Министерство образования Республики Беларусь

Учреждение образования
«Полоцкий государственный университет»

ЭЛЕМЕНТЫ ТЕОРИИ ИНФОРМАЦИИ

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС
для студентов специальности 1-40 01 01
«Программное обеспечение информационных технологий»

Составление и общая редакция
Р.П. Богуша

Новополоцк 2006

УДК 621.391.1 (075.8)
ББК 32.811 я 73
Э 44

РЕЦЕНЗЕНТЫ:

А.П. КОМАРОВСКИЙ, инженер электросвязи I категории
Витебского филиала РУП «Белтелеком»;
Д.О. ГЛУХОВ, канд. техн. наук, доцент, проректор по информатизации

Рекомендован к изданию методической комиссией радиотехнического факультета

Э 44 **Элементы теории информации:** Учеб.-метод. комплекс для студ. спец. 1-40 01 01 «Программное обеспечение информационных технологий» / Сост. и общ. ред. Р.П. Богуша. – Новополоцк: УО «ПГУ», 2006. – 160 с.
ISBN 985-418-408-0

Представлен курс лекций по элементам теории информации: рассмотрены вопросы количественной оценки информации, характеристики дискретных каналов связи, элементы теории сложности, методы и алгоритмы криптографического кодирования информации. Представлены методические указания к выполнению практических работ. Предлагается система оценки знаний студентов.

Предназначен для студентов и преподавателей радиотехнического факультета.

УДК 621.391.1 (075.8)
ББК 32.811 я 73

ISBN 985-418-408-0

СОДЕРЖАНИЕ

ВВЕДЕНИЕ В КУРС «ЭЛЕМЕНТЫ ТЕОРИИ ИНФОРМАЦИИ»

1. Цель и задачи дисциплины

Основная цель курса «Элементы теории информации»: изучение студентами элементов фундаментальной и прикладной теории информации, включая аспекты теории сложности проблем и алгоритмов и методы криптографического кодирования информации.

В результате изучения дисциплины студенты должны:

знать:

– свойста и этапы обращения информации;

– методы количественной оценки информации при различных состояниях источника сообщений и свойства энтропии;

– криптографические алгоритмы и системы.

уметь:

– определять энтропию зависимых и статистически независимых ансамблей;

– рассчитывать основные характеристики источников сообщений;

– использовать криптографические алгоритмы для защиты информации от несанкционированного доступа.

иметь представление:

– о сложности проблем и алгоритмов;

– о направлениях, перспективах и проблемах развития теории информации.

2. Структура дисциплины

Согласно учебному плану специальности 1-40 01 01 «Программное обеспечение информационных технологий» курс «Элементы теории информации» изучается студентами на 1 курсе (1 семестр), рассчитан на 36 часов лекций и 18 часов практических работ. Ниже представлено распределение курса по видам аудиторных занятий по разделам и темам.

ЛЕКЦИОННЫЙ КУРС

Наименование	Количество часов
Модуль 1. Основные понятия и определения	
1.1. Информация, ее свойства и этапы обращения. Модель информационной системы	2
Модуль 2. Количественная оценка информации. Информационные характеристики каналов связи	
2.1. Количество информации. Энтропия	2
2.2. Количество информации от опыта в общем случае. Энтропия эргодического источника	2
2.3. Избыточность источника сообщений. Основные модели каналов связи	2
2.4. Дискретный канал и его основные характеристики. Согласование характеристик сигнала и канала	2
Модуль 3. Элементы теории сложности. Элементы теории чисел	
3.1. Сложность алгоритмов и проблем	2
3.2. Модульная арифметика	2
Модуль 4. Основы криптографической защиты информации	
4.1. Основные понятия и определения	2
4.2. Шифры перестановки. Шифры простой замены	2
4.3. Шифрование методом гаммирования. Шифры сложной замены	2
Модуль 5. Современные симметричные криптосистемы	
5.1. Американский стандарт шифрования DES	2
5.2. Реализация функции шифрования в алгоритме DES. Алгоритм вычисления ключей	2
5.3. Основные режимы работы алгоритма DES. Комбинирование блочных алгоритмов	2
5.4. Логика построения шифра, структура ключевой информации и основной шаг криптопреобразования стандарта ГОСТ 28147-89	2
5.5. Базовые циклы, основные режимы шифрования алгоритма ГОСТ 28147-89	2
Модуль 6. Асимметричные криптосистемы	
6.1. Построения систем с открытым ключом. Алгоритмы рюкзака	2
6.2. Алгоритм RSA. Шифрование и дешифрования RSA	2
6.3. Криптосистема Эль-Гамаля. Алгоритм Рабина. Комбинированный метод шифрования	2

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

Наименование	Количество часов
1. Информация и этапы ее обращения. Модель информационной системы	2
2. Количественная оценка информации	2
3. Источники дискретных сообщений	2
4. Сложность алгоритмов и проблем. Модульная арифметика	2
5. Криптографическое кодирование информации с использованием шифрующих таблиц и системы шифрования Цезаря	2

6. Криптографическое кодирование информации с использованием аффинной системы подстановок Цезаря и алгоритма Вижинера	2
7. Криптографическое кодирование информации одноразовым шифром и гаммированием	2
8. Симметричные системы шифрования DES и ГОСТ 28147-89	2
9. Ассиметричные системы шифрования	2

3. Оценка знаний студентов

Для оценки работы и знаний студентов в рамках курса «Элементы теории информации» используется накопительная система. Результирующая оценка выставляется по сумме баллов, которые студент набирает в течение всего учебного семестра, а также в результате выходного итогового контроля – зачета.

Для получения аттестации студент должен успешно написать контрольную работу – два теоретических вопроса (максимальное количество баллов за коллоквиум – 50, для аттестации – 30 и более).

Распределение баллов по видам занятий

Вид занятий	Форма оценки учебной активности студента	Максимальное количество баллов по каждой форме оценки	Максимальное количество баллов по каждому виду занятий
Практические работы	Устные ответы на вопросы	5	20×9 = 180
	Решение задач и (или) выполнение тестовых заданий	15	
Зачет	Ответы на вопросы	120	120

Примечание. Аттестации – 100 баллов максимум.

Дополнительные баллы предусматриваются за подготовку докладов (до 80 баллов).

Итоговый зачет по данной дисциплине выставляется в случае, если студент набрал 210 баллов.

Для этого необходимо, например:

- получить две аттестации – минимум 60 баллов;
- по результатам практических работ набрать не менее 70 баллов;
- получить 80 баллов при сдаче зачета.

МОДУЛЬ 1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

1. Информация, ее свойства и этапы обращения. Модель информационной системы.

2. Вопросы и задания для самопроверки.

3. Практическое занятие № 1.

Цель модуля – изучение студентами основных понятий теории информации, включая модель системы передачи информации и уровни проблем информации.

В результате изучения модуля студенты должны:

– знать основные свойства информации;

– знать основные этапы обращения информации;

– знать модель системы передачи информации;

– уметь охарактеризовать уровни проблем информации и четко представлять, в чем их отличие;

– иметь представление о типах сигналов;

– иметь представление о принципах получения сообщений, их преобразования, передачи и приема.

1.1. Информация, ее свойства и этапы обращения. Модель информационной системы

1.1.1. Информация и ее свойства

Началом развития теории информации как науки считается опубликованная в 1948 г. работа К. Шеннона «Математическая теория связи». Эффективная организация обмена и обработки информации приобретает все большее значение как условие успешной практической деятельности людей. К информационной технике относятся средства, служащие для восприятия, подготовки, передачи, переработки, хранения и представления какой-либо информации, получаемой от человека, природы машины, вообще от какого-либо объекта наблюдения и управления. Комплексное применение этих средств приводит к созданию больших и сложных информационных систем.

Само понятие «информации» во многом остается интуитивным и получает различные смысловые наполнения в различных отраслях человеческой деятельности. Рассмотрим простую схему из трех понятий: «Объект», «Среда» и «Взаимодействие». «Объект» – это нечто устойчивое во време-

ни и ограниченное в пространстве интересующее нас как единое целое. «Среда» – это множество всех других потенциальных «Объектов», интересующих нас только с точки зрения их влияния на состояние выделенного «Объекта» и обратного влияния «Объекта» на их состояния. «Взаимодействие» – это растянутый во времени процесс взаимозависимого изменения параметров состояния «Объекта» и «Среды». Эта схема является замкнутой в том смысле, что «Среда» включает в себя все потенциальные «Объекты», способные влиять на состояние выделенного «Объекта». Любое взаимодействие между объектами, в процессе которого один приобретает некоторую субстанцию, а другой ее не теряет, называется информационным взаимодействием. При этом передаваемая субстанция называется информацией. Из этого определения следует два наиболее общих свойства информации: информация не может существовать вне взаимодействия объектов и информация не теряется ни одним из них в процессе этого взаимодействия.

Информация наряду с материей и энергией является первичным понятием нашего мира и поэтому в строгом смысле не может быть определена. Основные свойства информации:

– информация приносит сведения об окружающем мире, которых в рассматриваемой точке не было до ее получения;

– информация не материальна, но она проявляется в форме материальных носителей дискретных знаков или первичных сигналах;

– знаки и первичные сигналы несут информацию только для получателя способного распознать.

В узком практическом смысле под информацией обычно понимают совокупность сведений об окружающем мире, являющихся объектом хранения, передачи и преобразования.

Понятие информации связано с некоторыми моделями реальных вещей, отражающими их сущность в той степени, в какой это необходимо для практических целей. Таким образом, под информацией нужно понимать не сами предметы и процессы, а их представительные характеристики отражения или отображения в виде чисел, формул, описаний, чертежей, символов, образов и других абстрактных характеристик.

Сама по себе информация может быть отнесена к области абстрактных категорий, подобных, например, математическим формулам. Однако проявляется она всегда в материально-энергетической форме в виде сигналов.

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

Теория информации разделилась на фундаментальную и прикладную. Фундаментальная теория информации включает: анализ сигналов как средства передачи сообщений и оценка переносимого «количества информации»; анализ информационных характеристик источников сообщений и каналов связи и обоснование принципиальной возможности кодирования и декодирования сообщений, обеспечивающих предельно допустимую скорость передачи сообщений по каналу связи как при отсутствии, так и при наличии помех.

Прикладная теория информации основывается на практических результатах, полученных при рассмотрении фундаментальных законов. Можно определить прикладную теорию информации двояко. Первое определение (узкое) – разработка конкретных методов и средств различного кодирования сообщений. Второе, более широкое: предметом теории информации является изучение любых процессов, связанных с получением, передачей, хранением, обработкой и использованием информации. Второе определение затрагивает проблемы буквально всех наук (от математики до педагогики).

1.1.2. Этапы обращения информации

Роль информации может ограничиваться неопределенным эмоциональным воздействием на человека, но в чисто технических (автоматических) и человеко-машинных (автоматизированных) системах она чаще всего используется для выработки управляющих воздействий.

При обращении информации в системах можно выделить отдельные этапы.

Этап восприятия информации: осуществляется извлечение и анализ информации об объекте (процессе) и формирование образа объекта, проводится его опознание и оценка. Полезный сигнал отделяется от шума, т.е. мешающей информации. Происходит выявление или измерение полезного сигнала.

Этап подготовки информации: проводятся операции нормализации, аналого-цифрового преобразования, шифрование. В результате восприятия и подготовки получается сигнал в форме, удобной для передачи или обработки.

Этап передачи и хранения: информация пересылается либо из одного места в другое, либо от одного момента времени до другого. Поскольку

задачи, возникающие на этих этапах, близки друг другу, хранение информации часто в самостоятельный этап не выделяется. Для передачи на расстояние используются каналы различной физической природы. Для хранения используются магнитные и другие носители. Извлечение сигнала на выходе канала, подверженного действию шумов, носит характер вторичного восприятия.

Этап обработки информации: выявляются ее взаимозависимости, представляющие интерес для системы. Формализуемый процесс обработки может выполняться техническими средствами без участия человека. В системах управления целью обработки является решение задачи выбора управляющих воздействий (этап принятия решения).

Этап отображения информации: должен предшествовать этапам, связанным с участием человека. Цель – предоставить человеку информацию в форме, доступной для его органов чувств.

Этап воздействия: информация используется для осуществления необходимых изменений в системе.

1.1.3. Модель информационной системы

Совокупность средств информационной техники и людей, объединенных для достижения определенных целей, называют информационной системой. Системы бывают автоматическими или автоматизированными (человеко-машинными). Автоматизированная информационная система (АИС) становится автоматизированной системой управления (АСУ), если входящая в систему информация извлекается из какого-либо объекта, а выходящая используется для изменения состояния того же объекта.

Большинство АИС и АСУ являются локальными, т.е. системами ограниченных размеров от устройства (например, видеокамера) до размеров предприятия. Однако сегодня такие системы интегрируются и взаимодействуют на региональном и глобальном уровнях.

Системы становятся территориально рассредоточенными, иерархичными по функциям. Обеспечение взаимодействия рассредоточенных систем требует протяженных высокоскоростных и надежных каналов связи, а большой объем информации – ЭВМ высокой производительности. Развитие таких систем уменьшает роль телефона, телеграфа, почты и т.д.

Информация поступает в АИС в форме сообщений. Под сообщением понимают совокупность знаков или первичных сигналов, содержащих информацию. Иначе говоря, сообщение – это информация, представленная в какой-либо форме. Пример сообщений: текст телеграммы, данные на вы-

ходе ЭВМ, речь, музыка и т.д. Для того чтобы сообщение можно было передать получателю, необходимо воспользоваться некоторым физическим процессом, способным с той или иной скоростью распространяться от источника к получателю сообщения. Изменяющийся во времени физический процесс, отражающий передаваемое сообщение, называется сигналом. Сообщения могут быть функциями времени (когда информация представлена в виде первичных сигналов: речь, музыка) и не является ими (когда информация представлена в виде совокупности знаков). Сигнал всегда является функцией времени.

Сигналы подразделяют на дискретные, непрерывные и дискретно-непрерывные. Сигнал считают дискретным по данному параметру, если число значений, которое может принимать этот параметр, конечно (или счетно). Если множество возможных значений параметра образует континуум, то сигнал считают непрерывным. Сигнал, дискретный по одному параметру и непрерывный по другому, называют дискретно-непрерывным.

Как математическая модель используются:

– непрерывная функция непрерывного аргумента (например, времени) (рис. 1.1);

– непрерывная функция дискретного аргумента, например, функция, значения которой отсчитывают только в определенные моменты времени (рис. 1.2). Временной интервал Δt между соседними отсчетами называется шагом дискретизации;

– дискретная функция непрерывного аргумента, например функция времени, квантованная по уровню (рис. 1.3);

– дискретная функция дискретного аргумента, принимающая одно из конечного множества возможных значений в дискретные моменты времени (рис. 1.4).

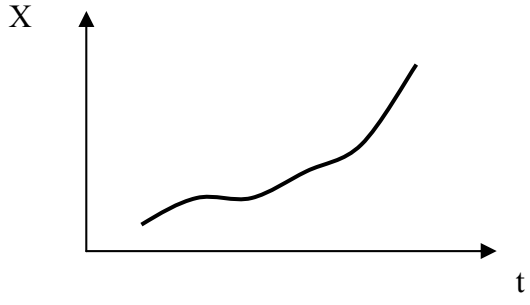


Рис. 1.1. Модель непрерывного сигнала

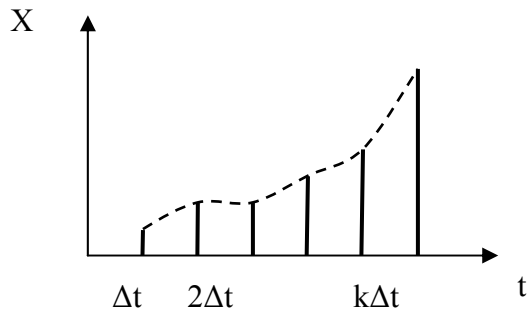


Рис. 1.2. Модель дискретного по времени сигнала

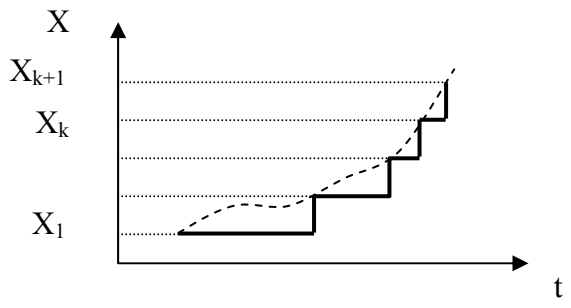


Рис. 1.3. Модель квантованного сигнала

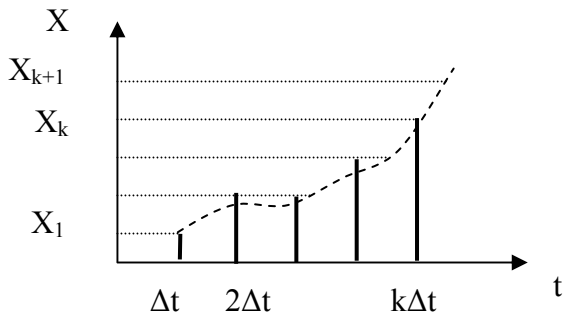


Рис. 1.4. Модель дискретного сигнала

Источник информации или сообщения – это физический объект, система или явление, формирующие передаваемое сообщение в виде двоичных символов (рис. 1.5).

Кодер первичного кода преобразует двоичный код с выхода источника в первичный код, который более удобен для дальнейших преобразований последующими устройствами.

Кодер источника обеспечивает сокращение объема (сжатие) информации с целью повышения скорости ее передачи или сокращения полосы частот, требуемых для передачи.

Криптографический кодер выполняет криптографическое кодирование (шифрование) для обеспечения секретности передачи информации.

Кодер канала выполняет помехоустойчивое кодирование, которое представляет собой способ обработки передаваемых данных, обеспечивающий уменьшение количества ошибок, возникающих в процессе передачи по каналу с помехами.

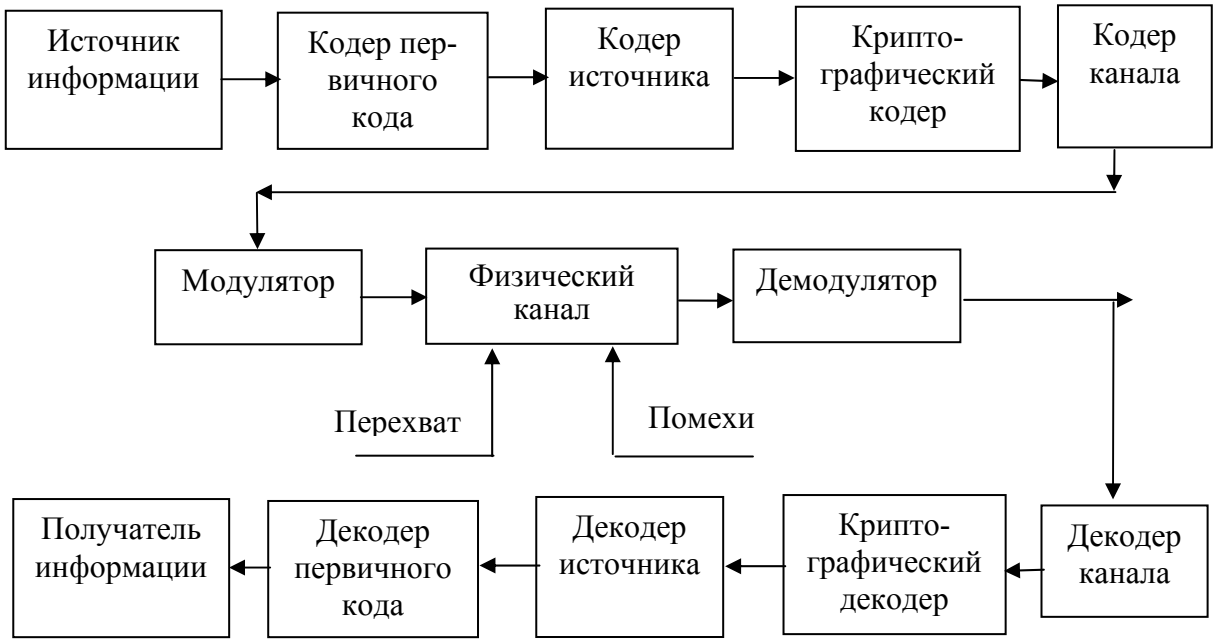


Рис. 1.5. Классическая модель канала передачи, хранения, обработки и распределения информации

Модулятор порождает множество непрерывных сигналов конечной длительности и реализует отображение выходных последовательностей кодера в это множество сигналов.

