Основы кодирования

Михаил Шихов
m.m.shihov@gmail.com

Лекция по дисциплине «информатика»
(15 июня 2016 г.)

Содержание

Определение информации

Definition (Юридическое)

Информация — это сведения (сообщения, данные) независимо от формы их представления[a].

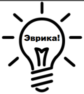
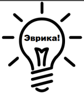
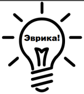
[a] №149-ФЗ от 27 июля 2006 г «Об информации, информационных технологиях и о защите информации»

Definition

Информация — это упорядоченная последовательность (цепочка) кодовых символов, принадлежащих конечному алфавиту. При этом каждый символ последовательности несёт определённую смысловую нагрузку.


Материально-энергетический носитель
Форма представления
Сигнал
Живучесть
Уникальность
Принадлежность
Отправитель (источник, создатель)
Информация
Количество
Структура
Получатель (приёмник, потребитель)
Адекватность
Способ кодирования
Избыточность
Отражаемый объект
Полнота
Релевантность
Своевременность
Конфиденциальность
Доступность
Целостность
Эргономичность
Назначение

Уровни доступа к информации

Носителя			
Средств взаимодействия			
Представления (код, формат)	слово word كلمة 字	NTFS FAT32 ExFAT *.txt *.png *.doc *.xls *.htm *.svg	
Семантики (понимания)			

Трудности оценки количества информации

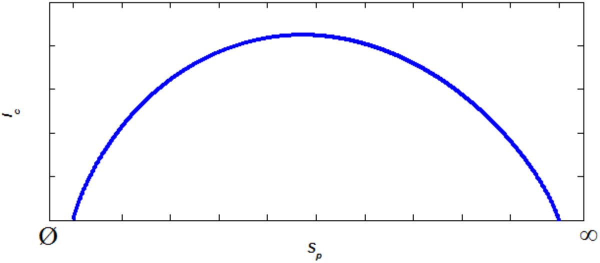
- Просто оценить количество явно заданной информации: достаточно посчитать количество кодовых символов последовательности.
- Сложно оценить необходимое количество информации для адекватного «отражения» исходного объекта.

Семантический подход к количественной оценке

Зависимость I_c в сообщении от тезауруса S_p получателя

$$I = I_s + I_c + I_n.$$

где I_s — известна, I_c — неизвестна и понятна, I_n — шум. $C = \frac{I_c}{I}$ называется коэффициентом содержательности информации.



Прагматический подход к количественной оценке

Оценка Александра Александровича Харкевича

$$I = \log_m \frac{p_2}{p_1} = \log_m p_2 - \log_m p_1, \tag{1}$$

где m — основание логарифма, определяющее единицы измерения ($m > 1$), p_1 — вероятность достижения потребителем цели до получения информации, p_2 — вероятность достижения потребителем цели после получения информации. Ценность информации в случае $p_1 > p_2 > 0$ положительна, в случае $0 < p_1 < p_2$ отрицательна, а в случае $p_1 = p_2$ равна нулю.

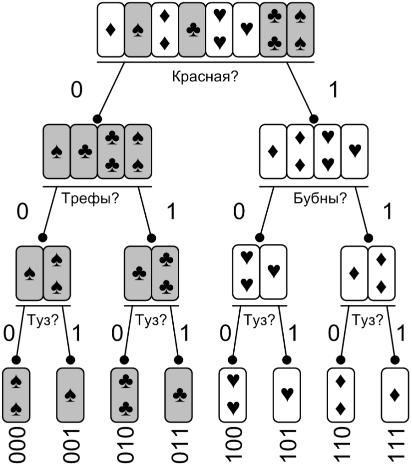
Синтаксический подход

Задача о картах (постановка)

Example

Имеется колода из восьми карт. По две карты (туз и двойка) каждой масти. Некто вытягивает наугад карту и готов честно отвечать только да или нет на любые задаваемые вопросы. Требуется минимальным количеством вопросов угадать вытащенную карту.

Задача о картах (решение)



Количественная оценка Ральфа Хартли

$m^H \geq N$

$$H = \log_m N,$$

где m — количество кодовых символов; N — количество состояний отражаемого объекта.

Example

В случае примера с картами: количество состояний $N = 8$, количество символов $m = 2$. Количество информации по Хартли: $H = \log_2 8 = 3$ бита.

Количественная оценка по Клоду Шеннону

Постулаты

Отражаемый объект — источник событий.

1. Количество информации есть непрерывная функция от вероятности события.
2. Количество информации I_i одиночного i-го события $s_i \in S$, $1 \leq i \leq N$ происходящего с вероятностью p_i, имеет положительное значение.

$$I_i \geq 0; I_i = I(p_i); \sum_{i=1}^{N} p_i = 1.$$

3. Количество информации I_{ij} двух независимых событий $s_i, s_j \in S$ с вероятностью $p_{ij} = p_i \cdot p_j$, равно сумме количеств информаций событий в отдельности: $I_{ij} = I_i + I_j; I(p_i \cdot p_j) = I(p_i) + I(p_j)$.

Количественная оценка по Клоду Шеннону

Зависимость информации от вероятности

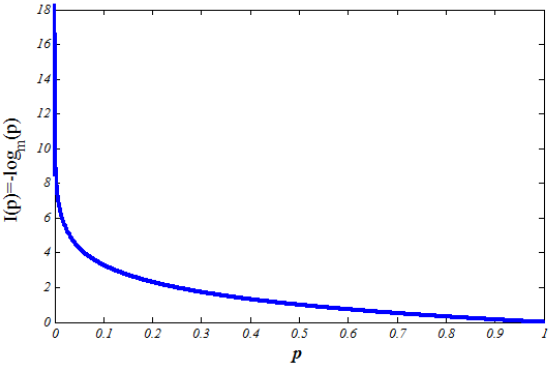
$$I(p) = -\log_m(p),$$

где $I(p)$ — информация события, происходящего с вероятностью p; m — количество кодовых символов.

Example (m определяет единицы измерения информации)

- $m = 2$. бит.
- $m = e$. нат.
- $m = 3$. трит.
- $m = 10$. дит.

