0,54	0,4800	0,74	0,3215	0,94	0,0839
0,15	0,4105	0,35	0,5301	0,55	0,4744	0,75	0,3113	0,95	0,0703
0,16	0,4230	0,36	0,5306	0,56	0,4684	0,76	0,3009	0,96	0,0565
0,17	0,4346	0,37	0,5307	0,57	0,4623	0,77	0,2903	0,97	0,0426
0,18	0,4453	0,38	0,5305	0,58	0,4558	0,78	0,2796	0,98	0,0286
0,19	0,4552	0,39	0,5298	0,59	0,4491	0,79	0,2687	0,99	0,0144

Таблица П.2

Распределение вероятностей букв русского алфавита в текстах

Буква	проб.	о	е	а	и	т	н	с
Вероятность	0,175	0,090	0,072	0,062	0,062	0,053	0,045	0,045
Буква	р	в	л	к	м	д	п	у
Вероятность	0,040	0,038	0,035	0,028	0,026	0,025	0,023	0,021
Буква	я	ы	з	ь, ъ	б	г	ч	й
Вероятность	0,018	0,016	0,016	0,014	0,014	0,013	0,012	0,010
Буква	х	ж	ю	ш	ц	щ	э	ф
Вероятность	0,009	0,007	0,006	0,006	0,004	0,003	0,003	0,002

Примечание.

Для русского алфавита, с учетом пробела, а также с учетом отсутствия связи между буквами $H_1 = 4{,}31$ бит/символ; с учетом двухбуквенных сочетаний $H_2 = 3{,}5$ бит/символ; с учетом трехбуквенных сочетаний $H_3 = 2{,}98$ бит/символ; для последовательностей букв неограниченной длины $H_\infty = 1{,}40$ бит/символ.

П.3. Таблица Виженера

Таблица П.3

Таблица Виженера для алфавита русского языка

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_
Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А
В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б
Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В
Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г
Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д
Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е
Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё
З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж
И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З
Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И
К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й
Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К
М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л
Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М
О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П
С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р
Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т
Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф
Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц
Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
Щ	Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
Ъ	Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
Ы	Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
Ь	Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы
Э	Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
Ю	Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
Я	_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю
_	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я

П.4. Вопросы и задачи к экзамену

В экзаменационном билете присутствуют два теоретических вопроса и одна практическая задача.

Для получения положительной оценки необходимо решить задачу и ответить хотя бы на один теоретический вопрос.

Список теоретических вопросов на экзамен.

1. Предметная область теории информации, её связь с другими науками
2. Понятия информации и неопределенности. Понятия сигнала, сообщения и данных
3. Система передачи информации, её основные элементы
4. Количественная оценка информации и неопределённости. Формула Хартли.
5. Количество информации и вероятность. Частное количество информации.
6. Понятие кода. Характеристики и виды кодов
7. Энтропия дискретного источника сообщений без памяти. Формула Шеннона
8. Математические свойства энтропии дискретных сообщений
9. Понятие о цепях Маркова. Марковские источники сообщений
10. Статистически связанные источники. Условная энтропия и ее свойства
11. Объединение источников. Энтропия объединения и ее свойства
12. Взаимная информация и ее свойства
13. Эффективность, избыточность и производительность источников сообщений
14. Виды каналов связи. Дискретные каналы связи без памяти
15. Пропускная способность дискретного канала связи без памяти
16. Равномерные и неравномерные коды. Кодовые деревья. Неравенство Крафта

17. Эффективное кодирование. Теорема Шеннона о кодировании источников
18. Классификация методов сжатия информации
19. Метод Шеннона-Фано
20. Метод Хаффмана
21. Метод арифметического кодирования
22. Словарные методы сжатия информации. Метод RLE
23. Особенности алгоритма LZ77
24. Особенности алгоритма LZ78
25. Помехоустойчивое кодирование. Теорема Шеннона о кодировании каналов
26. Классификация помехоустойчивых кодов
27. Особенности и характеристики блоковых корректирующих кодов
28. Математическое описание линейных блоковых кодов
29. Код Хэмминга
30. Порождающие матрицы линейных блоковых кодов
31. Проверочные матрицы линейных блоковых кодов
32. Декодирование линейных блоковых кодов
33. Понятие циклического кода. Полиномы и операции над ними
34. Построение циклических кодов. Порождающие полиномы
35. Проверочные полиномы. Декодирование циклических кодов
36. Методы защиты секретной информации. Предмет и задачи криптографии
37. Основные понятия криптографии. Виды криптографических атак
38. Формальные модели шифров
39. Классификация симметричных шифров
40. Классификация современных алгоритмов шифрования
41. Блочные алгоритмы шифрования. Сеть Фейстеля
42. Особенности блочного алгоритма DES
43. Режимы работы алгоритма DES
44. Энтропия открытого текста и ключа. Расстояние единственности
45. Правило Керкгоффа. Абсолютно стойкий шифр

Практические задачи на экзамен.

1. Определить энтропию, эффективность и избыточность источника сообщений без памяти, задаваемого ансамблем *S*. Определить среднее количество информации в сообщении $\bar{s}$, получаемом от данного источника.

2. Определить информационные потери ΔI при передаче n символов через дискретный канал связи, заданный матрицей переходных вероятностей.

3. Определить пропускную способность *C* дискретного канала связи, заданного матрицей переходных вероятностей. Каждый символ вырабатывается источником сообщений за время $\tau_{ср}$.

4. Произвести эффективное кодирование алфавита источника сообщений, задаваемого ансамблем *S*, с помощью метода Хаффмана. Построить кодовое дерево для полученного кода Хаффмана и определить его эффективность.

5. Для заданной модели источника без памяти произвести арифметическое кодирование последовательности символов. Декодировать полученный арифметический код.

6. Сжать заданное сообщение с помощью алгоритма LZ77 при известном объёме словаря и буфера. Определить объём сжатых данных. Декодировать полученные кодовые слова.

7. Перекодировать информационный вектор x в код Хэмминга. Задать одиночную ошибку в полученном кодовом векторе Хэмминга и произвести его декодирование по синдрому, исправив эту ошибку.

8. Перекодировать заданный информационный вектор x в линейный блоковый код на основе порождающей матрицы. Задать одиночную ошибку в полученном векторе линейного блокового кода и произвести его декодирование по синдрому.

9. Перекодировать заданный информационный вектор x в циклический код на основе порождающего полинома. Задать одиночную ошибку в полученном кодовом полиноме и произвести его декодирование.