Физический канал – это вся аппаратура и вся физическая среда, через которую проходит сигнал на пути от выхода модулятора до входа демодулятора. Физический канал не обязательно представляет собой систему связи, работающую в режиме реального времени; он может быть системой хранения, обработки или распределения информации. Обычно выходной сигнал канала является суммой входного сигнала, умноженного на коэффициент передачи, и случайного шума. Таким образом, сигналы на выходе физического канала могут отличаться от переданных вследствие затухания, искажения и воздействия помех. Помехами называют сторонние возмущения, искажающие полезный сигнал. Эффект воздействия помех на различные блоки системы стараются учесть эквивалентным изменением характеристик линии связи. Поэтому источник помех условно относят к линии связи. Меру соответствия принятого сообщения посланному называют достоверностью передачи.

Демодулятор – это устройство, которое на основе наблюдения принятого сигнала оценивает, какой из возможных символов был передан. Кроме этого, демодулятор часто выполняет еще одну функцию, состоящую в передаче декодеру информации о степени надежности оценки каждого символа.

Декодеры выполняют функции, обратные функциям соответствующих кодеров.

1.1.4. Уровни проблем передачи информации

Обмен информацией предполагает использование некоторой системы знаков, например, естественного или искусственного (формального) языка. Информация о непрерывных процессах также может быть выражена посредством знаков. Изучение знаковых систем наукой о знаках и языках (семиотикой) проводится, по крайней мере, на трех уровнях.

На синтаксическом уровне рассматривают внутренние свойства текстов, т.е. отношения между знаками, отражающие структуру данной знаковой системы. Внешние свойства текстов изучают на семантическом и прагматическом уровнях.

На семантическом уровне анализируют отношения между знаками и обозначаемыми ими предметами, действиями, качествами, т.е. смысловое содержание текста, его отношение к источнику информации.

На прагматическом уровне рассматривают отношения между текстом и теми, кто его использует, т.е. потребительское содержание текста.

Решение проблем синтаксического уровня помогает создать теоретические основы построения систем связи с показателями работы, близкими к предельно возможным. Это чисто технические проблемы совершенствования методов передачи сообщений и их материального воплощения – сигналов, проблемы доставки получателю сообщений как совокупности знаков, при этом полностью абстрагируется их смысловое и прагматическое содержание.

Прикладная теория информации решает проблемы именно этого, синтаксического, уровня. Она опирается на понятие «количество информации», являющееся мерой частоты употребления знаков, которая никак не отражает ни смысла, ни важности передаваемых сообщений.

На семантическом уровне формализуют смысл передаваемой информации, судят о близости информации к истине, оценивают ее качество. Эти проблемы чрезвычайно сложны, т.к. смысловое содержание информации больше зависит от получателя, чем от семантики сообщения. Одно и то же сообщение может быть понято по-разному, поскольку зависит от личности и уровня знаний лица, эту информацию получившего. Мы еще не умеем измерять семантическую информацию.

На прагматическом уровне интересуют последствия от получения информации абонентом. Потребительская ценность полученной информации различна для разных получателей. Имеет значение скорость доставки, наличие реального масштаба времени для обмена информацией и др. Ряд проблем этого уровня решается. Предложены меры оценки потребительской информации, ведутся исследования в области старения информации, т.е. потери ее ценности в процессе доставки, и т.д.

1.2. Вопросы и задания для самопроверки

1. Что понимается под термином «информация»?
2. Назовите основные свойства информации.
3. Какие формы существования информации можете назвать?
4. Решением каких задач занимается фундаментальная (прикладная) теория информации?
5. Назовите этапы обращения информации.
6. Приведите структурную схему системы передачи информации и поясните ее.
7. Что понимается под сообщением и сигналом?
8. В чем отличие дискретных и непрерывных сообщений?
9. Что понимается под достоверностью передачи?
10. Назовите уровни проблем передачи информации и дайте характеристику каждому уровню?

1.3. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1
Информация и этапы ее обращения.
Модель информационной системы

Теория для практического занятия представлена в модуле 1.

Перед выполнением тестовых заданий проводится опрос с использованием вопросов, представленных в разделе 1.2.

1. Под информацией следует понимать:

a) различные предметы;

b) происходящие процессы;

c) представительные характеристики различных предметов и процессов отражения или отображения в виде чисел и формул;

d) представительные характеристики различных предметов и процессов отражения или отображения в виде описаний, чертежей, символов, образов.

2. Фундаментальная теория информации включает:

a) анализ сигналов как средства передачи сообщений и оценку переносимого «количества информации»;

b) анализ информационных характеристик источников сообщений и каналов связи и обоснование принципиальной возможности кодирования и декодирования сообщений, обеспечивающих предельно допустимую скорость передачи сообщений по каналу связи, как при отсутствии, так и при наличии помех;

c) разработку конкретных методов и средств различного кодирования сообщений;

d) изучение любых процессов, связанных с получением, передачей, хранением, обработкой и использованием информации.

3. К этапам обращения информации относятся:

a) этап воздействия;

b) этап отображения информации;

c) этап обработки информации;

d) этап передачи и хранения.

4. Информационной системой называется:

a) совокупность средств информационной техники, объединенных для достижения определенных целей;

b) совокупность средств различного вида техники, объединенных для достижения определенных целей;

c) совокупность средств информационной техники и людей, объединенных для достижения определенных целей;

d) средства электронной техники, объединенные для достижения определенных целей.

5. В какой форме поступает информация в АИС:

a) в форме непрерывных сигналов;

b) в форме дискретных сигналов;

c) в форме сообщений;

d) в форме любых видов сигналов.

6. Кодер источника в АИС предназначен:

a) для внесения избыточности в сообщение;

b) для криптографического кодирования;

c) для сжатия информации;

d) для преобразования в код, более удобный для дальнейших преобразований.

7. Код канала в АИС предназначен:

a) для внесения избыточности в сообщение;

b) для криптографического кодирования;

c) для сжатия информации;

d) для преобразования в код, более удобный для дальнейших преобразований.

8. Первичный кодер в АИС предназначен:

a) для внесения избыточности в сообщение;

b) для криптографического кодирования;

c) для сжатия информации;

d) для преобразования в код, более удобный для дальнейших преобразований.

9. На семантическом уровне передачи информации:

a) рассматривают внутренние свойства текстов;

b) анализируют смысловое содержание текста, его отношение к источнику информации;

c) рассматривают отношения между текстом и теми, кто его использует;

d) анализируют отношения между знаками и обозначаемыми ими предметами, действиями, качествами.

10. Проблемы какого уровня решает прикладная теория информации?

a) семантического;

b) синтаксического;

c) прагматического.

МОДУЛЬ 2. КОЛИЧЕСТВЕННАЯ ОЦЕНКА ИНФОРМАЦИИ. ИНФОРМАЦИОННЫЕ ХАРАКТЕРИСТИКИ КАНАЛОВ СВЯЗИ

1. Количество информации. Энтропия.
2. Количество информации от опыта в общем случае. Энтропия эргодического источника.
3. Избыточность источника сообщений. Основные модели каналов связи.
4. Дискретный канал и его основные характеристики. Согласование характеристик сигнала и канала.
5. Вопросы и задания для самопроверки.
6. Практическое занятие № 2.
7. Практическое занятие № 3.

Цель модуля – изучение студентами вопросов количественной оценки информации при различных состояниях источника сообщений, энтропии зависимых и статистически независимых ансамблей и ее свойств, характеристик сигнала и дискретного канала.

В результате изучения модуля студенты должны:

– знать методы количественной оценки информации при различных состояниях источника сообщений;

– знать свойства количества информации и энтропии;

– знать основные характеристики дискретного канала и сигнала;

– уметь определять точное и среднее количество информации, избыточность источника сообщений;

– иметь представление об отличии эргодических источников от источников, в которых отсутствуют коррелятивные связи.

2.1. Количество информации. Энтропия

2.1.1. Определение количества информации

Очевидно, что возникают трудности в оценке количества информации, которое содержится в сообщениях, т.к. сообщения разнятся как по своей природе, так и по содержанию и по назначению. Количество информации должно определяться через нечто общее, объективно присущее всему многообразию различной информации, оставаясь при этом созвучным нашим интуитивным представлениям, связанным с фактом получения информации. Этим общим, характеризующим фактом получения произвольной информации является, во-первых, наличие опыта. Всякая информация до-

бывается нами в результате опыта и только опыта. Во-вторых, до опыта должна существовать некоторая неопределенность в том или ином исходе опыта.

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

Таким образом, для установления формулы для вычисления количества информации необходимо уметь вычислять неопределенность некоторой ситуации до и после опыта. Разность между этими количествами неопределенности и дает нам искомое количество информации, полученное от такого опыта.

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

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

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

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

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

$$I(n = 1) = 0.$$

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

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

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

$$I = C \log_a N \ .$$

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

$$I = \log N \ .$$

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

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

$$C = \log\ m^n = n \log m\,. \qquad (2.1)$$

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

Следует отметить, что мера количества информации в виде (2.1) относится к весьма частному случаю, когда после опыта неопределенности в исходе нет и все исходы равновероятны.

Дальнейшее развитие теории информации шло в направлении учета статистических характеристик.

Если от источника информации по каналу связи передавать сообщение о событии, априорная (исходная, доопытная) вероятность которого на передающей стороне равна P_1, то после приема сообщения апостериорная (конечная, послеопытная) вероятность этого события для приемника информации равна P_2, и количество информации, полученное в результате приема сообщения, определяется как:

$$I = \log(P_2 / P_1) = \log P_2 - \log P_1. \qquad (2.2)$$

Для канала связи без помех и искажений прием сообщения становится достоверным событием, т.е. вероятность $P_2 = 0$, тогда из (2.2) следует, что:

$$I = -\log P_1. \qquad (2.3)$$

Из (2.3) следует, что чем меньше вероятность P_1, тем больше неопределенность исхода, т.е. тем большее количество информации содержится в принятом сообщении.

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

2.1.2. Энтропия ансамбля

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

$$X = \begin{pmatrix} X_1 & X_2 & \cdots & X_j & \cdots & X_k \\ P_1 & P_2 & \cdots & P_j & \cdots & P_k \end{pmatrix},\ \sum_{i=1}^{k} P_i = 1.$$

Пусть имеет место N возможных исходов опыта, из них k разных, и i-тый исход ($i = 1,2,...,k$) повторяется n_i раз и вносит информацию, коли-

чество которой оценивается как I_i Тогда средняя информация, доставляемая одним опытом, равна

$$I_{cp} = \frac{n_1 I_1 + n_2 I_2 + ... + n_k I_k}{N}.$$

Но количество информации в каждом исходе согласно (2.3) будет

$$I_i = -\log P_i.$$

Тогда

$$I_{cp} = \frac{n_1\left(-\log P_1\right) + n_2\left(-\log P_2\right) + ... + n_k\left(-\log P_k\right)}{N}. \quad (2.4)$$

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

$$\frac{n_i}{N} = P_i. \quad (2.5)$$

Подставляя (2.5) в (2.4), получим

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

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

$$H = I_{cp} = -\sum_{i=1}^{k} P_i \log P_i. \quad (2.6)$$

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

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

Если все k различных состояний источника равновероятны, то:

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

и энтропия максимальна, согласно (2.6) имеем

$$H_{\max} = -\sum_{i=1}^{k} \frac{1}{k} \log \frac{1}{k} = \log k. \quad (2.7)$$

В частном случае при равновероятных сообщениях формулы (2.6) и (2.7) совпадают. Совпадение оценок количества информации по Шеннону и Хартли свидетельствуют о полном использовании информационной емкости системы. В случае неравных вероятностей количество информации по Шеннону меньше информационной емкости системы.

2.1.3. Энтропия объединения

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

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

$$P(X,Y) = \begin{bmatrix} P(x_1,y_1) & \dots & P(x_i,y_1) & \dots & P(x_n,y_1) \\ \dots & \dots & \dots & \dots & \dots \\ P(x_1,y_j) & \dots & P(x_i,y_j) & \dots & P(x_n,y_j) \\ \dots & \dots & \dots & \dots & \dots \\ P(x_1,y_m) & \dots & P(x_i,y_m) & \dots & P(x_n,y_m) \end{bmatrix}. \qquad (2.8)$$

Суммируя столбцы и строки матрицы (2.8), получим информацию об ансамблях X и Y исходных источников:

$$X = \begin{bmatrix} x_1 & \cdots x_i & \cdots x_n \\ P(x_1) \cdots & P(x_i) \cdots & P(x_n) \end{bmatrix}. \; Y = \begin{bmatrix} y_1 & \cdots y_j & \cdots y_m \\ P(y_1) \cdots & P(y_j) \cdots & P(y_m) \end{bmatrix}$$

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

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

$$P(x_i, y_i) = P(x_i)\, P(y_j / x_i) = P(y_j)\, P(x_i / y_j),$$

где $P(x_i, y_i)$ – вероятность реализации состояний x_i ансамбля X при условии, что реализовалось состояние y_j ансамбля Y; $P(y_j / x_i)$ – вероятность реализации состояний y_j ансамбля Y при условии, что реализовалось состояние x_i ансамбля X. Тогда выражение для энтропии объединения в соответствии с (2.6) принимает вид:

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

$$= -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i) P(y_j, x_i)\log P(x_i) P(y_j, x_i) = \qquad (2.9)$$

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

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

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

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

$$H(Y / X) = \sum_{i=1}^{n} P(x_i)\, H(Y / x_i) = -\sum_{i=1}^{n} P(x_i)\sum_{j=1}^{m} P(y_j / x_i)\log P(y_j / x_i). \qquad (2.10)$$

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

Подставляя (2.10) в (2.9), получаем

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

Выражая $P(x_i, y_i)$ через другую условную вероятность, найдем

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

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

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

В случае статистической независимости ансамблей X и Y имеют:

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

Тогда

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

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

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

получим

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

2.1.4. Свойства энтропии

1. Энтропия всегда неотрицательна, так как значения вероятностей выражаются дробными величинами, а их логарифмы – отрицательными величинами (2.6).

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

3. Энтропия имеет наибольшее значение при условии, когда все вероятности равны между собой (2.7).

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

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

График зависимости $H(X)$ в функции P

$$H(X) = -\left[P\log P + (1-P)\log(1-P)\right]$$

приведен на рис. 2.1.

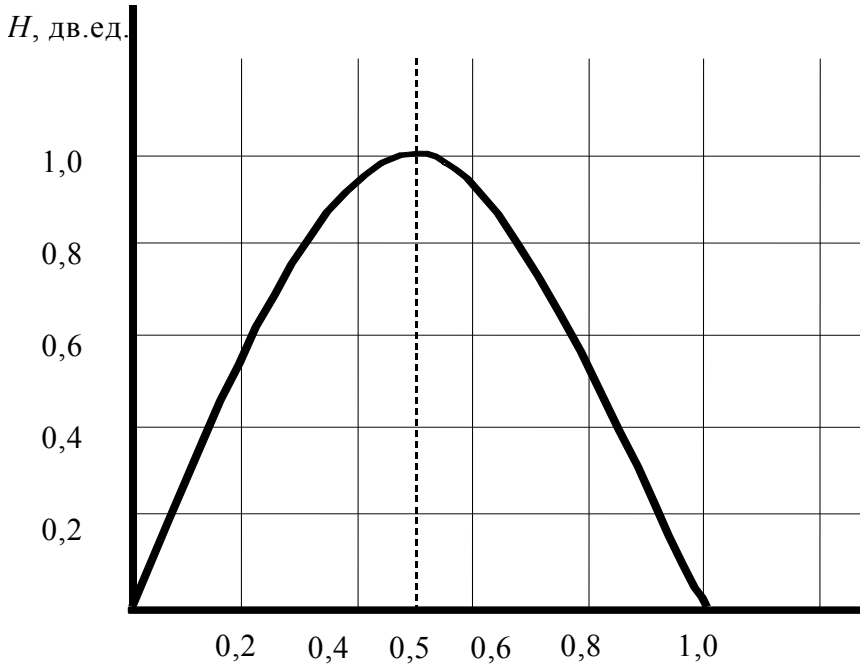


Рис. 2.1. Зависимость $H(X)$ в функции P

Энтропия непрерывно зависит от вероятности отдельных состояний, что непосредственно вытекает из непрерывности функции – $P\log P$.

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

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

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

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

7. Энтропия объединения любого числа зависимых ансамблей определяется из выражения

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

8. Энтропия не зависит от значений, принимаемых случайными величинами, а зависит только от вероятностей их появления (2.6).

9. Если события x_i и y_j статистически независимы, то:

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

Таким образом, сведения о результатах выбора состояний из одного ансамбля не снижает неопределенности выбора состояний из другого ансамбля. Если имеет место однозначная связь в реализациях состояний

x_i $(1 \le i \le n)$ ансамбля X и состояний y_j $(1 \le j \le m)$ ансамбля Y, то условная энтропия любого из ансамблей равна нулю:

$$H(X/Y) = 0 \text{ и } H(Y/X) = 0. \qquad (2.11)$$

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

Равенства (2.11) отражают факт отсутствия дополнительной неопределенности при выборе событий из второго ансамбля.

2.2. Количество информации от опыта в общем случае. Энтропия эргодического источника

2.2.1. Определение количества информации для общего случая

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

$$H(X) = -\sum_{i=1}^{m} P(x_i) \log P(x_i). \qquad (2.12)$$

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

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

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

Рассмотрим случай, когда между элементами сообщения и помехой статистические связи отсутствуют, искажения отдельных элементов сообщения являются событиями независимыми и адресату известна совокупность условных вероятностей $P(x_i / y_j)$ $(1 \le i \le m)$ $(1 \le j \le m)$ того, что вместо элемента сообщения x_i будет принят элемент сообщения y_j.