Зависимость количества информации от вероятности



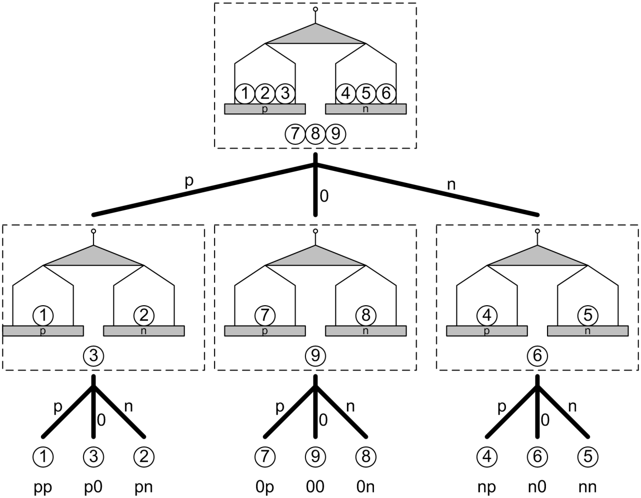
Задача о биллиардных шарах (постановка)

Example

Имеется восемь биллиардных шаров с номерами 1-8 соответственно. Все шары одинаковой массы, кроме одного, который тяжелее остальных. Имеются весы Фемиды (чашечные). Какое количество взвешиваний потребуется, чтобы определить номер тяжелого шара?

Задача о биллиардных шарах (решение)

Решение. $H = \log_3 9 = I(p) = -\log_3 \frac{1}{9} = 2$ трита



Энтропия

Мера информативности источника событий (сколько выдаёт I в среднем за раз)

$$E = -\sum_{i=1}^{N} p_i \cdot \log_m p_i .$$

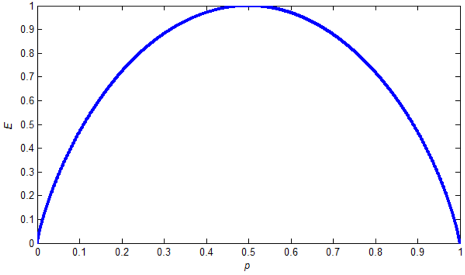


Рис.: Энтропия для источника с двумя состояниями

Война префиксов закончилась 19 марта 2005 года

Принят стандарт IEEE 1541. 1000 байт — 1 kB (килобайт), 1024 байт — 1KiB (кибибайт)

Множитель	СИ/SI	Множитель	IEEE 1541
$10^3 = 1000^1$	*kilo* (k) кило	$2^{10} = 1024^1$	*kibi* (Ki) киби
$10^6 = 1000^2$	*mega* (M) мега	$2^{20} = 1024^2$	*mebi* (Mi) меби
$10^9 = 1000^3$	*giga* (G) гига	$2^{30} = 1024^3$	*gibi* (Gi) гиби
$10^{12} = 1000^4$	*tera* (T) тера	$2^{40} = 1024^4$	*tebi* (Ti) теби
$10^{15} = 1000^5$	*peta* (P) пета	$2^{50} = 1024^5$	*pebi* (Pi) пеби
$10^{18} = 1000^6$	*exa* (E) экса	$2^{60} = 1024^6$	*exbi* (Ei) эксби
$10^{21} = 1000^7$	*zetta* (Z) зетта	$2^{70} = 1024^7$	*zebi* (Zi) зеби
$10^{24} = 1000^8$	*yotta* (Y) йотта	$2^{80} = 1024^8$	*yobi* (Yi) йоби

Кодирование

Definition

Кодирование — процесс перехода от источника событий к источнику информации. Т.е. сопоставление событиям цепочек из кодовых символов.

Некоторые назначения кодирования:

1. принципиальная возможность описания мира с помощью символов конечного алфавита;
2. устранение избыточности, сжатие информации, экономия памяти и снижение нагрузки на каналы передачи информации;
3. обеспечение устойчивости к помехам;
4. защита важных свойств информации (конфиденциальность, целостность, принадлежность и т.д.).

Формальное определение кодирования

Definition

Дано:

- Алфавит событий: $S = \{s_1, \dots, s_N\}$;
- Алфавит кодовых символов: $T = \{t_1, \dots, t_m\}$;

Требуется задать отображение $\delta : S \to T^+$ (таблицу кодов, схему кодирования):

$$\delta = \langle s_1 \mapsto \omega_1, \dots, s_N \mapsto \omega_N \rangle,$$

где $\omega_i = t_{i_1} \cdots t_{i_{k_i}}$, причем слово $\varsigma_j = s_{j_1} \cdots s_{j_t}$ будет кодироваться символами кодового алфавита как $\varsigma_j = \omega_{j_1} \cdots \omega_{j_t}$. Множество кодовых слов ω_i, соответствующих s_i называется множеством элементарных кодов:

$$\Omega = \{\omega_1, \dots, \omega_N\}.$$

Примеры кодирования

Example (Неоднозначное декодирование. δ — не биекция)

$S = \{A, B, C, D, E, F, G, H\}$; $T = \{0, 1\}$; $\delta = \langle A \to 0, B \to 1, C \to 10, D \to 11, E \to 100, F \to 101, G \to 110, H \to 111\rangle$.

- Кодирование однозначно: $ABAB \mapsto 0101$.
- Декодирование нет: 0101 разделяется на слова $ABAB$, AF и ACB.

Example (Однозначное декодирование. δ — биекция)

$S = \{A, B, C, D, E, F, G, H\}$; $T = \{0, 1\}$; $\delta = \langle A \mapsto 000, B \mapsto 001, C \mapsto 010, D \mapsto 011, E \mapsto 100, F \mapsto 101, G \mapsto 110, H \mapsto 111\rangle$.

- Кодирование: $ABBA \mapsto 000111000111$.
- Декодирование: $000111000111 \mapsto ABBA$.

Схемы кодирования

Definition

Схема кодирования δ является разделимой, если любое слово ς, составленное из элементарных кодов ω_i единственным образом разлагается на элементарные коды.

Разделимая схема допускает декодирование. Важным частным случаем разделимых схем являются префиксные схемы.

Definition

Схема называется префиксной, если ни один элементарный код ω_i из множества Ω не является префиксом[a] другого кода из того же множества.

[a] Префиксом слова ω называется слово ω_1, если $\omega = \omega_1\omega_2$

«Равномерное» кодирование

Наиболее простым вариантом кодирования является равномерное кодирование, когда все элементарные коды равной длины.
Для кодирования N событий требуется использовать цепочки длины

$$l(n) = \lceil \log_m(n) \rceil,$$

где m — количество кодовых символов; $\lceil X \rceil$ — наименьшее целое, большее или равное X.
Эта же оценка на основе постулатов Шеннона:

$$l(n) = \left\lceil -\log_m \left(\frac{1}{n} \right) \right\rceil = \lceil \log_m(n) \rceil.$$

Равномерное кодирование

Example

В соревновании учавствуют 33 спортсмена. Для регистрации пересечения финишной черты каждому спортсмену выдается радио-брелок. В момент пересечения финишной черты спортсменом, брелок передает двоичный код для идентификации спортсмена. Все брелки передают код одинаковой длины. Какое минимально необходиоме количество бит в общем случае должен передать брелок?

Равномерное кодирование

Example

В соревновании учавствуют 33 спортсмена. Для регистрации пересечения финишной черты каждому спортсмену выдается радио-брелок. В момент пересечения финишной черты спортсменом, брелок передает двоичный код для идентификации спортсмена. Все брелки передают код одинаковой длины. Какое минимально необходиоме количество бит в общем случае должен передать брелок?

Решение.

$\lceil \log_2(33) \rceil = 6.$ $\square$

Сигнал

Definition

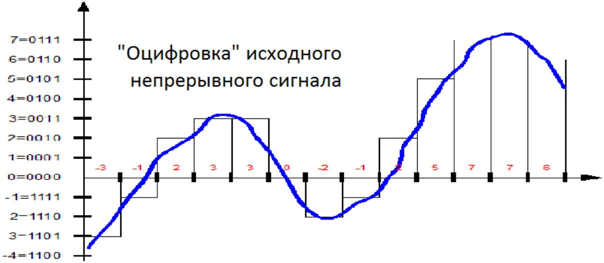
Сигнал — это изменение (во времени или пространстве) физической величины, несущее информацию, т.е. способ, позволяющий фиксировать символ в материально-энергетическом носителе

Выделяют два вида сигналов:

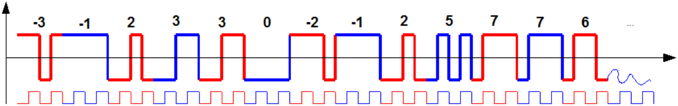
1. аналоговые;
2. цифровые.

Аналоговый сигнал

Цифровой сигнал



Цифровой сигнал (цифры 0 и 1 задаются "*высоким*" и "*низким*" значениями некоторой физической величины)



Информативность источника событий

Источнику событий после кодирования соответствует источник информации, выдающий коды событий. Оценку информативности источника событий дает величина, называемая энтропией источника событий:

$$E = -\sum_{i=1}^{N} p_i \cdot \log_m p_i, \tag{2}$$

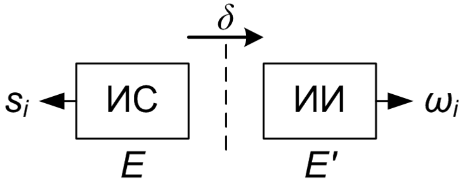
где p_i — вероятность i-го события $s_i \in S$ на выходе источника событий, m — количество кодовых символов, N — количество событий $N = |S|$.

Информативность источника информации

Так как вероятности появления кодов событий останутся прежними, то энтропия источника информации E' будет равна

$$E' = \sum_{i=1}^{n} p_i \cdot l_i, \tag{3}$$

где l_i — длина кода ω_i для i-го события.



Задача оптимального кодирования

Аксиома

Энтропия источника информации всегда больше энтропии отражаемого источника событий.

Задача оптимального кодирования: максимально приблизить энтропию источника информации к энтропии источника событий.

Оценка оптимальности кодирования

Пусть имеется источник событий s_i, о вероятности появления которых на его выходе известно следующее:

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1

Энтропия источника событий, формула (2), составляет:

$$E = -(0.5 \cdot \log_2 0.5 + 0.3 \cdot \log_2 0.3 + 0.1 \cdot log_2 0.1 + 0.1 \cdot log_2 0.1) \approx$$
$$\approx (0.5 + 0.521089678 + 0.332192809 + 0.332192809) \approx$$
$$\approx 1.685475297 \text{ бит.}$$

Оценка оптимальности кодирования

Для равномерного кодирования битами может быть получен такой вариант:

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i	00	01	10	11

Энтропия данного источника информации составит, по формуле (3):

$$E' = (0.5 \cdot 2 + 0.3 \cdot 2 + 0.1 \cdot 2 + 0.1 \cdot 2) = 2 \text{ бита.}$$

Оценка оптимальности кодирования

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i	0	10	110	111

Так же как и предыдущая, эта схема префиксная и разделимая, но неравномерная. Энтропия источника информации теперь составляет

$$E' = (0.5 \cdot 1 + 0.3 \cdot 2 + 0.1 \cdot 3 + 0.1 \cdot 3) = 1.7 \text{ бита.}$$

Оценка оптимальности кодирования

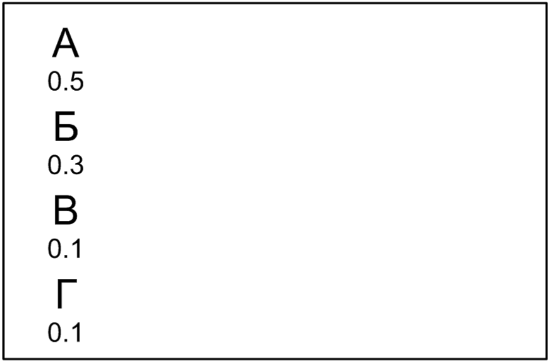
Представленные источники эквиваленты. Если запустить источник информации на выдачу, например, 100 кодов событий, то первый вариант кодирования выдаст цепочку длины ≈ 200, а второй — ≈ 170 бит.

Алгоритм Хаффмана для $m = 2$

1. События сортируются по убыванию вероятности.
2. Два события с минимальными вероятностями объединяются в одно составное событие с суммарной вероятностью исходных. При этом одно из исходных событий помечается кодовым символом 0, а второе — символом 1. Исходные события исключаются из множества событий, вместо них остается одно составное.
3. Шаги 1 и 2 последовательно повторяются до тех пор, пока все события не склеятся в единственное составное событие, вероятность которого равна 1. После этого кодовое слово ω_i для исходного события s_i есть цепочка из кодовых символов, которыми помечены все составные события от корня до s_i.