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

$$H(X / y_i) = -\log P(x_i / y_i).$$

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

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

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

$$H(X / Y) = \sum_{j=1}^{m} H(X / y_j) P(y_j) = -\sum_{i=1}^{m} \sum_{j=1}^{m} P(x_i, y_j) \log P(x_i, y_j). \qquad (2.13)$$

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

Таким образом, неопределенность передачи некоторого сигнала X до опыта $H(X)$, а после опыта $H(X/Y)$. Поэтому количество информации, имеющееся в Y и X:

$$I(Y,X)=H(X)-H(X/Y). \tag{2.14}$$

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

Подставим в выражение (2.14) необходимые значения $H(X)$ и $H(X/Y)$ из (2.12) и (2.13) соответственно получим:

$$\begin{aligned} I(Y,X) &= -\sum_{i=1}^{m} P(x_i)\log P(x_i) + \sum_{i=1}^{m}\sum_{j=1}^{m} P(x_i,y_j)\log P(x_i,y_j) = \\ &= -\sum_{i=1}^{m}\sum_{j=1}^{m} P(x_i,y_j)\log P(x_i) + \sum_{i=1}^{m}\sum_{j=1}^{m} P(x_i,y_j)\log P(x_i/y_j) = \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} P(x_i,y_j)\log\frac{P(x_i,y_j)}{P(x_i)P(y_j)}. \end{aligned} \tag{2.15}$$

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

2.2.2. Основные свойства количества информации

1. $I(X,Y)=I(Y,X)$, т.е. количество информации, содержащееся в случайном объекте Y о случайном объекте X, равно количеству информации, содержащемуся в случайном объекте X о случайном объекте Y. Данное свойство сразу же следует из (2.15), если учесть, что $P(x_i,y_i)=P(y_i,x_i)$.

2. $I(X,Y)\geq 0$, причем знак равенства имеет место, когда объекты X и Y независимы. Положительность $I(X,Y)$ следует из свойства энтропии: если события x_i *и* y_i статистически зависимы, то всегда $H(X/Y)<H(X)$ и $H(Y/X)<H(Y)$.

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

2.2.3. Энтропия эргодического источника

Эргодическим источником r-го порядка называется такой источник, у которого вероятность появления некоторого символа x_j зависит от r предыдущих. Для такого источника может быть найдено конечное число характерных состояний $S_1, S_2, ..,$ таких, что условная вероятность появления очередного символа зависит лишь от того, в каком из этих состояний находится источник. Вырабатывая очередной символ, источник переходит из одного состояния в другое либо возвращается в исходное состояние.

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

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

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

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

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

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

$$H(X)=\sum_k P(S_k)H(S_k)=-\sum_k\sum_{b/k} P(S_k)P(S_b/S_k)\log P(S_b/S_k)= \tag{2.16}$$

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

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

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

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

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

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

Из (2.16) тогда получим:

$$H(X)=-\sum_{k=1}^{n} P(x_k)\sum_{b=1}^{n} P(x_b/x_k)\log P(x_b/x_k)= \tag{2.17}$$

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

Источник, генерирующий *n* разных символов – $x_1, x_2, \ldots, x_n$, в этом случае может иметь *n* характерных состояний.

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

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

Из (2.17) получаем

$$H(X) = -\sum_{h=1}^{n}\sum_{j=1}^{n} P(x_h, x_j)\sum_{i=1}^{n} P\left(x_i / x_h, x_j\right)\log P\left(x_i / x_h, x_j\right) =$$

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

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

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

Таким образом, можно сделать следующие выводы относительно степени информативности источников сообщений:

– энтропия источника и количество информации тем больше, чем больше размер алфавита источника;

– энтропия источника зависит от статистических свойств сообщений. Энтропия максимальна, если сообщения источника равновероятны и статистически независимы;

– энтропия источника, вырабатывающего неравновероятные сообщения, всегда меньше максимально достижимой;

– при наличии статистических связей между элементарными сообщениями (памяти источника) его энтропия уменьшается.

В качестве примера рассмотрим источник с алфавитом, состоящим из букв русского языка а, б, в, ..., ю, я. Будем считать для простоты, что размер алфавита источника $K = 2^5 = 32$.

Если бы все буквы русского алфавита имели одинаковую вероятность и были статистически независимы, то средняя энтропия, приходящаяся на один символ, составила бы:

$$H_{\max} = \log_2 32 = 5 \text{ бит/букву}.$$

Если теперь учесть лишь различную вероятность букв в тексте, расчетная энтропия составит:

$$H = 4{,}39 \text{ бит/букву}.$$

С учетом корреляции (статистической связи) между двумя и тремя соседними буквами (после буквы «П» чаще встречается «А» и почти никогда – «Ю» и «Ц») энтропия уменьшится, соответственно, до

$$H = 3{,}52 \text{ бит/букву и } H = 3{,}05 \text{ бит/букву}.$$

Наконец, если учесть корреляцию между восемью и более символами, энтропия уменьшится до $H = 2$ *бит / букву* и далее остается без изменений.

2.3. Избыточность источника сообщений. Основные модели каналов связи

2.3.1. Избыточность источника

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

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

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

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

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

2.3.2. Поток информации источника сообщений

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

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

$$\bar{\tau} = \sum_k P(S_k) \sum_i P(x_i)\, \tau_{x_i},$$

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

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

$$\overline{H}(X)=\frac{H(X)}{\bar{\tau}},\ \ \frac{\text{дв. ед.}}{\text{символ}}.$$

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

$$\overline{H}(X)=\frac{H(X)}{\tau},\ \ \frac{\text{дв. ед.}}{\text{символ}}.$$

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

2.3.3. Основные модели каналов связи

Дискретным каналом называется совокупность средств, предназначенных для передачи дискретных сигналов. Для анализа информационных возможностей удобно пользоваться изученной информационной моделью канала связи, представленной на рис. 1.5. Источник информации создает сообщения, состоящие из последовательности знаков алфавита источника $Z=(z_1,z_2,...,z_n)$, которые в кодирующем устройстве преобразуются в последовательность символов. Объем алфавита символов $X=(x_1,x_2,...,x_m)$, как правило, меньше объема алфавита знаков, но они могут и совпадать. В результате модуляции каждой последовательности символов ставится в соответствие сложный сигнал. Множество сложных сигналов конечно. Они различаются числом, составом и взаимным расположением элементарных сигналов. В результате действия помех сигнал на приемной стороне может отличаться от переданного. Помехи имеют случайный характер и

подчиняются статистическим законам. Удобно условно считать, что помехи создаются некоторым воображаемым источником помех и поступают в линию связи в виде мешающего сигнала *E*. Приемная сторона содержит демодулятор, где сигналы преобразуются в последовательность символов $Y=(y_1,y_2,...,y_m)$, декодирующее устройство, выполняющее ответные функции кодированию, и приемник информации, перерабатывающий принятые сообщения $V=(v_1,v_2,...,v_n)$.

С математической точки зрения дискретный канал можно определить алфавитом единичных элементов на его входе $x_i\,(i=1,2,...,m)$ и выходе $y_j\,(j=1,2,...,m)$, а также вероятностями перехода единичного элемента одного вида (передаваемого) в элемент того же вида или другого вида в пункте приема $P(y_j/x_i)$. Значения вероятностей $P(y_j/x_i)$ зависят от характера ошибок в дискретном канале, т.е. от интенсивности ошибок и их статистического распределения во времени. Если при передаче i-того единичного элемента принят элемент такого же вида $(i=j)$, то считается, что ошибки нет. Если при передаче i-того элемента принят элемент нового вида, который не предусмотрен алфавитом (например, $i=m+1$), то его можно использовать для стирания принятого знака. При $i\neq j$ и $i\neq m+1$ считают, что произошла ошибка.

На рис. 2.2 и 2.3 приведены модели бинарного стирающего канала при отсутствии и при наличии трансформации символов соответственно.

Дискретные каналы классифицируются в зависимости от свойств вероятности перехода $P(y_j/x_i)$. Каналы, в которых $P(y_j/x_i)$, не зависят от времени для любых i,j, называются стационарными. Если $P(y_j/x_i)$ зависят от времени, то каналы называются нестационарными. Каналы, в которых $P(y_j/x_i)$ не зависят от значений ранее принятых элементов, называются каналами без памяти. При зависимости $P(y_j/x_i)$ от значений ранее принятых элементов возникает канал с памятью. Каналы, в которых вероятность перехода $P(y_j/x_i)=const$ не зависит от i,j, называются симметричными. В противном случае канал становится несимметричным. Большинство каналов, образуемых по кабельным и радиорелейным линиям связи, симметричны и обладают памятью. Каналы космической связи симметричны и памятью не обладают. На рис. 2.4 приведена модель бинарного канала без памяти.

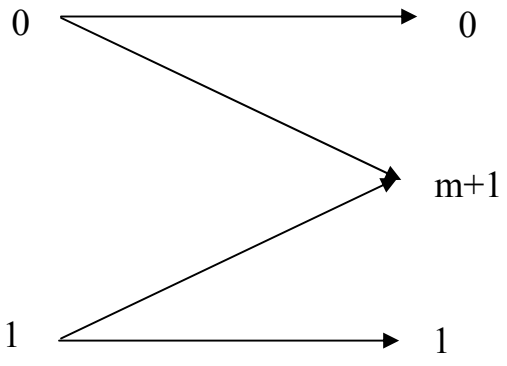


Рис. 2.2. Модель бинарного стирающего канала при отсутствии трансформации символов

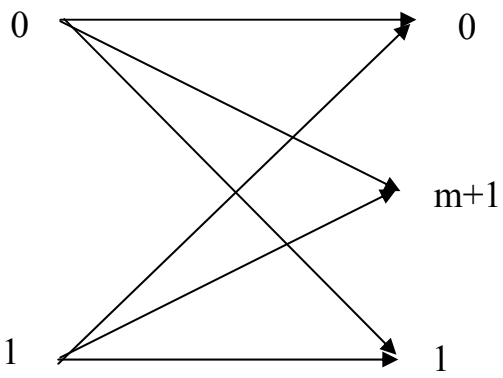


Рис. 2.3. Модель бинарного стирающего канала при наличии трансформации символов

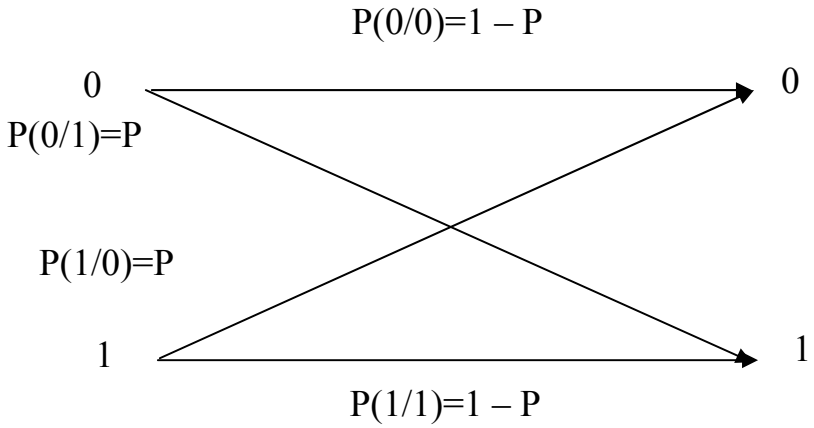


Рис. 2.4. Модель бинарного симметричного канала

Для организации эффективной передачи информации по каналу требуется решение следующих задач:

– определение максимально возможной скорости передачи информации по каналу;

– разработка кодов, позволяющих увеличить скорость передачи информации;

– согласование канала с источником с целью передачи информации с минимальными потерями.

Решение этих задач зависит от свойств источников, уровня и характера помех. Если уровень помех мал, и искажениями сигнала можно пренебречь, канал связи называется каналом без помех.

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

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

$$v_\tau = \frac{1}{\bar{\tau}},$$

где $\bar{\tau}$ – среднее значение длительности символа. При одинаковой продолжительности τ всех символов, передаваемых в канал $\bar{\tau} = \tau$.

Единицей измерения технической скорости служит бод-скорость, при которой за одну секунду передается один символ. Информационная скорость определяется средним количеством информации, которая передается по каналу в единицу времени. Она зависит как от характеристик данного канала связи (объем алфавита используемых символов, техническая скорость их передачи, статистические свойства помех в линии), так и от вероятностей поступающих на вход символов и их статистической взаимосвязи.

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

$$R_t = v_\tau I(Y, X),$$

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

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

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

$$C = \max R_t = v_\tau \max I(Y, X).$$

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

2.3.4. Энтропия источника и энтропия сообщения

Пусть источник информации выдает дискретные сообщения $Z = (z_1, z_2, ..., z_n)$. С помощью кодирующего устройства каждое сообщение превращается в код. Множество символов кода обозначим через

$X = (x_1, x_2, ..., x_m)$. Если исследуется канал связи, то можно не обращаться к источнику информации, а рассматривать лишь источник символов (кодирующее устройство). Тогда возникает необходимость связать свойства источника и отправителя. Эта связь возможна через энтропию.

Под энтропией сообщения будем понимать количество информации, содержащееся в любом усредненном сообщении. Тогда усредненная энтропия сообщения:

$$H(Z) = -\sum_{j=1}^{n} P(z_j) \log_2 P(z_j), \frac{\text{дв.ед.}}{\text{сообщ.}}.$$

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

$$H(X) = -\sum_{j=1}^{m} P(x_j) \log_2 P(x_j), \frac{\text{дв.ед.}}{\text{символ}}.$$

Пример. Пусть передается четыре равновероятных сообщения двоичным неизбыточным кодом. Сообщения отображаются кодом 00, 01, 10, 11. Найдем энтропию сообщения:

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

и энтропию источника

$$H(X) = -\sum_{j=1}^{2} \frac{1}{2} \log_2 \frac{1}{2} = 1, \frac{\text{дв.ед.}}{\text{символ}}.$$

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

Разделим энтропию сообщения на энтропию источника и получим число элементов в коде, т.е.:

$$n = \frac{H(Z)}{H(X)} = \frac{2}{1} = 2.$$

Если данное условие соблюдается, то код называется оптимальным, в противном случае в коде возникает избыточность, и он становится неоптимальным для канала без шума. Для получения оптимального кода необходимо, чтобы символы в нем встречались с равной вероятностью.

2.4. Дискретный канал и его основные характеристики. Согласование характеристик сигнала и канала

2.4.1. Дискретный канал без помех

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

$$I(Y,X) = I(X,Y) = H(X),$$

а максимальное значение:

$$I(Y,X) = H_m(X) ,$$

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

$$P(x_1) = P(x_2) = ... = P(x_m) = 1/m .$$

Известно, что максимальная энтропия выражается в единицах информации на символ сигнала:

$$H_m(X) = \log_a m .$$

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

$$C = v_\tau \log_a m .$$

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

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

$$v_{zm} = \frac{C}{H(Z)} .$$

где v_{zm} измеряется в символах сообщения за единицу времени;

2) не существует метода кодирования, позволяющего сделать эту скорость больше чем v_{zm}.

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

2.4.2. Дискретный канал с помехами

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

$$\begin{bmatrix} P(y_1/x_1) & P(y_2/x_1) & \cdots P(y_m/x_1) \\ P(y_1/x_2) & P(y_2/x_2) & \cdots P(y_m/x_2) \\ \cdots & \cdots & \cdots\cdots \\ P(y_1/x_m) & P(y_2/x_m) & \cdots P(y_m/x_m) \end{bmatrix}$$

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

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

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

В случае независимости отдельных символов сигнала энтропия на выходе линии

$$H\left(Y\right) = -\sum_{j=1}^{m} P\left(y_j\right) \log P\left(y_j\right)$$

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

Средняя условная энтропия:

$$H\left(Y/X\right) = -\sum_{i=1}^{m} P\left(x_i\right) \sum_{j=1}^{m} P\left(y_j / x_i\right) \log_a P\left(y_j / x_i\right).$$

Пропускная способность канала высчитывается по формуле:

$$C = v_\tau \max I(Y,X),$$

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

Пусть требуется определить пропускную способность канала связи, по которому передаются двоичные сигналы со скоростью v_z, если вероятность превращения в результате действия помех каждого из этих сигналов в противоположный равна P (вероятность правильного приема, следовательно, $1 - P$), передаваемые символы предполагаются независимыми.

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

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

Диаграмма (рис. 2.4) показывает возможные варианты передачи и соответствующие им вероятности. Канал такого типа носит название симметричного.

Средняя условная энтропия:

$$H(Y/X) = -\sum_{i=1}^{2} P(x_i) \sum_{j=1}^{2} P(y_j/x_i) \log P(y_j/x_i) =$$

$$= -P(x_1)\left[(1-P)\log(1-P) + P\log P\right] - P(x_2)\left[(1-P)\log(1-P) + P\log P\right] =$$

$$= -\left[(1-P)\log(1-P) + P\log P\right]\left[P(x_1) + P(x_2)\right],$$

очевидно, что:

$$P(x_1) + P(x_2) = 1.$$

Поэтому:

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

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

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

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

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

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

и, следовательно, пропускная способность:

$$C = v_\tau \max(I(Y,X)) = v_\tau[1 + P\log P + (1-P)\log(1-P)].$$

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

$$C_{\max} = v_\tau.$$

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

Минимальное значение пропускная способность имеет при $P = 1/2$. В этом случае, независимо от полученных сигналов, ничего нельзя сказать о том, какой сигнал был послан: имеет место такая ситуация, как если бы в линию связи вообще не посылались сигналы. Тогда, естественно, пропускная способность:

$$C_{\min} = 0.$$

2.4.3. Согласование характеристик сигнала и канала

Сигнал может быть охарактеризован различными параметрами. Таких параметров, вообще говоря, очень много, но для задач, которые приходится решать на практике, существенно лишь небольшое их число.

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

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

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

$$V_x = H_x T_x F_x.$$

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

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

Таким образом, канал также можно охарактеризовать объемом (емкостью):

$$V_k = H_k T_k F_k. \qquad (2.18)$$

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

$$T_x < T_k;\ \ F_x < F_k;\ \ H_x < H_k.$$

То есть сигнал полностью уместится в объеме V_k. При этом, конечно, $V_x < V_k$, однако, только этого условия недостаточно. Тем не менее, если

$V_x < V_k$, но условие (2.18) не выполняется, сигнал может быть определенным образом преобразован, так что передача окажется возможной.

2.5. Вопросы и задания для самопроверки

1. В чем сущность требования аддитивности к мере неопределенности выбора?

2. Назовите основной недостаток меры неопределенности, предложенной Р. Хартли.

3. Укажите достоинства и недостатки способа, предложенного К. Шенноном.

4. Что необходимо учитывать при выборе способа измерения количества информации?

5. В каких единицах измеряется количество информации?

6. Дайте определение энтропии.

7. Назовите основные свойства энтропии дискретного ансамбля.

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

9. Приведите выражение для энтропии двух взаимосвязанных ансамблей.

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

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

12. Когда энтропия источника с двумя состояниями достигает максимума?

13. От чего не зависит энтропия случайного процесса?

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

15. Перечислите свойства количества информации.

16. Дайте определение эргодическому источнику.

17. Запишите выражение для энтропии эргодического источника, когда корреляционные связи имеются между двумя и тремя символами.

18. Что характеризует избыточность источника сообщений?

19. Что понимается под потоком информации источника сообщений?

20. Какой канал называется каналом без помех?

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

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

23. Что понимается под энтропией сообщения и энтропией источника?

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

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

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2
Количественная оценка информации

Теория для практического занятия представлена в разделах 2.1 и 2.2.

Перед выполнением тестовых заданий проводится опрос с использованием вопросов, представленных в разделе 2.5.

Тестовые задания:

1. Какое количество информации несет опыт с единственным исходом:

a) $I = 1$;

b) $I = 0$;

c) $I = 2$.

2. Какому свойству удовлетворяет мера, предложенная Хартли в 1928 г.:

a) мультипликативности;

b) аддитивности;

c)коммутативности.