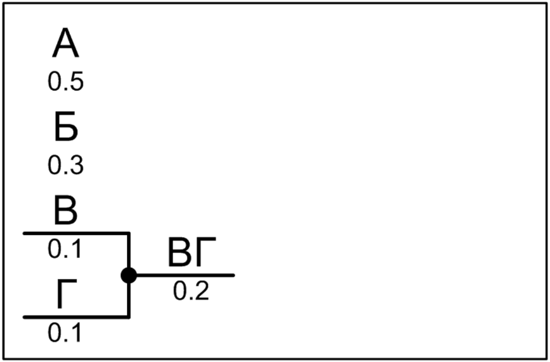
Оптимальное кодирование по Хаффману

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i				



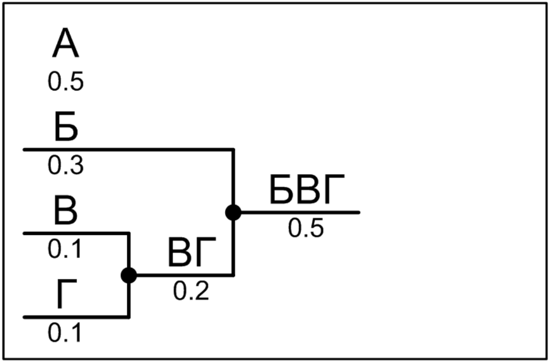
Оптимальное кодирование по Хаффману

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i				



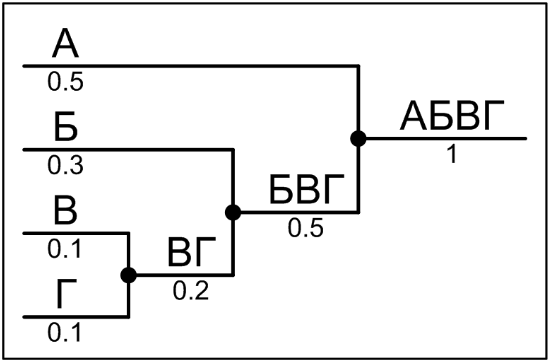
Оптимальное кодирование по Хаффману

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i				



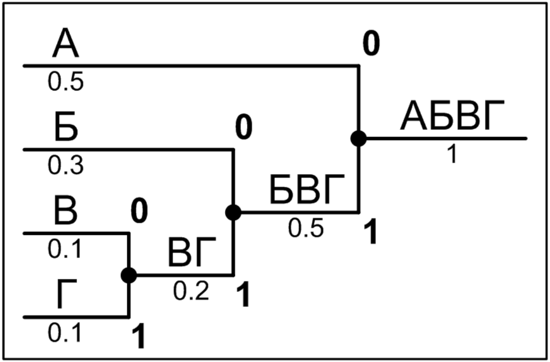
Оптимальное кодирование по Хаффману

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i				



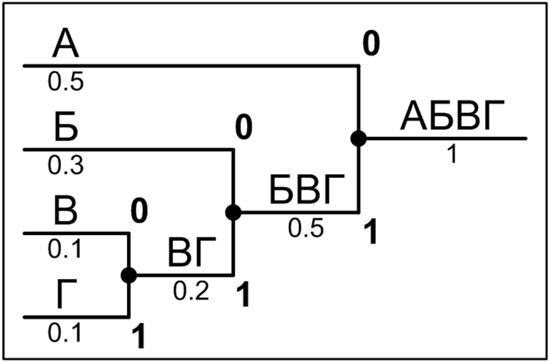
Оптимальное кодирование по Хаффману

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i				



Оптимальное кодирование по Хаффману

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i	0	10	110	111

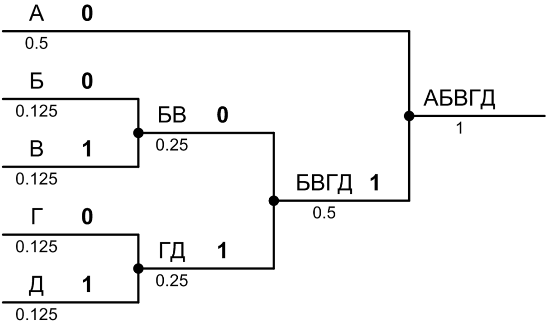


Оптимальное кодирование по Хаффману (задача)

Событие s_i	А	Б	В	Г	Д
Вероятность p_i события s_i	0.5	0.125	0.125	0.125	0.125
Код события ω_i					

Оптимальное кодирование по Хаффману (задача)

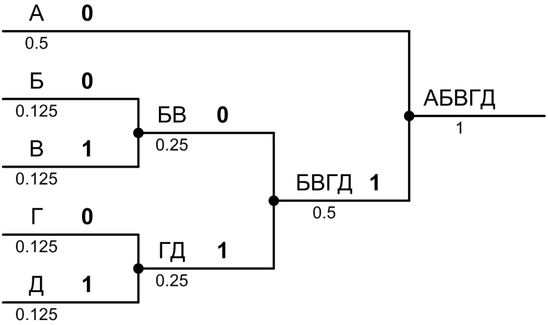
Событие s_i	А	Б	В	Г	Д
Вероятность p_i события s_i	0.5	0.125	0.125	0.125	0.125
Код события ω_i	0	100	101	110	111



$10001101110111 \to$

Оптимальное кодирование по Хаффману (задача)

Событие s_i	А	Б	В	Г	Д
Вероятность p_i события s_i	0.5	0.125	0.125	0.125	0.125
Код события ω_i	0	100	101	110	111



$10001101110111 \to$ БАГДАД

Алгоритм Фано для $m = 2$

1. Исходный массив событий, сортируется в порядке убывания вероятностей.
2. Массив разбивается на две части, так, чтобы разница сумм вероятностей событий каждой части была минимальна. Первый кодовый символ элементарного кода ω_i находится так: для всех событий левой части разбитого массива кодовый символ будет 0, а для всех событий правой части — 1.
3. Второй и последующие кодовые символы определяется так: каждая часть разбитого исходного массива, в которой более одного события, становится исходным массивом, и её разбиение выполняется так же, как исходного массива (шаг 2).

Оптимальное кодирование по алгоритму Фано

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i				

s_i	p_i
А	0.5
Б	0.3
В	0.1
Г	0.1

Оптимальное кодирование по алгоритму Фано

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i				

s_i	p_i	ω_i
А	0.5	0
Б	0.3	1
В	0.1	1
Г	0.1	1

Оптимальное кодирование по алгоритму Фано

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i				

s_i	p_i	ω_i	
А	0.5	0	
Б	0.3	1	0
В	0.1	1	1
Г	0.1	1	1

Оптимальное кодирование по алгоритму Фано

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i				

s_i	p_i	ω_i		
А	0.5	0		
Б	0.3	1	0	
В	0.1	1	1	0
Г	0.1	1	1	1

Оптимальное кодирование по алгоритму Фано

Событие s_i	А	Б	В	Г
Вероятность p_i события s_i	0.5	0.3	0.1	0.1
Код события ω_i	0	10	110	111

s_i	p_i	ω_i		
А	0.5	0		
Б	0.3	1	0	
В	0.1	1	1	0
Г	0.1	1	1	1

Оптимальное кодирование по алгоритму Фано (задача)

s_i	А	Б	В	Г	Д	Е	Ж
p_i	0.135	0.24	0.25	0.125	0.0635	0.124	0.0625
ω_i							

Оптимальное кодирование по алгоритму Фано (задача)

s_i	А	Б	В	Г	Д	Е	Ж
p_i	0.135	0.24	0.25	0.125	0.0635	0.124	0.0625
ω_i							

s_i	p_i	ω_i			
В	0.25	0	0		
Б	0.24	0	1		
А	0.135	1	0	0	
Г	0.125	1	0	1	
Е	0.124	1	1	0	
Д	0.0635	1	1	1	0
Ж	0.0625	1	1	1	1

Оптимальное кодирование по алгоритму Фано (задача)

s_i	А	Б	В	Г	Д	Е	Ж
p_i	0.135	0.24	0.25	0.125	0.0635	0.124	0.0625
ω_i	100	01	00	101	1110	110	1111

s_i	p_i	ω_i			
В	0.25	0	0		
Б	0.24	0	1		
А	0.135	1	0	0	
Г	0.125	1	0	1	
Е	0.124	1	1	0	
Д	0.0635	1	1	1	0
Ж	0.0625	1	1	1	1

Сжатие

Кодирование с целью сжатия (или просто сжатие) ставит себе в задачу уменьшить количество информации, не теряя (или оставаясь в допустимых рамках) при этом свойство адекватности отражаемому объекту. В случае сжатия, события s_i уже представляют собой слова в алфавите T (коды). При сжатии информация перекодируется в том же алфавите T.

Классы алгоритмов сжатия:

- сжатие с потерями (адекватности);
- сжатие без потерь.

Алгоритм Лемпела-Зива (LZ)

Кодирование

1. В словарь нулевым элементом помещается пустая цепочка ε. Пустое слово ε не содержит букв, и для любого слова ω справедливо $\omega = \varepsilon\omega = \omega\varepsilon$.
2. От исходной цепочки t отделяется слово ωa, где ω — максимально длинное слово из словаря, a — расширяющая буква. Т.е. $t = \omega a t'$.
3. В конец словаря добавляется новое слово ωa. К коду c добавляется пара $\langle i_\omega, a\rangle$, где i_ω — индекс слова ω в словаре. От исходного текста отделяется слово ωa: $t = t'$.
4. Пункты 2-3 последовательно повторяются до тех пор, пока в тексте t остается хоть одна буква.

В результате получается код $c = \langle i_1, a_1\rangle \cdots \langle i_n, a_n\rangle$.

Алгоритм Лемпела-Зива

Пример сжатия текста: «АБАКАНКАНКАНКИАНКИН»

i	t	ωa	$c = \langle i_\omega, a\rangle$
		$0 \to \varepsilon$	
1	ε**А**БАКАНКАНКАНКИАНКИН	$1 \to$ А	⟨0,А⟩
2	ε**Б**АКАНКАНКАНКИАНКИН	$2 \to$ Б	⟨0,Б⟩
3	**АК**АНКАНКАНКИАНКИН	$3 \to$ АК	⟨1,К⟩
4	**АН**КАНКАНКИАНКИН	$4 \to$ АН	⟨1,Н⟩
5	ε**К**АНКАНКИАНКИН	$5 \to$ К	⟨0,К⟩
6	**АНК**АНКИАНКИН	$6 \to$ АНК	⟨4,К⟩
7	**АНКИ**АНКИН	$7 \to$ АНКИ	⟨6,И⟩
8	**АНКИН**	$8 \to$ АНКИН	⟨7,Н⟩

$$\overbrace{\varepsilon}^{0}\ \underset{\langle 0,\text{А}\rangle}{\overbrace{\text{А}}^{1}}\ \underset{\langle 0,\text{Б}\rangle}{\overbrace{\text{Б}}^{2}}\ \underset{\langle 1,\text{К}\rangle}{\overbrace{\text{АК}}^{3}}\ \underset{\langle 1,\text{Н}\rangle}{\overbrace{\text{АН}}^{4}}\ \underset{\langle 0,\text{К}\rangle}{\overbrace{\text{К}}^{5}}\ \underset{\langle 4,\text{К}\rangle}{\overbrace{\text{АНК}}^{6}}\ \underset{\langle 6,\text{И}\rangle}{\overbrace{\text{АНКИ}}^{7}}\ \underset{\langle 7,\text{Н}\rangle}{\overbrace{\text{АНКИН}}^{8}}$$

Дать оценку длин кода и текста

Алгоритм Лемпела-Зива
Задание

Сжать текст:

«тартарарамитамтамывтартарарах»

Алгоритм Лемпела-Зива
Задание

Сжать текст:

«тартарарамитамтамывтартарарах»

0	1	2	3	4	5	6	7	8	9	10	11	12	13
ε	т	а	р	та	ра	рам	и	там	тамы	в	тар	тара	рах
	⟨0,т⟩	⟨0,а⟩	⟨0,р⟩	⟨1,а⟩	⟨3,а⟩	⟨5,м⟩	⟨0,и⟩	⟨4,м⟩	⟨8,ы⟩	⟨0,в⟩	⟨4,р⟩	⟨11,а⟩	⟨5,х⟩

Код:

$$\langle 0, \text{т}\rangle, \langle 0, \text{а}\rangle, \langle 0, \text{р}\rangle, \langle 1, \text{а}\rangle, \langle 3, \text{а}\rangle, \langle 5, \text{м}\rangle, \langle 0, \text{и}\rangle,$$
$$\langle 4, \text{м}\rangle, \langle 8, \text{ы}\rangle, \langle 0, \text{в}\rangle, \langle 4, \text{р}\rangle, \langle 11, \text{а}\rangle, \langle 5, \text{х}\rangle$$

Алгоритм Лемпела-Зива (LZ)

Декодирование

1. В словарь нулевым элементом помещается пустая цепочка ε. Текст t не содержит букв: $t = \varepsilon$.
2. От исходного кода c отделяется пара $\langle i, a \rangle$, в словарь добавляется слово $\omega_i a$, где ω_i — i-е слово из словаря. Восстанавливается текст $t = t\omega_i a$.
3. Пункт 2 последовательно повторяется до тех пор, пока в коде c остается хоть одна пара.