3. Получение информации – это..:

a) процесс раскрытия неопределенности;

b) процесс передачи неопределенности;

c) процесс обработки неопределенности.

4. Энтропия максимальна:

a) $H_{\max} = \sum_{i=1}^{k} \frac{1}{k} \log \frac{1}{k} = \log k$;

b) $H_{\max} = -\sum_{i=0}^{k} \frac{1}{k} \log \frac{1}{k} = \log k$;

c) $H_{\max} = -\sum_{i=1}^{k} \frac{1}{k} \log \frac{1}{k} = \log k$;

d) $H_{\max} = -\sum_{i=1}^{k} k \log \frac{1}{k} = \log k$.

5. Объединение двух ансамблей характеризуется матрицей:

a) $X = \begin{pmatrix} X_1 & X_2 & \cdots & X_j & \cdots & X_k \\ P_1 & P_2 & \cdots & P_j & \cdots & P_k \end{pmatrix}$;

b) $P(X,Y) = \begin{bmatrix} P(x_1,y_1) & \ldots & P(x_i,y_1) & \ldots & P(x_n,y_1) \\ \ldots\ldots\ldots & \ldots & \ldots\ldots\ldots & \ldots & \ldots\ldots\ldots \\ P(x_1,y_j) & \ldots & P(x_i,y_j) & \ldots & P(x_n,y_j) \\ \ldots\ldots\ldots & \ldots & \ldots\ldots\ldots & \ldots & \ldots\ldots\ldots \\ P(x_1,y_m) & \ldots & P(x_i,y_m) & \ldots & P(x_n,y_m) \end{bmatrix}$;

c) $X = \begin{bmatrix} x_1 & \cdots & x_i & \cdots & x_n \\ P(x_1) & \cdots & P(x_i) & \cdots & P(x_n) \end{bmatrix}$;

d) $\begin{bmatrix} P(y_1/x_1) & P(y_2/x_1) & \cdots & P(y_m/x_1) \\ P(y_1/x_2) & P(y_2/x_2) & \cdots & P(y_m/x_2) \\ \cdots & \cdots & \cdots\cdots & \\ P(y_1/x_m) & P(y_2/x_m) & \cdots & P(y_m/x_m) \end{bmatrix}$

6. Энтропия объединения двух статистически связанных ансамблей X и Y равна:

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

b) $H(X,Y) = H(X) + H(Y/X)$;

c) $H(X,Y) = H(Y) + H(X/Y)$;

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

7. В случае статистической независимости ансамблей частную условную энтропию можно определить по выражению:

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

b) $H(X,Y) = H(X) + H(Y/X)$;

c) $H(X,Y) = H(Y) + H(X/Y)$;

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

8. Какие утверждения относятся к свойствам количества информации:

a) $I(X,Y) = I(Y,X)$;

b) $I(X,Y) \geq 0$;

c) $I(X,Y) = H(X)$;

d) $I(Y,X) = H(X) - H(X/Y)$.

9. Эргодическим источником r-го порядка называется такой источник:

a) у которого вероятность появления некоторого символа x_j не зависит от r предыдущих;

b) у которого вероятность появления некоторого символа x_j зависит от r предыдущих;

c) у которого вероятность появления некоторого символа x_j зависит от предыдущего;

d) у которого вероятность появления некоторого символа x_j не зависит от предыдущего.

10. Энтропия источника сообщений максимальна:

a) если сообщения источника неравновероятны;

b) если сообщения источника равновероятны;

c) если сообщения источника статистически независимы;

d) если сообщения источника статистически зависимы;

Задачи

1. Опыт X имеет три исхода x_1, x_2, x_3 с соответственными вероятностями $P(x_1) = 0{,}2$; $P(x_2)=0{,}5$; $P(x_3)=0{,}3$. Найти точные и средние количества информации, несомые исходами x_1, x_2, x_3.

2. Если предположить, что все буквы русского алфавита имеют одинаковую вероятность и статистически независимы, то какова в этом случае средняя энтропия, приходящаяся на один символ?

3. Для заданных вероятностей появления в тексте различных букв определить энтропию

А	Б	В	Г	Д	Е	Ж	З
$Ра = 0{,}6$	$Рб = 0{,}2$	$Рв = 0{,}1$	$Рг = 0{,}04$	$Рд = 0{,}025$	$Ре = 0{,}015$	$Рж = 0{,}01$	$Рз = 0{,}01$

4. Сигнал состоит из семи двоичных элементов. Определить количество информации в сигнале, когда элементы равновероятны, т.е. $P_1 = P_2 = 1/2$, и когда $P_1 = 3/4$, а $P_2 = 1/4$.

5. Ансамбли событий X и Y объединены. Вероятности совместных событий (x, y) описываются матрицей:

	x_1	x_2	x_3
y_1	0,1	0,2	0,3
y_2	0,25	0	0,15

Определить:

– энтропию ансамблей X и Y;

– энтропию объединенного ансамбля (X, Y);

– условные энтропии ансамблей;

– количество информации, содержащейся в событиях y относительно x.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3
Источники дискретных сообщений

Основная теория для практического занятия представлена в разделах 2.3 и 2.4.

Перед выполнением тестовых заданий проводится опрос с использованием вопросов, представленных в разделе 2.5.

Тестовые задания:

1. Оптимальным называется источник с избыточностью:

a) $R = 1/2$;

b) $R = 1$;

c) $R = 0$;

d) $R = 2$.

2. Энтропию источника, приходящуюся на единицу времени, можно определить по формуле:

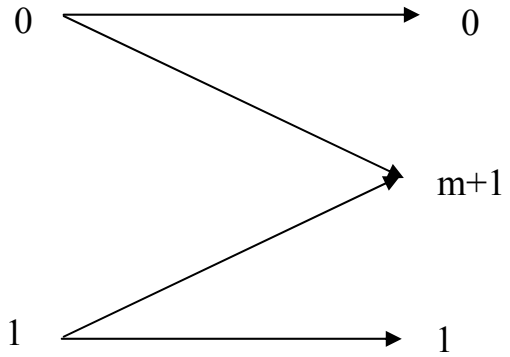
a) $\overline{H}(X) = \frac{\tau}{\tau}$;

b) $\overline{H}(X) = \frac{\bar{\tau}}{\tau}$;

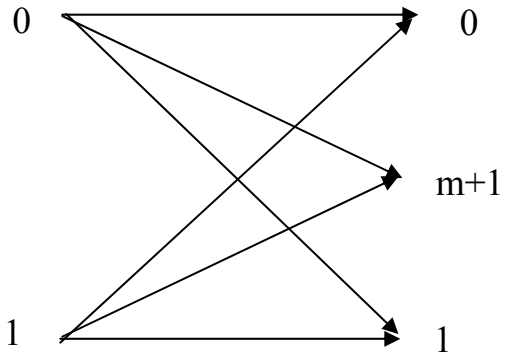
c) $\overline{H}(X) = \frac{H(X)}{\bar{\tau}}$;

d) $\overline{H}(X) = \frac{\bar{\tau}}{H(X)}$.

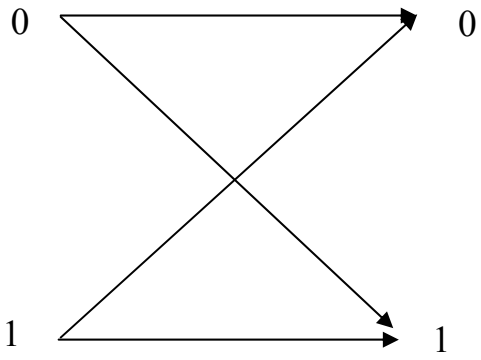
3. Определите модель бинарного стирающего канала при отсутствии трансформации символов:



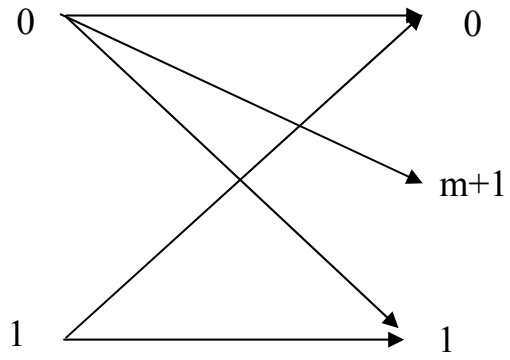
a)



b)



c)



d)

4. Пропускная способность канала определяется:

a) $C = \max H$;

b) $C = \max R_t$;

c) $C = v_\tau \max I(Y, X)$;

d) $C = \max \overline{H}$.

5. Какие параметры сигнала относятся к основным:

a) время передачи сигнала;

b) мощность сигнала;

c) спектр частот.

6. Основные параметры канала связи:

a) время ожидания;

b) время использования канала;

c) полоса пропускания;

d) динамический диапазон канала.

Задачи

Дополнительные формулы для решения задач:

Число типичных последовательностей длиной *M:*

$$N_T = 2^{MH(X)},$$

где *H(X)* – энтропия эргодического источника.

Число всевозможных последовательностей, которое можно составить из *n* букв:

$$N = n^M.$$

Число последовательностей, у которых из *M* мест n_A мест представлено букве *A*, равно числу сочетаний из *M* элементов n_A:

$$C_M^{n_A} = \frac{M!}{n_A!(M - n_A)!}.$$

Вероятность того, что в выработанной источником последовательности длиной M содержится n_A символов A, определяется из биномиального закона:

$$P_{M,n_A} = C_M^{n_A} P^{n_A} (1-P)^{M-n_A}.$$

1. Определить избыточность источника информации для условия задачи № 3 (практическая работа № 2).

2. На контролируемом пункте имеются три объекта, каждый из которых может находиться в одном из двух положений («включен» или «выключен»). С контролируемого пункта передаются сообщения об изменении положений объектов. Наблюдением в течение длительного отрезка времени установлено, что из 100 переданных сообщений 70 относятся к первому объекту, 20 – ко второму и 10 – к третьему. Определить количество информации, содержащейся в одном сообщении. Определить избыточность источников информации.

3. Источник генерирует два равновероятных символа x_1 и x_2, условные вероятности $P(x_1/x_1) = P(x_2/x_2) = 0{,}7$; $P(x_1/x_2) = P(x_2/x_1) = 0{,}3$. Определить энтропию и избыточность источника.

4. Вероятности появления символов источника равны $P(x_1) = 1/2$, $P(x_2) = 1/4$, $P(x_3) = P(x_4) = 1/8$. Корреляционные связи имеют место между двумя соседними символами, которые описываются в таблице. Определить энтропию и избыточность источника.

$x_i,\ x_j$	$P(x_i,x)$	$x_i,\ x_j$	$P(x_i,x_j)$	$x_i,\ x_j$	$P(x_i,\ x_j)$	$x_i,\ x_j$	$P(x_i,\ x_j)$
x_1x_1	13/32	x_2x_1	1/32	x_3x_1	0	x_4x_1	1/16
x_1x_2	3/32	x_2x_2	1/8	x_3x_2	0	x_4x_2	1/32
x_1x_3	0	x_2x_3	3/32	x_3x_3	0	x_4x_3	1/32
x_1x_4	0	x_2x_4	0	x_3x_4	1/8	x_4x_4	0

5. В информационном канале используется алфавит с четырьмя различными символами. Длительность всех символов одинакова и равна $\tau = 1$ мс. Определить пропускную способность канала при отсутствии шумов.

6. В канал связи передаются сообщения длиной $n = 10$ элементов, каждый из которых может принимать $m = 4$ состояния с вероятностями $P_1 = 0{,}2$; $P_2 = 0{,}3$; $P_3 = 0{,}1$; $P_4 = 0{,}4$. Время передачи одного сообщения $\tau = 0{,}1$ с. Определить скорость передачи информации и пропускную способность канала связи.

7. По бинарному каналу передаются сообщения: 1110011101, 1110000001. Длительность каждого элемента сообщения $\tau = 10$ мс. Определить скорость передачи каждого сообщения и пропускную способность двоичного канала.

МОДУЛЬ 3. ЭЛЕМЕНТЫ ТЕОРИИ СЛОЖНОСТИ. ЭЛЕМЕНТЫ ТЕОРИИ ЧИСЕЛ

1. Сложность алгоритмов и проблем.
2. Модульная арифметика.
3. Вопросы и задания для самопроверки.
4. Практическое занятие № 4.

Цель модуля – изучение студентами аспектов теории сложности и теории чисел в приложении к задаче криптографической защиты информации.

В результате изучения модуля студенты должны:

– знать классы сложности алгоритмов и проблем;
– знать основы модульной арифметики;
– уметь оценивать вычислительную сложность алгоритмов;
– иметь представление о классической и вероятностной машинах Тьюринга;
– иметь представление о теоретико-сложностном подходе к определению криптостойкости систем.

3.1. Сложность алгоритмов и проблем

3.1.1. Введение в теорию сложности

Теория сложности обеспечивает методологию анализа вычислительной сложности различных криптографических методов и алгоритмов. Она сравнивает криптографические методы и алгоритмы и определяет их безопасность. Теория информации сообщает о том, что все криптографические алгоритмы (кроме одноразовых блокнотов) могут быть взломаны. Теория сложности сообщает, за какое время алгоритмы могут быть взломаны.

Необходимо сказать, что с точки зрения классической математики проблемы в криптографии являются тривиальными в том смысле, что могут быть решены за конечное число попыток. Однако сведение к конечному числу случаев не имеет особого смысла, если число самих случаев практически не реализуемо, т.е. если система не способна расшифровать некоторое сообщение в разумных временных рамках.

Предположим теперь, что противник атакует криптосистему. Ему известен открытый ключ K_1, но неизвестен соответствующий секретный ключ K_2. Противник перехватил криптограмму d и пытается найти сообщение m, где $d = E_{K_1}(m)$. Поскольку алгоритм шифрования общеизвестен,

противник может просто последовательно перебрать все возможные сообщения длины n, вычислить для каждого такого сообщения m_i криптограмму $d_i = E_{K_1}(m_i)$ и сравнить d_i с d. То сообщение, для которого $d_i = d$ будет искомым открытым текстом. Если повезет, то открытый текст будет найден достаточно быстро. В худшем же случае перебор будет выполнен за время порядка $2^n T(n)$, где $T(n)$ – время, требуемое для вычисления функции E_{K_1} от сообщений длины n. Если сообщения имеют длину порядка 1000 битов, то такой перебор неосуществим на практике ни на каких самых мощных компьютерах.

Рассмотрен лишь один из возможных способов атаки на криптосистему и простейший алгоритм поиска открытого текста, называемый обычно алгоритмом полного перебора. Используется также и другое название: «метод грубой силы». Другой простейший алгоритм поиска открытого текста – угадывание. Этот очевидный алгоритм требует небольших вычислений, но срабатывает с пренебрежимо малой вероятностью (при больших длинах текстов). На самом деле противник может пытаться атаковать криптосистему различными способами и использовать различные, более изощренные алгоритмы поиска открытого текста. Естественно считать криптосистему стойкой, если любой такой алгоритм требует практически неосуществимого объёма вычислений или срабатывает с пренебрежимо малой вероятностью. (При этом противник может использовать не только детерминированные, но и вероятностные алгоритмы.) Это и есть теоретико-сложностной подход к определению стойкости. Для его реализации в отношении того или иного типа криптографических схем необходимо выполнить следующее:

– дать формальное определение схемы данного типа;
– дать формальное определение стойкости схемы;
– доказать стойкость конкретной конструкции схемы данного типа.

Здесь сразу же возникает ряд проблем.

Во-первых, в криптографических схемах используются фиксированные значения параметров. Например, криптосистемы разрабатываются для ключей длины, скажем, в 256 или 512 байтов. Для применения же теоретико-сложностного подхода необходимо, чтобы задача, вычислительную сложность которой предполагается использовать, была массовой. Поэтому в теоретической криптографии рассматриваются математические модели криптографических схем. Эти модели зависят от некоторого параметра, называемого параметром безопасности, который может принимать сколь

угодно большие значения (обычно для простоты предполагается, что параметр безопасности может пробегать весь натуральный ряд).

Во-вторых, определение стойкости криптографической схемы зависит от той задачи, которая стоит перед противником, и от того, какая информация о схеме ему доступна. Поэтому стойкость схем приходится определять и исследовать отдельно для каждого предположения о противнике.

В-третьих, необходимо уточнить, какой объем вычислений можно считать «практически неосуществимым». Из сказанного выше следует, что эта величина не может быть просто константой, она должна быть представлена функцией от растущего параметра безопасности. В соответствии с тезисом Эдмондса алгоритм считается эффективным, если время его выполнения ограничено некоторым полиномом от длины входного слова (в нашем случае – от параметра безопасности). В противном случае говорят, что вычисления по данному алгоритму практически неосуществимы. Заметим также, что сами криптографические схемы должны быть эффективными, т.е. все вычисления, предписанные той или иной схемой, должны выполняться за полиномиальное время.

В-четвертых, необходимо определить, какую вероятность можно считать пренебрежимо малой.

Итак, при наличии всех указанных выше определений, проблема обоснования стойкости криптографической схемы свелась к доказательству отсутствия полиномиального алгоритма, который решает задачу, стоящую перед противником. Но здесь возникает еще одно и весьма серьезное препятствие: современное состояние теории сложности вычислений не позволяет доказывать сверхполиномиальные нижние оценки сложности для конкретных задач рассматриваемого класса. Из этого следует, что на данный момент стойкость криптографических схем может быть установлена лишь с привлечением каких-либо недоказанных предположений. Поэтому основное направление исследований состоит в поиске наиболее слабых достаточных условий (в идеале – необходимых и достаточных) для существования стойких схем каждого из типов.

В основном, рассматриваются предположения двух типов – общие (или теоретико-сложностные) и теоретико-числовые, т.е. предположения о сложности конкретных теоретико-числовых задач. Все эти предположения в литературе обычно называются криптографическими.

3.1.2. Сложность алгоритмов

Сложность алгоритма определяется вычислительными мощностями, необходимыми для его выполнения. Вычислительная сложность алгоритма часто измеряется двумя параметрами: T (временная сложность) и S (пространственная сложность, или требования к памяти). И T, и S обычно представляются в виде функций от n, где n – это размер входных данных. (Существуют и другие способы измерения сложности: количество случайных бит, ширина канала связи, объем данных и т.п.).

Обычно вычислительная сложность алгоритма выражается с помощью нотации – «O», т.е. описывается порядком величины вычислительной сложности. Это просто член разложения функции сложности, быстрее всего растущий с ростом n, все члены низшего порядка игнорируются. Например, если временная сложность данного алгоритма равна $4n^2 + 7n + 12$, то вычислительная сложность порядка n^2, записываемая как $O(n^2)$. Временная сложность, измеренная таким образом, не зависит от реализации. Не нужно знать ни точное время выполнения различных инструкций, ни число битов, используемых для представления различных переменных, ни даже скорость процессора. Один компьютер может быть на 50 процентов быстрее другого, а у третьего шина данных может быть в два раза шире, но сложность алгоритма, оцененная по порядку величины, не изменится. При работе со сложными алгоритмами всем прочим можно пренебречь (с точностью до постоянного множителя) в сравнении со сложностью по порядку величины.

Нотация позволяет увидеть, как объем входных данных влияет на требования к времени и объему памяти. Например, если $T = O(n)$, то удвоение входных данных удвоит время выполнения алгоритма. Если $T = O(2^n)$, то добавление одного бита к входным данным удвоит время выполнения алгоритма.