Алгоритм Лемпела-Зива

Пример декодирования «0А,0Б,1К,1Н,0К,4К,6И,7Н»

i	$c = \langle i_\omega, a \rangle$	ωa	t
		$0 \to \varepsilon$	
1	⟨0,А⟩	$1 \to$ А	ε**А**
2	⟨0,Б⟩	$2 \to$ Б	Аε**Б**
3	⟨1,К⟩	$3 \to$ АК	АБ**АК**
4	⟨1,Н⟩	$4 \to$ АН	АБАК**АН**
5	⟨0,К⟩	$5 \to$ К	АБАКАНε**К**
6	⟨4,К⟩	$6 \to$ АНК	АБАКАНК**АНК**
7	⟨6,И⟩	$7 \to$ АНКИ	АБАКАНКАНК**АНКИ**
8	⟨7,Н⟩	$8 \to$ АНКИН	АБАКАНКАНКАНКИ**АНКИН**

$$\overbrace{\varepsilon}^{0}\ \underset{\text{А}}{\overbrace{\langle 0, \text{А}\rangle}^{1}}\ \underset{\text{Б}}{\overbrace{\langle 0, \text{Б}\rangle}^{2}}\ \underset{\text{АК}}{\overbrace{\langle 1, \text{К}\rangle}^{3}}\ \underset{\text{АН}}{\overbrace{\langle 1, \text{Н}\rangle}^{4}}\ \underset{\text{К}}{\overbrace{\langle 0, \text{К}\rangle}^{5}}\ \underset{\text{АНК}}{\overbrace{\langle 4, \text{К}\rangle}^{6}}\ \underset{\text{АНКИ}}{\overbrace{\langle 6, \text{И}\rangle}^{7}}\ \underset{\text{АНКИН}}{\overbrace{\langle 7, \text{Н}\rangle}^{8}}$$

Алгоритм Лемпела-Зива
Задание

Восстановить текст из кода:

$$\langle 0, \text{в} \rangle, \langle 0, \text{о} \rangle, \langle 0, \text{т} \rangle, \langle 1, \text{а} \rangle, \langle 0, \text{м} \rangle, \langle 4, \text{р} \rangle, \langle 6, \text{ы} \rangle, \langle 2, \text{т} \rangle, \langle 6, \text{ы} \rangle$$

Алгоритм Лемпела-Зива
Задание

Восстановить текст из кода:

$$\langle 0, \text{в}\rangle, \langle 0, \text{о}\rangle, \langle 0, \text{т}\rangle, \langle 1, \text{а}\rangle, \langle 0, \text{м}\rangle, \langle 4, \text{р}\rangle, \langle 6, \text{ы}\rangle, \langle 2, \text{т}\rangle, \langle 6, \text{ы}\rangle$$

$$\overbrace{\underset{}{\varepsilon}}^{0}\ \overbrace{\underset{\text{в}}{\langle 0, \text{в}\rangle}}^{1}\ \overbrace{\underset{\text{о}}{\langle 0, \text{о}\rangle}}^{2}\ \overbrace{\underset{\text{т}}{\langle 0, \text{т}\rangle}}^{3}\ \overbrace{\underset{\text{ва}}{\langle 1, \text{а}\rangle}}^{4}\ \overbrace{\underset{\text{м}}{\langle 0, \text{м}\rangle}}^{5}\ \overbrace{\underset{\text{вар}}{\langle 4, \text{р}\rangle}}^{6}\ \overbrace{\underset{\text{вары}}{\langle 6, \text{ы}\rangle}}^{7}\ \overbrace{\underset{\text{от}}{\langle 2, \text{т}\rangle}}^{8}\ \overbrace{\underset{\text{вары}}{\langle 6, \text{ы}\rangle}}^{9}$$

Текст:

«вотвамварварыотвары»

Свойства информации с точки зрения её защиты

- целостность:
- конфиденциальность:
- принадлежность:
- доступность:

Свойства информации с точки зрения её защиты

- целостность: контрольные суммы; корректирующие коды;
- конфиденциальность: шифрование; скрытая передача;
- принадлежность: цифровая подпись;
- доступность: надежность информационных систем.

Классификация ошибок

Ошибки, возникающие в цифровом (двоичном) канале могут быть следующими:

- замещения кодового символа;
- вставка кодового символа;
- выпадение кодового символа.

Далее рассматриваются только ошибки замещения. Существуют две стратегии защиты от ошибок замещения:

- с обнаружением и запросом на повторную передачу (ARQ — Automatic Repeat Request);
- с обнаружением и непосредственным исправлением на стороне получателя (FEC — Forward Error Correction).

ARQ

Примером стратегии ARQ может считаться контроль по четности (нечетности).

$$p_{\text{чётн}} = d_{n-1} \oplus \ldots \oplus d_1 \oplus d_0,$$
$$p_{\text{нечётн}} = d_{n-1} \oplus \ldots \oplus d_1 \oplus d_0 \oplus 1.$$

FEC

Можно кодировать каждый бит исходной последовательности по схеме

$$\delta = \{0 \mapsto 000, 1 \mapsto 111\},$$

а декодировать по схеме

$$\begin{aligned}\delta' = \{&000 \mapsto 0, 001 \mapsto 0, 010 \mapsto 0, 100 \mapsto 0,\\ &111 \mapsto 1, 110 \mapsto 1, 101 \mapsto 1, 011 \mapsto 1\},\end{aligned}$$

Example

Пусть передается слово 101. Кодируется: 111000111. Поступает в канал. Возникает одиночная ошибка: $11\boxed{0}000111$. Декодируется: 101. При этом декодер обнаруживает и исправляет одиночную ошибку. □

FEC

Пример кодирования «утроением»

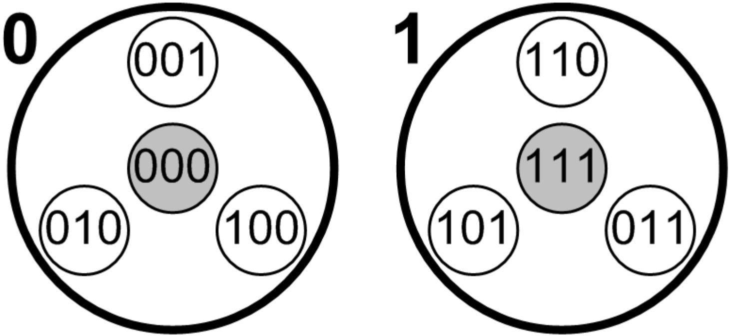
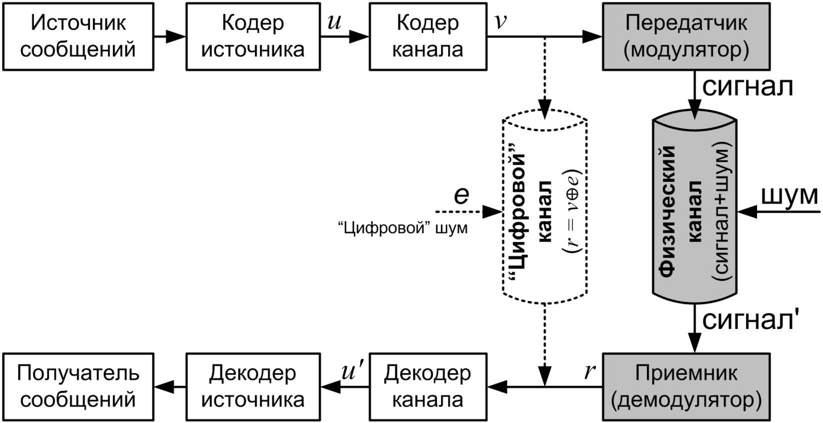
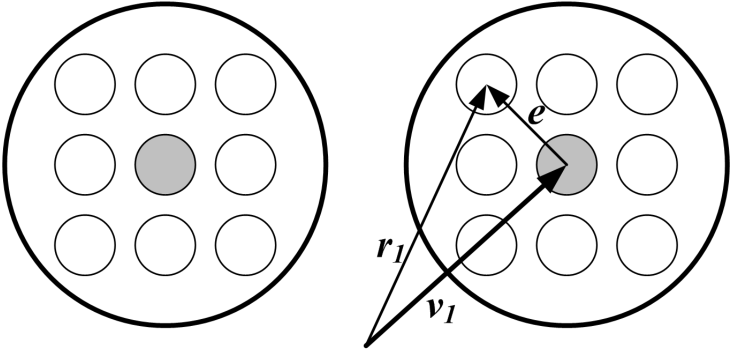