Обычно алгоритмы классифицируются в соответствии с их временной или пространственной сложностью. Алгоритм называют постоянным, если его сложность не зависит от n: $O(1)$. Алгоритм является линейным, если его временная сложность $O(n)$. Алгоритмы могут быть квадратичными, кубическими и т.д. Все эти алгоритмы – полиномиальны, их сложность – $O(n^m)$, где m – константа. Алгоритмы с полиномиальной временной сложностью называются алгоритмами с полиномиальным временем. Алгоритмы, сложность которых равна $O\left(t^{f(n)}\right)$, где t – константа, большая, чем 1, а

$f(n)$ – некоторая полиномиальная функция от n, называются экспоненциальными.

Подмножество экспоненциальных алгоритмов, сложность которых равна $O\left(c^{f(n)}\right)$, где c – константа, а $f(n)$ возрастает быстрее, чем постоянная, но медленнее, чем линейная функция, называется суперполиномиальным.

В идеале, криптограф хотел бы утверждать, что алгоритм, лучший для взлома спроектированного алгоритма шифрования, обладает экспоненциальной временной сложностью. На практике, самые сильные утверждения, которые могут быть сделаны при текущем состоянии теории вычислительной сложности, имеют форму «все известные алгоритмы вскрытия данной криптосистемы обладают суперполиномиальной временной сложностью». То есть, известные нам алгоритмы вскрытия обладают суперполиномиальной временной сложностью, но пока невозможно доказать, что не может быть открыт алгоритм вскрытия с полиномиальной временной сложностью.

С ростом n временная сложность алгоритмов может стать настолько огромной, что это повлияет па практическую реализуемость алгоритма. В табл. 3.1 показано время выполнения для различных классов алгоритмов при n, равном одному миллиону.

Таблица 3.1

Время выполнения для различных классов алгоритмов

Класс	Сложность	Количество операций $n = 10^6$	Время при 10^6 операций в секунду
Постоянные	$O(1)$	1	1 мкс
Линейные	$O(n)$	10^6	1 с
Квадратичные	$O\left(n^2\right)$	10^{12}	11,6 дня
Кубические	$O\left(n^3\right)$	10^{18}	32000 лет
Экспоненциальные	$O\left(2^n\right)$	10^{301030}	

При условии, что единицей времени для нашего компьютера является микросекунда, компьютер может выполнить постоянный алгоритм за микросекунду, линейный – за секунду, а квадратичный – за 11,6 дня. Выполнение кубического алгоритма потребует 32 тысячи лет, что в принципе реализуемо, компьютер в конце концов получил бы решение. Выполнение

экспоненциального алгоритма тщетно, независимо от экстраполяции роста мощи компьютеров или параллельной обработки.

Временная сложность такого вскрытия грубой силой пропорциональна количеству возможных ключей, которое экспоненциально зависит от длины ключа. Если n – длина ключа, то сложность вскрытия грубой силой равна $O\left(2^n\right)$.

3.1.3. Сложность проблем

Теория сложности также классифицирует и сложность самих проблем, а не только сложность конкретных алгоритмов решения проблемы. Теория рассматривает минимальное время и объем памяти, необходимые для решения самого трудного варианта проблемы на теоретическом компьютере, известном как машина Тьюринга. Для дальнейшего нам потребуется еще понятие вероятностной машины Тьюринга. В обычных машинах Тьюринга (их называют детерминированными, чтобы отличить от вероятностных) новое состояние, в которое машина переходит на очередном шаге, полностью определяется текущим состоянием и тем символом, который обозревает головка на ленте. В вероятностных машинах новое состояние может зависеть еще и от случайной величины, которая принимает значения 0 и 1 с вероятностью 1/2 каждое. Альтернативно можно считать, что вероятностная машина Тьюринга имеет дополнительную случайную ленту, на которой записана бесконечная двоичная случайная строка. Случайная лента может читаться только в одном направлении и переход в новое состояние может зависеть от символа, обозреваемого на этой ленте.

Проблемы, которые можно решить с помощью алгоритмов с полиномиальным временем, называются решаемыми, потому что для разумных входных данных обычно могут быть решены за разумное время. (Точное определение «разумности» зависит от конкретных обстоятельств). Проблемы, которые невозможно решить за полиномиальное время, называются нерешаемыми, потому что вычисление их решений быстро становится невозможным. Нерешаемые проблемы иногда называют трудными. Проблемы, которые могут быть решены только с помощью суперполиномиальных алгоритмов, вычислительно нерешаемы, даже при относительно малых значениях n. Что еще хуже, Алан Тьюринг доказал, что некоторые проблемы принципиально неразрешимы. Даже отвлекаясь от временной сложности алгоритма, невозможно создать алгоритм решения этих проблем.

Проблемы можно разбить на классы в соответствии со сложностью их решения. Самые важные классы и их предполагаемые соотношения показаны на рис. 3.1 (Следует отметить, что лишь малая часть этих утверждений может быть доказана математически).

Находящийся в самом низу класс P состоит из всех проблем, которые можно решить за полиномиальное время. Проблема называется (вычислительно) труднорешаемой, если она не принадлежит классу P. Легкорешаемые проблемы (т.е. проблемы из P) образуют в P несколько подклассов с очевидными определениями: задачи с линейной, квадратичной, кубичной и другой временной сложностью. Неформально понятие легкой проблемы означает, что степени многочлена малы, т.е. в указанных выше пределах.

Класс NP – из всех проблем, которые можно решить за полиномиальное время только на недетермированной машине Тьюринга: вариант обычной машины Тьюринга, которая может делать предположения. Машина предполагает решение проблемы – либо «удачно угадывая», либо перебирая все предположения параллельно – и проверяет свое предположение за полиномиальное время. Проблемы из данного класса обладают тем свойством, что легкорешаемой оказывается проверка: будет удачно угаданное решение проблемы верным или нет. Относительно разложения на множители чисел неизвестно, лежит ли эта проблема в классе P, хотя она из NP: достаточно угадать сомножители и проверить догадку, вычислив их произведение.

Важность NP в криптографии состоит в следующем: многие симметричные алгоритмы и алгоритмы с открытыми ключами могут быть взломаны за недетерминированное полиномиальное время. Для данного шифротекста C, криптоаналитик просто угадывает открытый текст, X, и ключ, k, и за полиномиальное время выполняет алгоритм шифрования со входами X и k и проверяет, равен ли результат C. Это имеет важное теоретическое значение, потому что устанавливает верхнюю границу сложности криптоанализа этих алгоритмов. На практике, конечно же, это выполняемый за полиномиальное время детерминированный алгоритм, который и ищет криптоаналитик. Более того, этот аргумент неприменим ко всем классам шифров, конкретно, он не применим для одноразовых блокнотов – для любого C существует множество пар X, k, дающих C при выполнении алгоритма шифрования, но большинство этих X представляют собой бессмысленные открытые тексты. Для криптографической стойкости необходимо существенно более сильное предположение, чем $\mathrm{P} \neq \mathrm{NP}$.

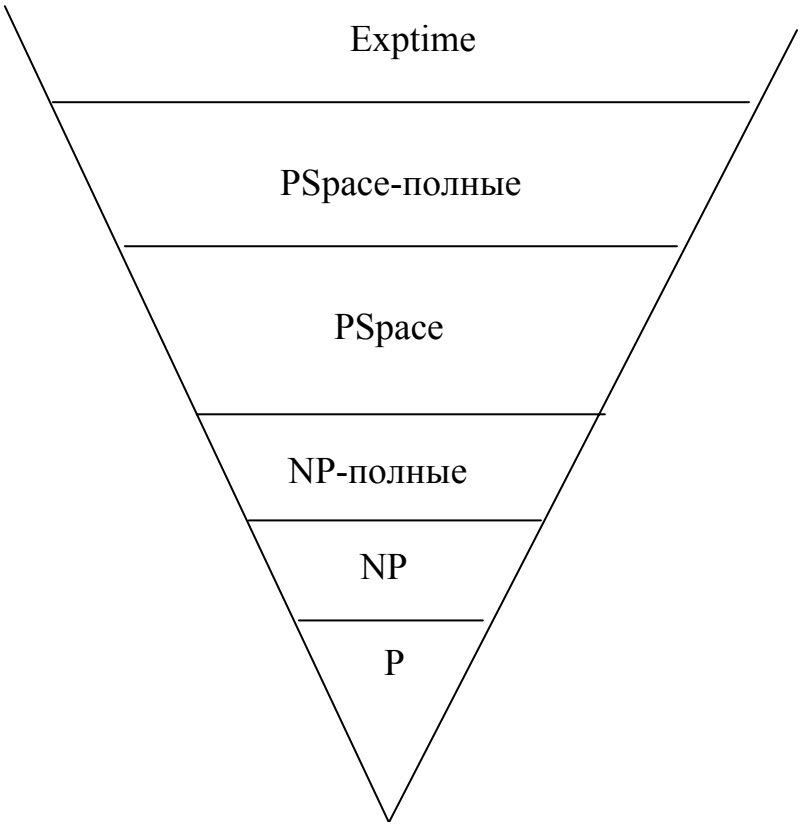


Рис. 3.2. Классы сложности

Класс NP включает класс P, так как любая проблема, решаемая за полиномиальное время на детерминированной машине Тьюринга, будет также решена за полиномиальное время на недетерминированной машине Тьюринга, просто пропускается этап предположения. Если все NP-проблемы решаются за полиномиальное время на детерминированной машине, то P = NP. Хотя кажется очевидным, что некоторые NP проблемы намного сложнее других (вскрытие алгоритма шифрования грубой силой против шифрования произвольного блока шифротекста), никогда не было доказано, что P $\neq$ NP (или что P = NP). Однако, большинство людей, работающих над теорией сложности, убеждены, что эти классы неравны. Что удивительно, можно доказать, что конкретные NP-проблемы настолько же трудны, как и любая проблема этого класса.

Стивен Кук (Steven Cook) доказал, что проблема Выполнимости (Satisfiability problem, существует ли способ присвоить правильные значения входящим в него переменным так, чтобы все выражение стало истиной?) является NP-полной. Это означает, что, если проблема выполнимости решается за полиномиальное время, то P = NP. Наоборот, если может быть доказано, что для любой проблемы класса NP не существует детерминированного алгоритма с полиномиальным временем решения, доказа-

тельство покажет, что и для проблемы выполнимости не существует детерминированного алгоритма с полиномиальным временем решения. В NP нет проблемы труднее, чем проблема выполнимости.

Пространственная сложность определяется аналогично. Когда вход машины Тьюринга имеет длину n, то в исходной ситуации занято n ячеек ленты. В процессе вычислений могут понадобиться новые ячейки, их число и дает пространственную сложность. Полиномиальные ограничения могут быть рассмотрены и в этой ситуации. Это приводит к следующим классам. Класс PSPACE – проблемы класса PSPACE могут быть решены в полиномиальном пространстве, но не обязательно за полиномиальное время. PSPACE включает NP, но ряд проблем PSPACE кажутся сложнее, чем NP. Конечно, и это пока недоказуемо, существует класс проблем, так называемых PSPACE-полных, обладающих следующим свойством: если любая из них является NP-проблемой, то PSPACE = NP, и если любая из них является P-проблемой, то PSPACE = P.

И, наконец, существует класс проблем EXPTIME. Эти проблемы решаются за экспоненциальное время. Может быть действительно доказано, что EXPTIME-полные проблемы не могут быть решены за детерминированное полиномиальное время. Также показано, что P не равно EXPTIME.

NP-полные проблемы:

– Майкл Кэри (Michael Carey) и Дэвид Джонсон (David Johnson) составили список более чем 300 NP-полных проблем. Вот некоторые:

– Проблема путешествующего коммивояжера. Путешествующему коммивояжеру нужно посетить различные города, используя только один бак с горючим (существует максимальное расстояние, которое он может проехать). Существует ли маршрут, позволяющий ему посетить каждый город только один раз, используя этот единственный бак с горючим?

– Проблема тройного брака. В комнате n мужчин, n женщин и n чиновников. Есть список разрешенных браков, записи которого состоят из одного мужчины, одной женщины и одного регистрирующего чиновника. Дан этот список троек, возможно ли построить n браков так, чтобы любой либо сочетался браком только с одним человеком или регистрировал только один брак?

3.2. Модульная арифметика

Целое число a делит другое число b, символически a/b, если и только если $b = da$ для некоторого целого числа d. В этом случае a называется делителем или множителем b. Пусть a – целое число, большее 1.

Тогда a – простое число, если его единственными положительными делителями будут 1 и само a, в противном случае a называется составным. Любое целое $n>1$ может быть представлено единственным образом с точностью до порядка сомножителей как произведение простых. Существенный с точки зрения криптографии факт состоит в том, что неизвестно никакого эффективного алгоритма разложения чисел на множители, хотя, с другой стороны, не было получено и никакой тривиальной нижней оценки временной сложности разложения. Никаких эффективных методов не известно даже в таком простом случае, когда необходимо восстановить два простых числа p и q из их произведения $n=pq$.

Наибольший общий делитель a и b, обозначение – НОД(a,b) или просто (a,b), есть наибольшее целое, делящее одновременно и a, и b. В эквивалентной форме (a,b) есть то единственное натуральное число, которое делит a и b и делится на любое целое, делящее и a и b. Аналогично, наименьшее общее кратное, НОК(a,b), есть наименьшее натуральное число, делящееся и на a и на b.

Наибольший общий делитель может быть вычислен с помощью алгоритма Евклида. Он состоит из следующей цепочки равенств:

$$\begin{aligned} a &= bq_1 + r_1, & 0 &< r_1 < b,\\ b &= r_1q_2 + r_2, & 0 &< r_2 < r_1,\\ r_1 &= r_2q_3 + r_3, & 0 &< r_3 < r_2,\\ r_{k-2} &= r_kq_{k+1} & 0 &< r_k < r_{k-1}\\ r_{k-1} &= r_kq_{k+1} \end{aligned}$$

Остановка гарантируется, поскольку остатки от деления r_i образуют строго убывающую последовательность натуральных чисел. Из этой цепочки немедленно получаем, что r_k есть общий делитель a и b и, более того, что любой общий делитель a и b делит, в свою очередь, r_k. Таким образом, $r_k=(a,b)$.

Оценим теперь временную сложность этого алгоритма. Алгоритм, выполняющий обычное деление, работает за квадратичное время. Итоговая оценка была бы все еще экспоненциальной, если бы выполнялось только $r_{i+1}<r_i$. $r_{i+2}<r_i/2$ для всех i. Это дает верхнюю оценку $2\log_2 a$ для числа равенств. Таким образом, временная сложность в целом кубическая.

Считывая цепочку равенств снизу вверх, найдем суммарно за кубическое время целых числа x и y, такие, что

$$(a,b)=xa+yb.$$

Два целых a и b взаимно просты, если $(a,b)=1$.

Функция Эйлера $\varphi(b), n > 1$, определяется как число неотрицательных $a < n$, таких, что a и n взаимно просты. Имеем $\varphi(1) = 1$ и $\varphi(p^b) = p^b - p^{b-1}$, где p – простое и $b \geq 1$. $\varphi(mn) = \varphi(m)\varphi(n)$, если m и n взаимно просты. Опираясь на эти факты, можно вычислять $\varphi(n)$ для любого n. Это вычисление будет эффективным, если знать разложение n.

Говорят, что a сравнимо с b по модулю m,

$$a \equiv b (\mathrm{mod}\, m),$$

если m делит разность $a - b$. Число m называют модулем. Предполагается, что $m \geq 2$. Для любого целого x, в точности одно из чисел $0, 1, ..., m-1$ сравнимо с x по модулю m. Так определенное число называется наименьшим неотрицательным остатком x по модулю m и обозначается $(x, \mathrm{mod}\, m)$.

Обозначим далее через $[x]$ целую часть x, т.е. наибольшее целое $\leq x$. Имеем

$$(x, \mathrm{mod}\, m) = x - \left[\frac{x}{m}\right] \cdot m.$$

Если a и m взаимно просты, то тогда существуют x и y, такие, что $1 = xa + ym$. Отсюда $xa \equiv 1 (\mathrm{mod}\, m)$. Целое число x будем называть обратным к a по модулю m и обозначать $a^{-1} (\mathrm{mod}\, m)$. Обратное число определяется однозначно, если считать сравнимые числа равными. Сложность нахождения обратного числа примерно такая же, как и у алгоритма Евклида. Отсюда следует, что также и сравнение

$$az \equiv b (\mathrm{mod}\, m), (a, m) = 1$$

может быть решено за кубическое время. Для нахождения z сперва вычисляем $a^{-1} (\mathrm{mod}\, m)$ и умножаем его на b.

Если $(a, m) = 1$, то согласно теореме Эйлера

$$a^{\varphi(m)} \equiv 1 (\mathrm{mod}\, m).$$

Если m простое число, не делящее a, этот результат принимает вид

$$a^{m-1} \equiv 1 (\mathrm{mod}\, m)$$

и называется малой теоремой Ферма.

Если модули m_i попарно взаимно просты, то система сравнений

$$x \equiv a_i (\mathrm{mod}\, m_i),\ i = 1, ..., k,$$

имеет решение x, единственное с точностью до сравнений по модулю $M = m_1 ... m_k$. Этот результат известен как китайская теорема об остатках.

Поле F есть множество, на котором определены операции сложения и умножения, удовлетворяющие обычным требованиям: ассоциативности, коммутативности, дистрибутивности, существования аддитивного 0 и мультипликативной 1, аддитивных обратных и мультипликативных обратных для всех элементов за исключением 0. Рациональные числа и действительные числа образуют поля.

Конечное поле $F(q)$ с q элементами играет важную роль в криптографии. $q = p^h$ для некоторого простого p и $h \geq 1$. Удобный способ представления элементов поля $F(q)$ приводится в параграфе 3.5.

Обозначим через $F^*(q)$ множество всех ненулевых элементов $F(q)$. Некоторый элемент g из $F^*(q)$ называется образующей или порождающим элементом $F^*(q)$, если для всех a из $F^*(q)$ найдется такое целое x, что $g^x = a$ в поле $F^*(q)$. Всего имеется $\varphi(q-1)$ образующих g. Число x при этом будет дискретным логарифмом a по основанию g. Известно, что вычисление дискретных логарифмов (когда g, a и q заданы) примерно такая же труднорешаемая задача, как и разложение на множители.

Рассмотрим некоторое простое $p > 2$. Если элемент a из $F^*(q)$ есть квадрат, т.е. $a = x^2$ для подходящего x, то a называется квадратичным вычетом по модулю p. В противном случае a называется квадратичным невычетом по модулю p. Понятно, что $a, 1 \leq a \leq p-1$, будет квадратичным вычетом по модулю p тогда и только тогда, когда сравнение

$$x^2 \equiv a (\mathrm{mod}\, p)$$

имеет решение. В таком случае также и $-x$ будет решением, т.е. a имеет два квадратичных корня по модулю p. Все квадратичные вычеты находятся возведением в квадрат элементов $1, 2, ..., (p-1)/2$. Таким образом, имеется всего по $(p-1)/2$ квадратичных вычетов и квадратичных невычетов.

Символ Лежандра в случае целого a и простого $p > 2$ определяется соотношением:

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \textit{если } p \textit{ делит } a, \\ 1, & \textit{если } a - \textit{квадратичный вычет по модулю } p, \\ -1, & \textit{если } a - \textit{квадратичный невычет по модулю } p. \end{cases}$$

Понятно, что a можно заменить любым целым числом, сравнимым с $a (\mathrm{mod}\, p)$, не изменяя значения символа Лежандра. Элементарный результат о символе Лежандра есть

$$\left(\frac{a}{p}\right) = a^{(p-1)/2} (\mathrm{mod}\, p) \qquad (3.1)$$

Символ Якоби является обобщением символа Лежандра. Рассмотрим целое a и нечетное $n > 2$. Далее, пусть $n = p_1^{i_1} \ldots p_k^{i_k}$ есть разложение n на простые множители. Тогда символ Якоби определятся как произведение соответствующих символов Лежандра:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{i_1} \ldots \left(\frac{a}{p_k}\right)^{i_k}.$$

Ясно, что и здесь a может быть заменено без изменения символа Якоби на число, сравнимое с $a (\mathrm{mod}\, n)$. Из (3.1) легко вытекает свойство мультипликативности

$$\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right) \cdot \left(\frac{b}{n}\right).$$

Следовательно

$$\left(\frac{ab^2}{n}\right) = \left(\frac{a}{n}\right).$$

Для некоторых значений a символ Якоби вычисляется так:

$$\left(\frac{1}{n}\right) = 1, \left(\frac{1}{n}\right) = (-1)^{(n-1)/2}, \left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}.$$

При вычислении символа Якоби основное сведение выполняется на основе закона взаимности:

$$\left(\frac{m}{n}\right) = (-1)^{(m-1)(n-1)/4} \left(\frac{n}{m}\right),$$

где m и n – нечетные числа больше 2. В эквивалентном виде

$$\left(\frac{m}{n}\right) = \left(\frac{n}{m}\right),$$

если только не выполняется

$$m \equiv n \equiv 3 (\mathrm{mod}\, 4),$$

а в этом случае

$$\left(\frac{m}{n}\right) = -\left(\frac{n}{m}\right).$$

Значение $\left(\frac{m}{n}\right)$ может быть теперь найдено без разложения на множители следующим образом (за исключением случая степеней 2). При необходимости m заменяется на $(m, \mathrm{mod}\, n)$; аналогичная замена осуществ-

ляется также и на последующих шагах процедуры. Применение закона взаимности позволяет уменьшить «знаменатель» в $\left(\frac{m}{n}\right)$. Как и в случае алгоритма Евклида, это уменьшение на одном шаге может быть малым, однако два последовательных шага сокращают знаменатель по крайней мере в два раза. В итоге это дает примерно ту же оценку сложности вычисления $\left(\frac{m}{n}\right)$, как и в случае алгоритма Евклида.

Если p – простое, то описанный метод является быстрым алгоритмом также и для определения: будет ли данное число a квадратичным вычетом или невычетом по модулю p. Никаких подобных быстрых алгоритмов неизвестно в случае, когда вместо простого p имеют дело с произвольным n. Рассмотрим более детально важный для криптографии случай, когда n есть произведение двух простых чисел, $n = pq$.

Как отмечалось выше, половина из чисел $1,..., p-1$ является квадратичными вычетами по модулю p, а другая половина – невычетами. Конечно, аналогичное утверждение верно и для q. С другой стороны, некоторое число a будет квадратичным вычетом по модулю n, т.е. $x^2 \equiv a(\mathrm{mod}\, n)$ для подходящего x, если и только если a будет квадратичным вычетом одновременно и по модулю p, и по модулю q. Все это означает, что в точности половина из чисел a

$$0 < a < n \text{ и } (a,n) = 1$$

удовлетворяет равенству $\left(\frac{a}{n}\right) = +1$, а для другой половины выполняется $\left(\frac{a}{n}\right) = -1$. Более того, половина из чисел a, удовлетворяющих равенству $\left(\frac{a}{n}\right) = +1$, будут квадратичными вычетами по модулю n, а именно те, для которых

$$\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = +1.$$

Другая половина, а именно те, для которых

$$\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$$

будут невычетами. И, похоже, что нет способа выяснить, какой из этих двух случаев имеет место, если только n не будет разложено на множители.

Предположим, нам известно, что a, $0 < a < n$, является квадратичным вычетом по модулю n. Тогда для некоторого x

$$x^2 \equiv a(\mathrm{mod}\, n).$$

Нахождение x, т.е. извлечение квадратных корней по модулю n является весьма важной задачей в криптографии. Давайте снова рассмотрим случай $n = pq$. По предположению, a является квадратичным вычетом как по модулю p, так и по модулю q. Из этого вытекает существование чисел y и z, таких, что

$$(\pm y)^2 \equiv a(\mathrm{mod}\, p) \text{ и } (\pm z)^2 \equiv a(\mathrm{mod}\, q).$$

Более того, y и z могут быть найдены за полиномиальное время (со степенью полинома не выше 4), при условии, что p и q известны. Из сравнений

$$x \equiv \pm y(\mathrm{mod}\, p) \text{ и } x \equiv \pm z(\mathrm{mod}\, q)$$

по китайской теореме об остатках теперь можно получить четыре квадратичных корня x по модулю n. Эти квадратичные корни можно записать в виде $\pm u$ и $\pm w$, где $u \neq \pm w(\mathrm{mod}\, n)$. Назовем такие u и w различными квадратичными корнями. Следующие два факта важны для криптографии. Знание двух различных квадратичных корней позволяет разложить n. Действительно,

$$u^2 - w^2 = (u + w)(u - w) \equiv 0(\mathrm{mod}\, n).$$

Это означает, что n делит $(u + w)(u - w)$. Однако по выбору u и w n не делит ни $u + w$, ни $u - w$. Отсюда следует, что наибольший общий делитель $u + w$ и n (быстровычисляемый алгоритмом Евклида) есть p или q.

Второй важный факт состоит в том, что при $p \equiv q \equiv 3(\mathrm{mod}\, 4)$ два различных квадратичных корня u и w из одного и того же числа a по модулю n имеют различные символы Якоби:

$$\left(\frac{u}{n}\right) = -\left(\frac{w}{n}\right).$$

Это следует из того, что, как было показано выше, или

$$u \equiv w(\mathrm{mod}\, p) \text{ и } u \equiv -w(\mathrm{mod}\, q),$$

или

$$u \equiv -w(\mathrm{mod}\, p) \text{ и } u \equiv w(\mathrm{mod}\, q),$$

а по предположению относительно p и q

$$\left(\frac{-1}{p}\right) = \left(\frac{-1}{q}\right) = -1.$$

3.3. Вопросы и задания для самопроверки

1. Сформулируйте алгоритм полного перебора для поиска открытого текста.

2. Решение каких задач предполагает теоретико-сложностной подход определения криптостойкости систем?

3. Какими параметрами может оцениваться вычислительная сложность алгоритма?

4. Что такое нотация и для чего она используется?

5. Приведите классификацию алгоритмов в соответствии с их временной или пространственной сложностью.

6. Какой величиной характеризуется сложность вскрытия грубой силой, если длина ключа равна m?

7. Поясните принцип работы машины Тьюринга.

8. Какие проблемы называются решаемыми?

9. Приведите классификацию проблем.

10. В чем заключается важность NP-проблем в криптографии?

11. Приведите примеры NP-полных проблем.

12. Что понимают под НОД и НОК?

13. Сформулируйте алгоритм Евклида для вычисления НОД.

14. Дайте определение функции Эйлера.

15. В каком случае числа сравнимы по модулю? Приведите примеры.

16. Что понимают под квадратичным вычетом (невычетом)?

17. Как можно определить символ Лежандра в случае целого a и простого $p > 2$?

18. Запишите выражение для определения символа Якоби.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4
Сложность алгоритмов и проблем. Модульная арифметика

Теория для практического занятия представлена в модуле 3.

Перед выполнением тестовых заданий проводится опрос с использованием вопросов, представленных в разделе 3.3.

Тестовые задания:

1. Временная сложность алгоритма равна $4n^3 + 7n + 12n^4 + 44n$, определить вычислительную сложность:

a) $O(n^2)$; b) $O(n^3)$; c) $O(n\ )$; d) $O(n^4)$.

2. Если $T = O(2^n)$, то какая процедура удвоит время выполнения алгоритма:

a) увеличение входных данных в два раза;

b) увеличение входных данных в 1,5 раза;

c) добавление двух бит к входным данным;

d) добавление одного бита к входным данным.

3. Если сложность определяется величиной $O(n^m)$, то алгоритмы называются:

a) постоянными;

b) линейными;

c) полиномиальными;

d) экспоненциальными.

4. Сложность суперэкспоненциальных алгоритмов оценивается величиной:

a) $O\left(c^{f(n)}\right)$; b) $O(n^m)$; c) $O(n)$; d) $O(1)$.

5. Какой временной сложностью, в идеале, должен обладать алгоритм, лучший для взлома спроектированной системы шифрования?

a) постоянной;

b) линейной;

c) полиномиальной;

d) экспоненциальной.

6. Из какого рода проблем состоит класс NP?

a) из всех проблем, которые можно решить за линейное время только на недетермированной машине Тьюринга;

b) из всех проблем, которые можно решить за полиномиальное время только на детермированной машине Тьюринга;

c) из всех проблем, которые можно решить за полиномиальное время только на недетермированной машине Тьюринга;

d) из всех проблем, которые можно решить за минимально возможное время только на недетермированной машине Тьюринга.

7. Какое выражение справедливо?

a) $P \neq NP$; b) $P = NP$; c) $P \in NP$; d) $P \notin NP$.

8. Если a и m взаимно просты, то тогда существуют x и y, такие, что:

a) $1 = yx + am$; b) $1 = xa + ym$; c) $y = xa + m$; d) $x = a + ym$.

9. В каком случае $a, 1 \le a \le p-1$, является квадратичным вычетом по модулю p:

a) $x^2 \equiv a(\mathrm{mod}\, p)$; b) $x \equiv a(\mathrm{mod}\, p)$; c) $x^2 \equiv a^2(\mathrm{mod}\, p)$; d) $x \equiv a^2(\mathrm{mod}\, p)$.

10. При $p \equiv q \equiv 3(\mathrm{mod}\, 4)$ два различных квадратичных корня u и w из одного и того же числа a по модулю n имеют:

a) $\left(\frac{u}{n}\right) = -\left(\frac{w}{n}\right)$; b) $\left(\frac{u}{n}\right) = \left(\frac{w}{n}\right)$; c) $\left(\frac{u}{w}\right) = -\left(\frac{w}{n}\right)$; d) $\left(\frac{u}{w}\right) = \left(\frac{w}{n}\right)$.

Задачи

1. Определить вычислительную сложность алгоритма умножения вектора на квадратную матрицу.

2. Определить вычислительную сложность умножения прямоугольных матриц.

3. По алгоритму Евклида найти НОД чисел: 1065 и 45; 6066 и 478.

4. Определить, какие числа являются сравнимыми по модулю 6: 5 и 11; 11 и 22; 17 и –1.

5. Определить все квадратичные вычеты и невычеты, если $p = 11$, $p = 19$.

МОДУЛЬ 4. ОСНОВЫ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

1. Основные понятия и определения.
2. Шифры перестановки. Шифры простой замены.
3. Шифрование методом гаммирования. Шифры сложной замены.
4. Вопросы и задания для самопроверки.
5. Практическое занятие № 5.
6. Практическое занятие № 6.
7. Практическое занятие № 7.

Цель модуля – изучение студентами традиционных методов шифрования и получение навыков по их использованию для криптографической защиты информации.

В результате изучения модуля студенты должны:

– знать требования, предъявляемые к криптосистемам;

– знать методы шифрования с использованием шифров перестановки;

– знать алгоритмы шифрования с использованием простой и сложной замены;

– знать технологию шифрования гаммированием и методы получения гаммы;

– уметь использовать шифры перестановки, простой и сложной замены и гаммирования для криптографического кодирования;

– иметь представление о надежности шифрования информации классическими методами.

4.1. Основные понятия и определения

4.1.1. Введение в криптографию

Проблема защиты информации путем ее преобразования, исключающего ее прочтение посторонним лицом, является актуальной с давних времен. Бурное развитие криптографические системы получили в годы первой и второй мировых войн. Начиная с послевоенного времени и по нынешний день, появление вычислительных средств ускорило разработку и совершенствование криптографических методов. Почему проблема использования криптографических методов в информационных системах (ИС) стала в настоящий момент особо актуальна?

С одной стороны, расширилось использование компьютерных сетей, в частности глобальной сети Интернет, по которым передаются большие объемы информации государственного, военного, коммерческого и частного характера, не допускающего возможность доступа к ней посторонних лиц.

С другой стороны, появление новых мощных компьютеров, технологий сетевых и нейронных вычислений сделало возможным дискредитацию криптографических систем еще недавно считавшихся практически не раскрываемыми. Есть три возможности криптозащиты данных:

– создать абсолютно надежный, недоступный для других канал связи между абонентами. При современном уровне развития науки и техники сделать такой канал связи между удаленными абонентами для неоднократной передачи больших объемов информации практически нереально;

– использовать общедоступный канал связи, но скрыть сам факт передачи информации. Разработкой средств и методов скрытия факта передачи сообщения занимается стенография;

– использовать общедоступный канал связи, но передавать нужную информацию в преобразованном виде, чтобы восстановить ее мог только адресат. Разработкой методов преобразования (шифрования) информации с целью ее защиты от незаконных пользователей занимается криптография. Такие методы и способы преобразования информации называются шифрами.

Проблемой защиты информации путем ее преобразования занимается криптология (kryptos – тайный, logos – наука). Криптология разделяется на два направления – криптографию и криптоанализ. Цели этих направлений прямо противоположны.

Криптография – прикладная наука, она использует самые последние достижения фундаментальных наук и, в первую очередь, математики. С другой стороны, все конкретные задачи криптографии существенно зависят от уровня развития техники и технологии, от применяемых средств связи и способов передачи информации.

Криптоанализ – наука (и практика ее применения) о методах и способах вскрытия шифров. Сфера интересов криптоанализа – исследование возможности расшифровывания информации без знания ключей.

Современная криптография включает в себя четыре крупных раздела:

1. Симметричные криптосистемы.
2. Криптосистемы с открытым ключом.
3. Системы электронной подписи.
4. Управление ключами.

Основные направления использования криптографических методов – передача конфиденциальной информации по каналам связи (например, электронная почта), установление подлинности передаваемых сообщений, хранение информации (документов баз данных) на носителях в зашифрованном виде.

В качестве информации, подлежащей шифрованию и дешифрированию, будут рассматриваться тексты, построенные на некотором алфавите. Под этими терминами понимается следующее:

Алфавит – конечное множество используемых для кодирования информации знаков.

Текст – упорядоченный набор из элементов алфавита.

В качестве примеров алфавитов, используемых в современных ИС можно привести:

алфавит Z_{33} – 32 буквы русского алфавита и пробел;

алфавит Z_{256} – символы, входящие в стандартные коды ASCII и КОИ-8;

бинарный алфавит $Z_2 = \{0, 1\}$;

восьмеричный алфавит или шестнадцатеричный алфавит.

Процесс кодирования сообщения называется шифрованием (или зашифровкой), а процесс декодирования – расшифровыванием (или расшифровкой). Само кодированное сообщение называется шифрованным (или просто шифровкой), а применяемый метод называется шифром.

Основное требование к шифру состоит в том, чтобы расшифровка (и, может быть, зашифровка) была возможна только при наличии санкции, то есть некоторой дополнительной информации (или устройства), которая называется ключом шифра. Процесс декодирования шифровки без ключа называется дешифрированием (или дешифрацией, или просто раскрытием шифра).

Ключ – информация, необходимая для беспрепятственного шифрования и дешифрирования текстов.

Криптографическая система представляет собой семейство **T** преобразований открытого текста. Члены этого семейства индексируются или обозначаются символом k; параметр k является ключом. Пространство ключей K – это набор возможных значений ключа. Обычно ключ представляет собой последовательный ряд букв алфавита. Криптосистемы разделяются на симметричные и с открытым ключом.

В симметричных криптосистемах и для шифрования, и для дешифрирования используется один и тот же ключ.

Термины распределения ключей и управления ключами относятся к процессам системы обработки информации, содержанием которых является составление и распределение ключей между пользователями.

Электронной (цифровой) подписью называется присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и подлинность сообщения.

Например, пусть пользователю A необходимо подписать сообщение x. Он, зная секрет K, находит такое y, что $F_K(y) = x$, и вместе с сообщением x посылает y пользователю B в качестве своей цифровой подписи. Пользователь B хранит y в качестве доказательства того, что A подписал сообщение x.

Сообщение, подписанное цифровой подписью, можно представлять себе как пару (x, y), где x – сообщение, y – решение уравнения.

$F_K(y) = x$, $F_K: X \rightarrow Y$ – функция с секретом, известная всем взаимодействующим абонентам. Из определения функции F_K очевидны следующие полезные свойства цифровой подписи:

1) подписать сообщение x, т.е. решить уравнение $F_K(y) = x$, может только абонент – обладатель данного секрета K; другими словами, подделать подпись невозможно;

2) проверить подлинность подписи может любой абонент, знающий открытый ключ, т.е. саму функцию F_K;

3) при возникновении споров отказаться от подписи невозможно в силу ее подлинности;

4) подписанные сообщения можно, не опасаясь ущерба, пересылать по любым каналам связи.

Криптостойкостью называется характеристика шифра, определяющая его стойкость к дешифрированию без знания ключа (т.е. криптоанализу). Имеется несколько показателей криптостойкости, среди которых:

– количество всех возможных ключей;

– среднее время, необходимое для криптоанализа.

Преобразование T_k определяется соответствующим алгоритмом и значением параметра k. Эффективность шифрования с целью защиты информации зависит от сохранения тайны ключа и криптостойкости шифра.

Таким образом, криптография представляет собой совокупность методов кодирования данных, направленных на то, чтобы сделать эти данные бесполезными для противника. Такое кодирование позволяет решить две главные проблемы защиты данных: проблему секретности – лишение противника возможности извлечь информацию из канала передачи и проблему

имитостойкости – лишение противника возможности ввести ложную информацию в канал передачи или изменить сообщение так, чтобы изменился его смысл.

4.1.2. Основные модели криптосистем

Проблемы секретности и имитостойкости тесно связаны между собой, поэтому методы решения одной из них часто применимы для решения другой. Из двух названных проблем секретность рассматривается первой, как наиболее исследованная на протяжении веков. На рис. 4.1 представлена модель канала передачи данных, обеспечивающая секретность благодаря криптографическому кодированию.

Источник информации генерирует открытый текст или незашифрованное сообщение M, которое должно быть передано соответствующему получателю по незащищенному каналу, за которым следит перехватчик. Для того чтобы перехватчик не смог распознать сообщение M, отправитель шифрует (кодирует) его с помощью обратимого кодирования (преобразования) S_K и получает криптограмму или шифрованный текст $E = S_K(M)$, который отправляет получателю.

Законный получатель, приняв криптограмму E, декодирует (расшифровывает) ее с помощью обратного преобразования S_K^{-1} и получает исходное сообщение в виде открытого сообщения M: $S_K(E) = S_K^{-1}(S_K(M)) = M$.

Преобразование S_K выбирается из семейства криптографических преобразований, называемых криптографической системой или общей системой.