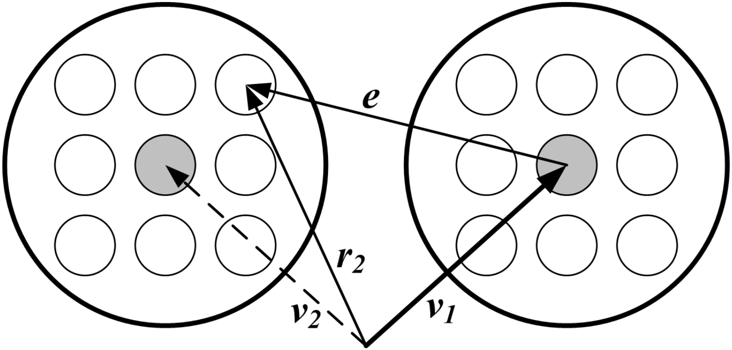
Схема канала передачи данных



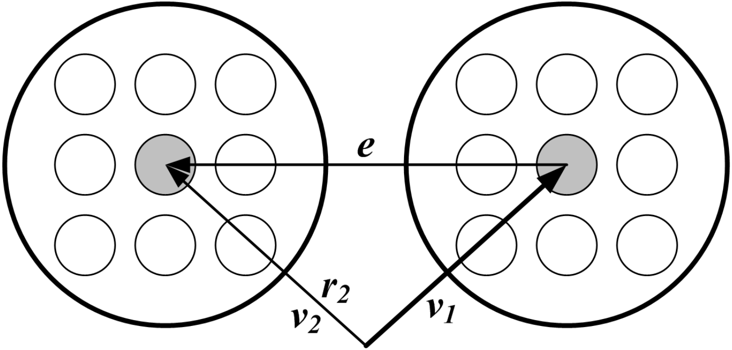
FEC — Ошибка обнаружена и верно исправлена



FEC — ошибка обнаружена, но исправлена неверно



FEC — необнаружимая ошибка



FEC: Код Хемминга

- Код Хемминга формирует номер ошибочного разряда.
- Признаком отсутствия ошибок является нулевой номер.
- Поэтому вводится «фиктивный» нулевой разряд.
- Если исходное слово имеет длину n бит, тогда к нему нужно добавить m дополнительных бит, исходя из неравенства

$$2^m \geq n + m + 1, \tag{4}$$

где левая часть неравенства — количество m-разрядных двоичных чисел, правая — общая длина кода с учетом «фиктивного» разряда. Выбирается минимальное m из возможных.

Кодирование и декодирование по Хеммингу

Алгоритм кодирования

1. В двоичном числе длиной $m + n$ бит (без фиктивного разряда) контрольные m бит размещаются в разрядах с номерами, равными степени двойки ($2^i, 0 \leq i < m$). А n бит исходного слова размещаются в оставшихся разрядах. Контрольный биты при этом инициализируются нулевыми значениями.
2. Каждый контрольный бит c_{2^i} в разряде 2^i пересчитывается как сумма по XOR бит кода, находящихся в разрядах с номерами, двоичное представление которых содержит единицу в i разряде (включая и сам контрольный разряд).

При декодировании контрольные разряды пересчитываются в соответствии с пунктом 2 алгоритма кодирования. В результате в контрольных разрядах будет получено двоичное представление номера разряда ошибочного бита.

Построить код Хемминга для слова $u = 0011$

$n = 4$. Исходя из формулы (4) выбирается $m = 3$.

$u=$

0	0	1	1

$v=$

$v=$	0	0	1	0	1	0	0	
	7	6	5	4	3	2	1	0
v_1	1	0	1	0	1	0	1	0
v_2	1	1	0	0	1	1	0	0
v_4	1	1	1	1	0	0	0	0

$$v_1 = v_1 \oplus v_3 \oplus v_5 \oplus v_7 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$
$$v_2 = v_2 \oplus v_3 \oplus v_6 \oplus v_7 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$
$$v_4 = v_4 \oplus v_5 \oplus v_6 \oplus v_7 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

1	1	0

$v=$

0	0	1	1	1	1	0	

Код Хемминга
Обнаружение и исправление одиночных ошибок

$v=$	0	0	1	1	1	1	0	
$r=$	0	0	0*	1	1	1	0	
	7	6	5	4	3	2	1	0
	1	0	1	0	1	0	1	0	r_1
	1	1	0	0	1	1	0	0	r_2
	1	1	1	1	0	0	0	0	r_4

$$r_1=r_1\oplus r_3\oplus r_5\oplus r_7=0\oplus 1\oplus 0\oplus 0=1$$
$$r_2=r_2\oplus r_3\oplus r_6\oplus r_7=1\oplus 1\oplus 0\oplus 0=0$$
$$r_4=r_4\oplus r_5\oplus r_6\oplus r_7=1\oplus 0\oplus 0\oplus 0=1$$

1	0	1

$u'=$	0	0	1	1

Код Хемминга

Задание

Получена последовательность бит r. Перед передачей из исходного 4-х битного слова был получен код Хемминга. Выяснить, были ли ошибки в процессе передачи. Если были, то выполнить коррекцию, предполагая, что ошибка одиночная. Выделить исходное 4-х битное слово.

r=	1	1	0	0	1	0	1	
	7	6	5	4	3	2	1	0
r_1	1	0	1	0	1	0	1	0
r_2	1	1	0	0	1	1	0	0
r_4	1	1	1	1	0	0	0	0

u'=

Код Хемминга

Задание

Получена последовательность бит r. Перед передачей из исходного 4-х битного слова был получен код Хемминга. Выяснить, были ли ошибки в процессе передачи. Если были, то выполнить коррекцию, предполагая, что ошибка одиночная. Выделить исходное 4-х битное слово.