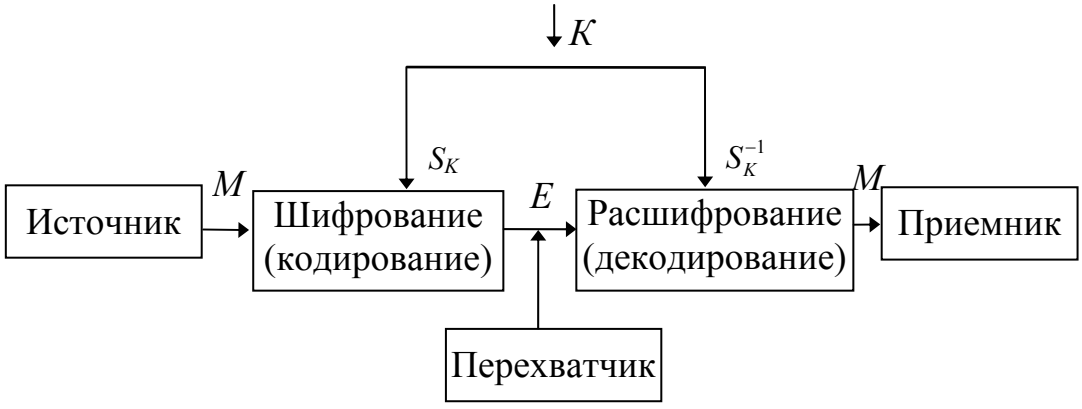


Рис. 4.1. Модель канала передачи с криптографическим кодированием

Параметр, выбираемый в качестве отдельного преобразования, называется криптографическим ключом или просто ключом. Общая система – это набор инструкций (часть аппаратуры или программа ЭВМ), с помощью

которой можно закодировать открытый текст и декодировать шифрованный текст различными способами, один из которых выбирается с помощью конкретного ключа. Формально, криптографическая система – это однопараметрическое семейство обратимых преобразований кодирования $S_к$: $M \to E$ из пространства M сообщений открытых данных в пространство E (закодированных шифрованных сообщений). Причем ключ K_i выбирается из конечного множества K, называемого пространством ключей.

Обычно общая система, являющаяся семейством преобразований, рассматривается как общедоступная система. С одной стороны, то, что открытая для всех часть называется общей системой, отражает очень важное правило техники криптографического кодирования: защищенность системы не должна зависеть от секретности чего-либо такого, что нельзя быстро изменить в случае утечки секретной информации. Обычно общая система является некоторой совокупностью аппаратуры, которую можно изменить только со значительными затратами времени и средств, тогда как ключ является легко и просто изменяемым объектом. Криптографическая система подобна кодовому замку: структура замка известна любому, кто его приобрел, однако конкретная используемая комбинация неизвестна и может быть изменена всякий раз, когда есть подозрение, что она стала известна постороннему лицу. Даже если противник знает множество всех возможных комбинаций, он может все же оказаться не в состоянии определить, какая из них правильная.

Поскольку вся секретность сосредоточена в секретности ключа, то его надо передавать отправителю и получателю по защищенному каналу распространения ключей. На рис. 4.1 этот канал показан экранированной линией. В системе используются одинаковые секретные ключи в блоке шифрования и блоке расшифровывания.

На рис. 4.2 представлена модель канала передачи данных, обеспечивающая секретность с использованием асимметричного криптографического кодирования. В этой криптосистеме один из ключей является открытым, а другой – секретным.

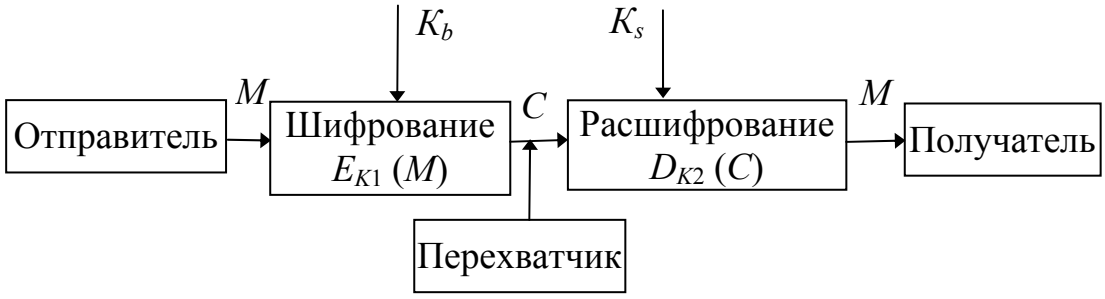


Рис. 4.2. Обобщенная схема асимметричной криптосистемы

В асимметричной криптосистеме для зашифровывания данных используется один ключ, а для расшифровывания – другой ключ (отсюда и название – асимметричная). Первый ключ является открытым и может быть опубликован для использования всеми пользователями системы, которые зашифровывают данные. Расшифровывание данных с помощью открытого ключа невозможно.

Для расшифровывания данных получатель зашифрованной информации использует второй ключ, который является секретным. Разумеется, ключ расшифровывания не может быть определен из ключа зашифровывания.

В асимметричных системах нет необходимости в защищенном канале для передачи ключей, так как открытый ключ передается по незащищенному каналу.

Любая попытка со стороны перехватчика расшифровать криптограмму E для получения открытого текста M или зашифровать свой собственный текст M' для получения приемлемой криптограммы E' без получения ключа из канала распространения ключей называется криптоанализом. Если криптоанализ невозможен и криптоаналитик не может вывести M из E или E' из M' без предварительного получения ключа, то такая криптографическая система называется криптостойкой.

4.1.3. Требования к криптосистемам

Процесс криптографического закрытия данных может осуществляться как программно, так и аппаратно. Аппаратная реализация отличается существенно большей стоимостью, однако ей присущи и преимущества: высокая производительность, простота, защищенность и т.д. Программная реализация более практична, допускает известную гибкость в использовании.

Для современных криптографических систем защиты информации сформулированы следующие общепринятые требования:

– зашифрованное сообщение должно поддаваться чтению только при наличии ключа;

– число операций, необходимых для определения использованного ключа шифрования по фрагменту шифрованного сообщения и соответствующего ему открытого текста, должно быть не меньше общего числа возможных ключей;

– число операций, необходимых для расшифровывания информации путем перебора всевозможных ключей должно иметь строгую нижнюю оценку и выходить за пределы возможностей современных компьютеров (с учетом возможности использования сетевых вычислений);

– знание алгоритма шифрования не должно влиять на надежность защиты;

– незначительное изменение ключа должно приводить к существенному изменению вида зашифрованного сообщения даже при использовании одного и того же ключа;

– структурные элементы алгоритма шифрования должны быть неизменными;

– дополнительные биты, вводимые в сообщение в процессе шифрования, должны быть полностью и надежно скрыты в шифрованном тексте;

– длина шифрованного текста должна быть равной длине исходного текста;

– не должно быть простых и легко устанавливаемых зависимостей между ключами, последовательно используемых в процессе шифрования;

– любой ключ из множества возможных должен обеспечивать надежную защиту информации;

– алгоритм должен допускать как программную, так и аппаратную реализацию, при этом изменение длины ключа не должно вести к качественному ухудшению алгоритма шифрования.

Шеннон определил точную математическую модель понятия безопасности криптосистемы. Смысл работы криптоаналитика состоит в определении ключа К, открытого текста Р или и того, и другого. Однако, его может устроить и некоторая вероятностная информация о Р: является ли этот открытый текст оцифрованным звуком, немецким текстом, данными электронных таблиц или еще чем-нибудь.

В реальном криптоанализе у криптоаналитика есть некоторая вероятностная информация о Р еще до начала работы. Он, скорее всего, знает язык открытого текста. Этот язык обладает определенной, связанной с ним избыточностью. Если это сообщения для друга, оно, возможно, начинается словами «Дорогой друг». Целью криптоаналитика является изменение вероятностей, связанных с каждым возможным открытым текстом. В конце концов, из всех возможных открытых текстов будет выбран один конкретный (или, по крайней мере, весьма вероятный).

Существуют криптосистемы, достигающие совершенной безопасности. Такой является криптосистема, в которой шифротекст не дает никакой информации об открытом тексте (кроме, возможно, его длины). Шеннон теоретически показал, что такое возможно только, если число возможных ключей также велико, как и число возможных сообщений. Другими словами, ключ должен быть не короче самого сообщения и не может использо-

ваться повторно. Это означает, что единственной системой, которая достигает идеальной безопасности, может быть только криптосистема с одноразовым блокнотом. За исключением идеально безопасных систем, шифротекст неизбежно даст определенную информацию о соответствующем шифротексте. Хороший криптографический алгоритм сохраняет минимум этой информации, хороший криптоаналитик пользуется этой информацией для определения открытого текста. Криптоаналитики используют естественную избыточность языка для уменьшения числа возможных открытых текстов. Чем избыточнее язык, тем легче его криптоанализировать. По этой причине многие криптографические реализации перед шифрованием используют программы сжатия для уменьшения размера текста. Сжатие уменьшает избыточность сообщения вместе с объемом работы, необходимым для его шифрования и дешифрирования.

В общем случае, чем больше энтропия, тем тяжелее взломать криптосистему.

Двумя основными методами маскировки избыточности открытого текста сообщения, согласно Шеннону, служат путаница и диффузия. Путаница маскирует связь между открытым текстом и шифротекстом. Она затрудняет попытки найти в шифротексте избыточность и статистические закономерности. Простейшим путем создать путаницу является подстановка. В простом подстановочном шифре, например, шифре Цезаря, все одинаковые буквы открытого текста заменяются другими одинаковыми буквами шифротекста. Современные подстановочные шифры являются более сложными: длинный блок открытого текста заменяется блоком шифротекста, и способ замены меняется с каждым битом открытого текста или ключа. Такого типа подстановки обычно недостаточно – сложный алгоритм немецкой Энигмы был взломан в ходе второй мировой войны.

Диффузия рассеивает избыточность открытого текста, распространяя ее по всему шифротексту. Криптоаналитику потребуется немало времени для поиска избыточности. Простейшим способом создать диффузию является транспозиция (также называемая перестановкой). Простой перестановочный шифр только переставляет буквы открытого текста. Современные шифры также выполняют такую перестановку, но они также используют другие формы диффузии, которые позволяют разбросать части сообщения по всему сообщению.

Потоковые шифры используют только путаницу, хотя ряд схем с обратной связью добавляют диффузию. Блочные алгоритмы применяют и путаницу, и диффузию. Как правило, диффузию саму по себе несложно

взломать (хотя шифры с двойной перестановкой оказываются устойчивее, чем другие некомпьютерные системы).

4.2. Шифры перестановки. Шифры простой замены

4.2.1. Шифры перестановки

Перестановкой δ набора целых чисел *(*0, 1, ..., $N-1$*)* называется его переупорядочение. Для того чтобы показать, что целое i перемещено из позиции i в позицию $\delta(i)$, где $0 \le (i) < N$, будем использовать запись:

$\delta = (\delta(0), \delta(1), ..., \delta(N-1))$.

Число перестановок из (0, 1, ..., $N-1$) равно $N! = 1*2*...*(N-1)*N$. Введем обозначение δ для взаимно однозначного отображения (гомоморфизма) набора $S = \{S_0, S_1, ..., S_{N-1}\}$, состоящего из n элементов, на себя.

$\delta: S \to S$

$\delta: S_i \to S_{\delta(i)}$, $0 \le i < N$

Будем говорить, что в этом смысле δ является *перестановкой элементов S*. И, наоборот, автоморфизм S соответствует перестановке целых чисел (0, 1, 2, …, $N-1$).

Криптографическим преобразованием T для алфавита Z_m называется последовательность автоморфизмов: $T = \{T^{(n)} : 1 \le n < \infty\}$

$T^{(n)} : Z_{m,n} > Z_{m,n}$, $1 \le n < \infty$.

Каждое $T^{(n)}$ является, таким образом, перестановкой n-грамм из $Z_{m,n}$. Поскольку $T^{(i)}$ и $T^{(j)}$ могут быть определены независимо при $i \ne j$, число криптографических преобразований исходного текста размерности n равно $(m^n)!$. Оно возрастает непропорционально при увеличении m и n: так, при $m = 33$ и $n = 2$ число различных криптографических преобразований равно 1089!. Отсюда следует, что потенциально существует большое число отображений исходного текста в шифрованный.

Практическая реализация криптографических систем требует, чтобы преобразования $\{T_k : k \in K\}$ были определены алгоритмами, зависящими от относительно небольшого числа параметров (ключей).

Таким образом, при шифровании перестановкой символы шифруемого текста переставляются по определенному правилу в пределах блока этого текста. Шифры перестановки являются самыми простыми и, вероятно, самыми древними шифрами.

Шифрующие таблицы

В качестве ключа в шифрующих таблицах используются: размер таблицы, слово или фраза, задающие перестановку, особенности структуры таблицы.

Г	М	С	Ц	П	К	А
О	И	Т	А	У	И	Р
Р	Н	О	Р	Б	Б	У
О	С	Л	Е	Л	Е	С
Д	К	И	С	И	Л	Ь

Рис. 4.3. Заполнение сообщения в таблице из 5 строк и 7 столбцов

Одним из самых примитивных табличных шифров перестановки является простая перестановка, для которой ключом служит размер таблицы. Например, сообщение ГОРОД МИНСК СТОЛИЦА РЕСПУБЛИКИ БЕЛАРУСЬ записывается в таблицу поочередно по столбцам. Результат заполнения таблицы из 5 строк и 7 столбцов показан на рис. 4.3. После заполнения таблицы текстом сообщения по столбцам для формирования шифротекста считывают содержимое таблицы по строкам. Если шифротекст записывать группами по пять букв, получается такое шифрованное сообщение: ГМСЦП КАОИТ АУИРР НОРББ УОСЛЕ ЛЕСДК ИСИЛЬ.

Естественно, отправитель и получатель сообщения должны заранее условиться об общем ключе в виде размера таблицы. Следует заметить, что объединение букв шифротекста в 5-буквенные группы не входит в ключ шифра и осуществляется для удобства записи несмыслового текста. При расшифровывании действия выполняют в обратном порядке.

Несколько большей стойкостью к раскрытию обладает метод шифрования, называемый одиночной перестановкой по ключу. Этот метод отличается от предыдущего тем, что столбцы таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы. Применим в качестве ключа, например, слово РАДОСТЬ, а текст сообщения возьмем из предыдущего примера. На рис. 4.4 показаны две таблицы, заполненные текстом сообщения и ключевым словом, при этом левая таблица соответствует заполнению до перестановки, а правая таблица – заполнению после перестановки.

Ключ →

Р	А	Д	О	С	Т	Ь
4	1	2	3	5	6	7
Г	М	С	Ц	П	К	О
2	И	Т	А	У	И	Р
Р	Н	О	Р	Б	Б	У
О	С	Л	Е	Л	Е	С
Д	К	И	С	И	Л	Ь

До перестановки

А	Д	О	Р	С	Т	Ь
1	2	3	4	5	6	7
М	С	Ц	Г	П	К	О
И	Т	А	О	У	И	Р
Н	О	Р	Р	Б	Б	У
С	Л	Е	О	Л	Е	С
К	И	С	Д	И	Л	Ь

После перестановки

Рис. 4.4. Таблицы, заполненные ключевым словом и текстом сообщения

В верхней строке левой таблицы записан ключ, а номера под буквами ключа определены в соответствии с естественным порядком соответствующих букв ключа в алфавите. Если бы в ключе встретились одинаковые буквы, они были бы пронумерованы слева направо. В правой таблице столбцы переставлены в соответствии с упорядоченными номерами букв ключа. При считывании содержимого правой таблицы по строкам и записи шифротекста группами по пять букв получим шифрованное сообщение:

МСЦГП КОИТА ОУИРН ОРРББ УСЛЕО ЛЕСКИ СДИЛЬ

Для обеспечения дополнительной скрытности можно повторно зашифровать сообщение, которое уже было зашифровано. Такой метод шифрования называется двойной перестановкой. В случае двойной перестановки столбцов и строк таблицы перестановки определяются отдельно для столбцов и отдельно для строк. Сначала в таблицу записывается текст сообщения, а потом поочередно переставляются столбцы, а затем строки. При расшифровывании порядок перестановок должен быть обратным. Пример выполнения шифрования методом двойной перестановки показан на рис. 4.5. Если считывать шифротекст из правой таблицы построчно блоками по четыре буквы, то получится следующее: ИАЙК ВДОА ИСНМ ТЗОВ.

Ключом к шифру двойной перестановки служит последовательность номеров столбцов и номеров строк исходной таблицы (в нашем примере последовательности 4132 и 3142 соответственно).

Исходная таблица

	4	1	3	2
3	М	И	Н	С
1	К	И	Й	А
4	В	Т	О	З
2	А	В	О	Д

Перестановка столбцов

	1	2	3	4
3	И	С	Н	М
1	И	А	Й	К
4	Т	З	О	В
2	В	Д	О	А

Перестановка строк

	1	2	3	4
1	И	А	Й	К
2	В	Д	О	А
3	И	С	Н	М
4	Т	З	О	В

Рис. 4.5. Шифрование методом двойной перестановки

Число вариантов двойной перестановки быстро возрастает при увеличении размера таблицы: для таблицы 3×3 – 36 вариантов, для таблицы 4×4 – 576 вариантов, для таблицы 5×5 – 14400 вариантов.

Однако двойная перестановка не отличается высокой стойкостью и сравнительно просто «взламывается» при любом размере таблицы шифрования.

4.2.2. Шифры простой замены

Системы подстановок

Определение: Подстановкой π на алфавите Z_m называется автоморфизм Z_m, при котором буквы исходного текста t замещены буквами шифрованного текста $\pi(t)$: $Z_m \rightarrow Z_m$; $\pi: t \rightarrow \pi(t)$.

Набор всех подстановок называется симметрической группой Z_m и будет в дальнейшем обозначаться как $SYM(Z_m)$.

Утверждение*:* $SYM(Z_m)$ с операцией произведения является группой, т.е. операцией, обладающей следующими свойствами:

1. Замкнутость: произведение подстановок $\pi_1 \pi_2$ является подстановкой:

$$\pi : t \rightarrow \pi_1 (\pi_2(t)).$$

2. Ассоциативность: результат произведения $\pi_1 \pi_2 \pi_3$ не зависит от порядка расстановки скобок:

$$(\pi_1 \pi_2)\ \pi_3 = \pi_1\ (\pi_2\ \pi_3).$$

3. Существование нейтрального элемента: подстановка i, определяемая как $i(t) = t$, $0 \le t < m$, является нейтральным элементом $SYM(Z_m)$ по операции умножения: $i\pi = \pi\ i$ для $\forall\ SYM(Z_m)$.

4. Существование обратного: для любой подстановки π существует единственная обратная подстановка π^{-1}, удовлетворяющая условию

$$\pi \pi^{-1} = \pi^{-1} \pi = i.$$

Число возможных подстановок в симметрической группе Z_m называется порядком $SYM(Z_m)$ и равно $m!$.

Ключом подстановки k для Z_m называется последовательность элементов симметрической группы Z_m:

$$k = (p_0, p_1, ..., p_{n-1}, ...)\ p_n \in SYM(Z_m),\ 0 \le n < \infty$$

Подстановка, определяемая ключом k, является криптографическим преобразованием T_k, при помощи которого осуществляется преобразова-

ние n-граммы исходного текста $(x_0, x_1, ..., x_{n-1})$ в n-грамму шифрованного текста $(y_0, y_1, ..., y_{n-1})$:

$$y_i = p(x_i),\ 0 \le i < n,$$

где n – произвольное (n = 1, 2, ..). T_k называется моноалфавитной подстановкой, если p неизменно при любом i (i = 0, 1, ...,), в противном случае T_k называется многоалфавитной подстановкой.