	7	6	5	4	3	2	1	0
$r=$	1	1	0	0	1*	0	1	
r_1	1	0	1	0	1	0	1	0
r_2	1	1	0	0	1	1	0	0
r_4	1	1	1	1	0	0	0	0

$$r_1 = r_1 \oplus r_3 \oplus r_5 \oplus r_7 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$$
$$r_2 = r_2 \oplus r_3 \oplus r_6 \oplus r_7 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$$
$$r_4 = r_4 \oplus r_5 \oplus r_6 \oplus r_7 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$$

0	1	1

$u'=$	1	1	0	0

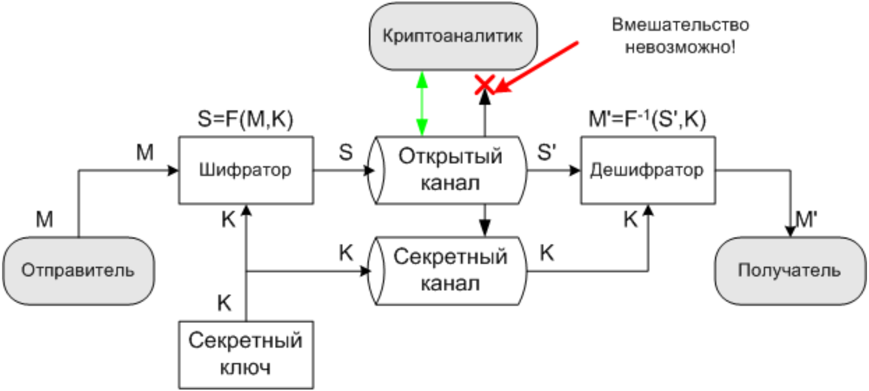
Базовая схема передачи информации

Можно выделить следующие виды каналов связи:

- Секретный гарантирует конфиденциальность, целостность и принадлежность M;
- Аутентичный гарантирует только целостность и принадлежность M;
- Открытый не гарантирует ничего в отношении M.

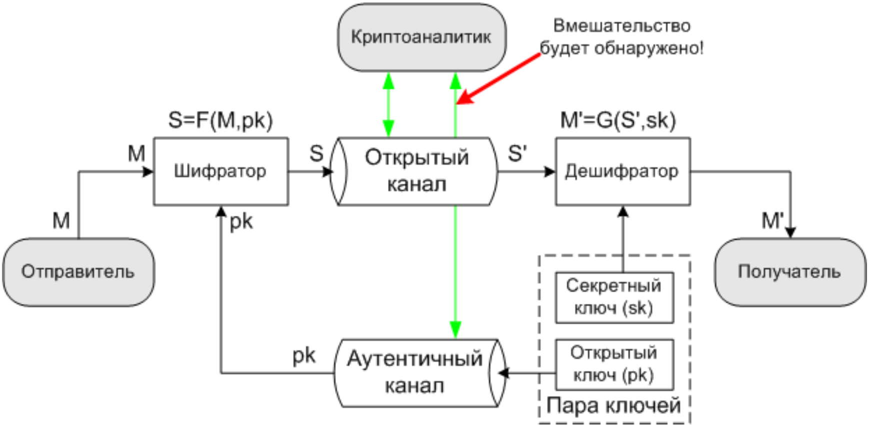
Схемы шифрования

Симметричная схема

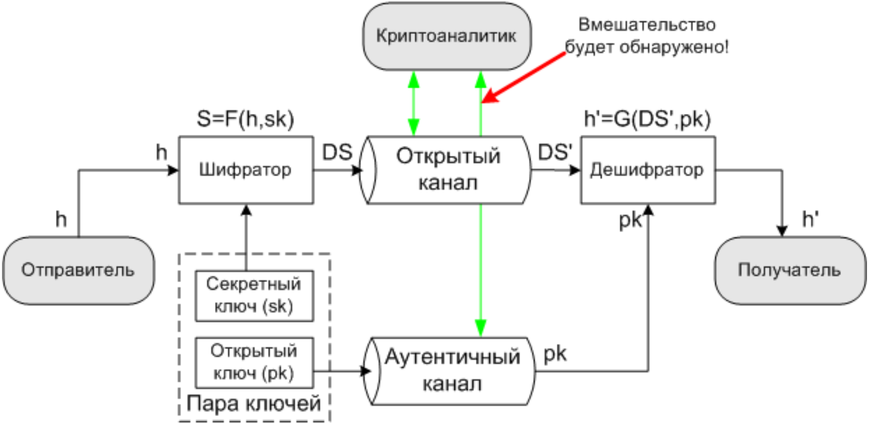


Схемы шифрования

Ассиметричная схема



Цифровая подпись



В заключение

Изложение математических основ кодирования можно найти, например, в [1, 2]. По основам теории информации можно рекомендовать книгу [3]. Основы кодирования подробно изложены в [4]. Заинтересовавшимся алгоритмами сжатия можно рекомендовать книгу [5].

Как введение по вопросам безопасности информации можно рекомендовать [6]. Обзор задач и протоколов информационной безопасности прекрасно описан в [7]. Математические основы шифрования для сильно интересующихся можно найти в [7, 8, 9].

Библиография I

Ф.А.Новиков. Дискретная математика для программистов / Ф.А.Новиков. —
СПб.: Питер, 2000. —
304 с.

С.В.Яблонский. Введение в дискретную математику: учебное пособие для вузов / С.В.Яблонский; Под ред. В.А.Садовничего. —
М.: Высшая школа, 2001. —
384 с.

Библиография II

- *В.В.Панин*. Основы теории информации: учебное пособие для вузов / В.В.Панин. —
  3 изд. —
  М.: БИНОМ, 2009. —
  438 с.
- *М.Вернер*. Основы кодирования: учебник для ВУЗов / М.Вернер. —
  М.: Техносфера, 2004. —
  288 с.
- *Д.Сэлмон*. Сжатие данных, изображений и звука / Д.Сэлмон. —
  М.: Техносфера, 2004. —
  368 с.

Библиография III

Э.Танненбаум. Современные операционные системы / Э.Танненбаум. —
3 изд. —
СПб.: Питер, 2010. —
1120 с.

Б.Шнайер. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б.Шнайер. —
М.: Триумф, 2002. —
816 с.

Н.Смарт. Криптография / Н.Смарт. —
М.: Техносфера, 2006. —
528 с.

Библиография IV

В.Мао. Современная криптография: теория и практика / В.Мао. —
М.: Издательский дом «Вильямс», 2005. —
786 с.