К наиболее существенным особенностям подстановки T_k относятся следующие:

1. Исходный текст шифруется посимвольно. Шифрования n-граммы $(x_0, x_1, ..., x_{n-1})$ и ее префикса $(x_0, x_1, ..., x_{S-1})$ связаны соотношениями

$$T_k(x_0, x_1, ..., x_{n-1}) = (y_0, y_1, ..., y_{n-1})$$
$$T_k(x_0, x_1, ..., x_{S-1}) = (y_0, y_1, ..., y_{S-1})$$

2. Буква шифрованного текста y_i является функцией только i-той компоненты ключа p_i и i-той буквы исходного текста x_i.

Подстановка Цезаря

Подстановка Цезаря является самым простым вариантом подстановки. Она относится к группе моноалфавитных подстановок. Этот шифр реализует следующее преобразование открытого текста: каждая буква открытого текста заменяется третьей после нее буквой в алфавите, который считается написанным по кругу, т.е. после буквы «я» следует буква «а». Отметим, что Цезарь заменял букву третьей после нее буквой, но можно заменять и какой-нибудь другой. Главное, чтобы тот, кому посылается сообщение, знал эту величину сдвига. Класс шифров, к которым относится шифр Цезаря, называется шифрами замены.

Подмножество $C_m = \{C_k: 0 \le k < m\}$ симметрической группы $SYM(Z_m)$, содержащее m подстановок: $C_k: j \to (j + k)\ (mod\ m)$, $0 \le k < m$, называется подстановкой Цезаря.

Умножение коммутативно, $C_k C_j = C_j C_k = C_{j+k}$, C_0 – идентичная подстановка, а обратной к C_k является $C_k^{-1} = C_{m-k}$, где $0 \le k < m$. Семейство подстановок Цезаря названо по имени римского императора Гая Юлия Цезаря, который поручал Марку Туллию Цицерону составлять послания с использованием 50-буквенного алфавита и подстановки C_3.

Подстановка определяется по таблице замещения, содержащей пары соответствующих букв «исходный текст – шифрованный текст». Для C_3 подстановки приведены ниже. Стрелка ($\to$) означает, что буква исходного

текста (слева) шифруется при помощи C_3 в букву шифрованного текста (справа).

Системой Цезаря называется моноалфавитная подстановка, преобразующая *n*-грамму исходного текста (x_0, x_1, ..., x_{n-1}) в *n*-грамму шифрованного текста (y_0, y_1, ..., y_{n-1}) в соответствии с правилом:

$$y_i = C_k(x_i),\ 0 \le i < n.$$

А→г	Й→м	Т→х	Ы→ю
Б→д	К→н	У→ц	Ь→я
В→е	Л→о	Ф→ч	Э→_
Г→ж	М→п	Х→ш	Ю→а
Д→з	Н→р	Ц→щ	Я→б
Е→и	О→с	Ч→ъ	_→в
Ж→й	П→т	Ш→ы	
З→к	Р→у	Щ→ь	
И→л	С→ф	Ъ→э	

A→D	J→M	S→V
B→E	K→N	T→W
C→F	L→O	U→X
D→G	M→P	V→Y
E→H	N→Q	W→Z
F→I	O→R	X→A
G→J	P→S	Y→B
H→K	Q→T	Z→C
I→L	R→U	

Например, известное послание Цезаря VENI VIDI VICI (в переводе на русский означает «Пришел, увидел, победил») выглядело бы в зашифрованном виде так: YHQL YLGL YLFL.

Достоинством системы шифрования Цезаря является простота шифрования и расшифровывания. К недостаткам системы Цезаря можно отнести следующее:

– подстановки, выполняемые в соответствии с системой Цезаря, не маскируют частот появления различных букв исходного открытого текста;

– сохраняется алфавитный порядок в последовательности заменяющих букв; при изменении значения *K* изменяются только начальные позиции такой последовательности;

– число возможных ключей *K* мало;

– шифр Цезаря легко вскрывается на основе анализа частот появления букв в шифротексте.

Криптоаналитическая атака против системы одноалфавитной замены начинается с подсчета частот появления символов: определяется число появлений каждой буквы в шифротексте. Затем полученное распределение частот букв в шифротексте сравнивается с распределением частот букв в алфавите исходных сообщений, например, в английском языке. Буква с наивысшей частотой появления в шифротексте заменяется на букву с наивысшей частотой появления в английском языке и т.д. Вероятность успешного вскрытия системы шифрования повышается с увеличением длины шифротекста. Вместе с тем идеи, заложенные в системе шифрования Цезаря, оказались весьма плодотворными, о чем свидетельствуют многочисленные модификации.

Аффинная система подстановок Цезаря

В системе шифрования Цезаря использовались только аддитивные свойства множества чисел. Однако символы можно также умножать по модулю m. Применяя одновременно операции сложения и умножения по модулю m над буквами алфавита, можно получить систему подстановок, которую называют аффинной системой подстановок Цезаря.

В данном преобразовании буква, соответствующая числу t, заменяется на букву, соответствующую числовому значению $(at + b)$ по модулю m. Следует заметить, что данное преобразование является взаимно однозначным отображением тогда и только тогда, когда НОД (a, m) – наибольший общий делитель чисел a и m равен единице, т.е. если a и m – взаимно простые числа.

Пример. Пусть $m = 26$, $a = 3$, $b = 5$. Тогда, очевидно, НОД (3, 26) = 1, и мы получаем следующее соответствие между числовыми кодами букв:

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
3t+5	5	8	11	14	17	20	23	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2

Преобразуя числа в буквы английского языка, получаем следующее соответствие для букв открытого текста и шифротекста:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	I	L	O	R	U	X	A	D	G	J	M	P	S	V	Y	B	E	H	K	N	Q	T	W	Z	C

Исходное сообщение HOPE преобразуется в шифротекст AVYR.

Достоинством аффинной системы является удобное управление ключами – ключи шифрования и дешифрования представляются в компактной форме в виде пары чисел (a, b). Недостатки аффинной системы

аналогичны недостаткам системы шифрования Цезаря. Аффинная система использовалась на практике несколько веков назад, а сегодня ее применение ограничивается большей частью иллюстрациями основных криптологических положений.

4.3. Шифрование методом гаммирования. Шифры сложной замены

4.3.1. Одноразовые системы

Шифры сложной замены называют многоалфавитными, так как для шифрования каждого символа исходного сообщения применяют свой шифр простой замены. Многоалфавитная подстановка последовательно и циклически меняет используемые алфавиты.

Многоалфавитная подстановка определяется ключом $\pi = (\pi_1, \pi_2, ...)$, содержащим не менее двух различных подстановок. В начале рассмотрим многоалфавитные системы подстановок с нулевым начальным смещением.

Пусть $\{K_i : 0 \le i < n\}$ – независимые случайные переменные с одинаковым распределением вероятностей, принимающие значения на множестве Z_m.

$$P\,кл\,\{(K_0, K_1, ..., K_{n-1}) = (K_0, K_1, ..., K_{n-1})\} = (1/m)^n$$

Система одноразового использования преобразует исходный текст:

$X = (X_0, X_1, ..., X_{n-1})$

в шифрованный текст

$Y = (Y_0, Y_1, ..., Y_{n-1})$

при помощи подстановки

$$Y_i = C_{k_i}(X_i) = (K_i + X_i)\ (mod\ m)\ (i = 0...n-1)$$

Для такой системы подстановки используют также термин «одноразовая лента» и «одноразовый блокнот». Пространство ключей K системы одноразовой подстановки является вектором рангов $(K_0, K_1, ..., K_{n-1})$ и содержит m^n точек.

Эффект использования многоалфавитной подстановки заключается в том, что обеспечивается маскировка естественной статистики исходного языка, так как конкретный символ из исходного алфавита A может быть преобразован в несколько различных символов шифровальных алфавитов B_j. Степень обеспечиваемой защиты теоретически пропорциональна длине периода r в последовательности используемых алфавитов B_j.

В классическом понимании одноразовый блокнот является большой неповторяющейся последовательностью символов ключа, распределенных случайным образом, написанных на кусочках бумаги и приклеенных к листу блокнота. Он был изобретен в 1917 году. Первоначально это была одноразовая лента для телетайпов. Отправитель использовал каждый символ ключа блокнота для шифрования только одного символа открытого текста. Шифрование представляет собой сложение по модулю 26 (для английского алфавита) символа открытого текста и символа ключа из одноразового блокнота.

Каждый символ ключа используется только единожды и для единственного сообщения. Отправитель шифрует сообщения и уничтожает использованные страницы блокнота или использованную часть ленты. Получатель, в свою очередь, используя точно такой же блокнот, дешифрирует каждый символ шифротекста. Расшифровав сообщение, получатель уничтожает соответствующие страницы блокнота или часть ленты. Новое сообщение – новые символы ключа. Например, если сообщением является:

ONE

а ключевая последовательность в блокноте:

TOO

то шифротекст будет выглядеть как:

HBS

O + T mod 26 = H;
N + O mod 26 = B;
E + O mod 26 = S.

Таким образом, одноразовая система шифрует исходный открытый текст $X = (X_0, X_1, ..., X_{n-1})$ в шифротекст $Y = (Y_0, Y_1, \ldots, Y_{n-1})$ посредством подстановки Цезаря $Y_i = (X_i + K_i) \bmod m$, $0 \le i < n$, где K_i – i-тый элемент случайной ключевой последовательности.

Процедура расшифровывания описывается соотношением:

$$X_i = (Y_i - K_i) \bmod m,$$

где K_i – i-тый элемент той же самой случайной ключевой последовательности.

В предположении, что злоумышленник не сможет получить доступ к одноразовому блокноту, использованному для шифрования сообщения, эта схема совершенно безопасна. Данное шифрованное сообщение на вид соответствует любому открытому сообщению того же размера. Так как все ключевые последовательности совершенно одинаковы (помните, символы ключа генерируются случайным образом), у противника отсутствует информация, позволяющая подвергнуть шифротекст криптоанализу. Кусочек

шифротекста может быть похож на: POY, что дешифрируется как: SAL или на: BXE, что дешифрируется как: GRE.

Повторим еще раз: так как все открытые тексты равновероятны, у криптоаналитика нет возможности определить, какой из открытых текстов является правильным. Случайная ключевая последовательность, сложенная с неслучайным открытым текстом, даст совершенно случайный шифротекст, и никакие вычислительные мощности не смогут это изменить.

Необходимо напомнить, что символы ключа должны генерироваться случайным образом. Любые попытки вскрыть такую схему сталкиваются со способом, которым создастся последовательность символов ключа. Использование генераторов псевдослучайных чисел не считается, у них всегда неслучайные свойства. Если вы используете действительно случайный источник – это намного труднее, чем кажется на первый взгляд – это совершенно безопасно.

Другой важный момент: ключевую последовательность никогда нельзя использовать второй раз. Даже если вы используете блокнот размером в несколько гигабайт, то если криптоаналитик получит несколько текстов с перекрывающимися ключами, он сможет восстановить открытый текст. Он сдвинет каждую пару шифротекстов относительно друг друга и подсчитает число совпадений в каждой позиции. Если шифротексты смещены правильно, соотношение совпадений резко возрастет – точное значение зависит от языка открытого текста. С этой точки зрения криптоанализ не представляет труда. Это похоже на показатель совпадений, но сравниваются два различных периода. Не используйте ключевую последовательность повторно.

Идея одноразового блокнота легко расширяется на двоичные данные. Вместо одноразового блокнота, состоящего из букв, используется одноразовый блокнот из битов. Вместо сложения открытого текста с ключом одноразового блокнота используйте XOR (операцию сложения по mod 2: $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$). Для дешифрирования примените XOR к шифротексту с тем же одноразовым блокнотом. Все остальное не меняется, и безопасность остается такой же совершенной.

Существует несколько проблем. Так как ключевые биты должны быть случайными и не могут использоваться снова, длина ключевой последовательности должна равняться длине сообщения. Одноразовый блокнот удобен для нескольких небольших сообщений, но его нельзя использовать для работы по каналу связи с пропускной способностью 1,544 Мбит/с. Вы можете хранить 650 Мбайт случайных данных на CD-ROM, но и тут есть проблемы. Во-первых, вам нужно только две копии случайных битов, но CD-ROM экономичны только при больших тиражах. И, во-вторых, вам

нужно уничтожать использованные биты. Для CD-ROM нет другой возможности удалить информацию, кроме как физически разрушить весь диск. Гораздо больше подходит цифровая лента.

Даже если проблемы распределения и хранения ключей решены, вам придется точно синхронизировать работу отправителя и получателя. Если получатель пропустит бит (или несколько бит пропадут при передаче), сообщение потеряет всякий смысл. С другой стороны, если несколько бит изменятся при передаче (и ни один бит не будет удален или добавлен – что гораздо больше похоже на влияние случайного шума), то лишь эти биты будут расшифрованы неправильно. Но одноразовый блокнот не обеспечивает проверку подлинности.

Одноразовые блокноты используются и сегодня, главным образом для сверхсекретных каналов связи с низкой пропускной способностью. По слухам «горячая линия» между Соединенными Штатами и бывшим Советским Союзом шифруется с помощью одноразового блокнота. Многие сообщения шпионов зашифрованы с использованием одноразовых блокнотов. Эти сообщения нераскрыты сегодня и навсегда останутся нераскрытыми. На этот факт не повлияет время работы суперкомпьютеров над этой проблемой. Наложение белого шума в виде бесконечного ключа на исходный текст меняет статистические характеристики языка источника. Системы одноразового использования теоретически не расшифруемы, так как не содержат достаточной информации для восстановления текста.

Почему же эти системы неприменимы для обеспечения секретности при обработке информации? Ответ простой – они непрактичны, так как требуют независимого выбора значения ключа для каждой буквы исходного текста.

4.3.2. Система шифрования Вижинера

Начнем с конечной последовательности ключа:

$$K = (K_1, ..., K_n),$$

которая называется ключом пользователя, и продлим ее до бесконечной последовательности, повторяя цепочку. Таким образом, получим рабочий ключ:

$$K = (K_0, K_1, ..., K_n), K_j = K_{j \bmod r} \ 0 \le j < \infty.$$

Например, при $r = \infty$ и ключе пользователя 15 8 2 10 11 4 18 рабочий ключ будет периодической последовательностью:

15 8 2 10 11 4 18 15 8 2 10 11 4 18 15 8 2 10 11 4 18 ...

Подстановка Вижинера VIG_k определяется как:

$$VIG_k : (X_0, X_1, ..., X_{n-1}) \to (Y_0, Y_1, ..., Y_{n-1}) = (X_0 + k, X_1 + k, ..., X_{n-1} + k).$$

Таким образом:

1) исходный текст X делится на *r фрагментов*

$$X_i = (X_i, X_{i+r}, ..., X_{i+r(n-1)}),\ 0 \le i < r;$$

2) *i*-тый фрагмент исходного текста X_i шифруется при помощи подстановки Цезаря

$$C_k : (X_i, X_{i+r}, ..., X_{i+r(n-1)}) \to (Y_i, Y_{i+r}, ..., Y_{i+r(n-1)}).$$

Вариант системы подстановок Вижинера при $m = 2$ называется системой Вернама (1917 г).

В то время ключ $K = (k_0, k_1, ..., k_{n-1})$ записывался на бумажной ленте. Каждая буква исходного текста в алфавите, расширенном некоторыми дополнительными знаками, сначала переводилась с использованием кода Бодо в пятибитовый символ. К исходному тексту Бодо добавлялся ключ (по mod2). Старинный телетайп фирмы AT&T со считывающим устройством Вернама и оборудованием для шифрования, использовался корпусом связи армии США.

Очень распространена плохая, с точки зрения секретности, практика использовать слово или фразу в качестве ключа для того, чтобы $K = (K_0, K_1, ..., K_{n-1})$ было легко запомнить. В ИС для обеспечения безопасности информации это недопустимо. Для получения ключей должны использоваться программные или аппаратные средства случайной генерации ключей.

Таким образом, система Вижинера подобна такой системе шифрования Цезаря, у которой ключ подстановки меняется от буквы к букве. Этот шифр многоалфавитной замены можно описать таблицей шифрования, называемой таблицей (квадратом) Вижинера. На рис. 4.6 и 4.7 показаны таблицы Вижинера для русского и английского алфавитов соответственно. Таблица Вижинера используется как для зашифровывания, так и для расшифровывания. Она имеет два входа:

– верхнюю строку подчеркнутых символов, используемую для считывания очередной буквы исходного открытого текста;

– крайний левый столбец ключа.

Последовательность ключей обычно получают из числовых значений букв ключевого слова. При шифровании исходного сообщения его выписывают в строку, а под ним записывают ключевое слово (или фразу). Если ключ оказался короче сообщения, то его циклически повторяют. В процессе шифрования находят в верхней строке таблицы очередную букву исходного текста и в левом столбце очередное значение ключа. Очередная буква шифротекста находится на пересечении столбца, определяемого шифруемой буквой, и строки, определяемой числовым значением ключа.

Рассмотрим пример получения шифротекста с помощью таблицы Вижинера. Пусть выбрано ключевое слово АМБРОЗИЯ. Необходимо зашифровать сообщение ПРИЛЕТАЮ СЕДЬМОГО. Выпишем исходное сообщение в строку и запишем под ним ключевое слово с повторением. В третью строку будем выписывать буквы шифротекста, определяемые из таблицы Вижинера:

Сообщение	П	Р	И	Л	Е	Т	А	Ю	С	Е	Д	Ь	М	О	Г	О
Ключ	А	М	Б	Р	О	З	И	Я	А	М	Б	Р	О	З	И	Я
Шифротекст	П	Ъ	Й	Ы	У	Щ	И	Э	С	С	Е	К	Ь	Х	Л	Н

Ключ	**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**	**ь**	**ы**	**ъ**	**э**	**ю**	**я**
0	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	ь	ы	ъ	э	ю	я
1	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	ь	ы	ъ	э	ю	я	а
2	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б
3	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в
4	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г
5	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д
6	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е
7	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж
8	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з
9	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и
10	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й
11	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к
12	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л
13	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м
14	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н
15	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о
16	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
17	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р
18	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с
19	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т
20	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у
21	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф
22	ц	ч	ш	щ	ь	ы	ъ	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
23	ч	ш	щ	ь	ы	ъ	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
24	ш	щ	ь	ы	ъ	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
25	щ	ь	ы	ъ	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
26	ь	ы	ъ	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
27	ы	ъ	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	ь
28	ъ	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	ь	ы
29	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	ь	ы	ъ
30	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	ь	ы	ъ	э
31	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	ь	ы	ъ	э	ю

Рис. 4.6. Таблица Вижинера для русского алфавита

Ключ	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
2	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
3	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
4	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
5	f	g	h	I	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
6	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
7	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
8	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
9	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
10	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
11	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
12	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
13	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
14	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
15	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
16	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
17	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
18	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
19	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
20	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
21	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
22	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
23	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
24	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
25	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

Рис. 4.7. Таблица Вижинера для английского алфавита

4.3.3. Шифрование методом гаммирования

Гаммирование является также широко применяемым криптографическим преобразованием. На самом деле граница между гаммированием и использованием бесконечных ключей и шифров Вижинера, о которых речь шла выше, весьма условная.

Гамма шифра – это псевдослучайная последовательность, выработанная по заданному алгоритму для зашифровывания открытых данных и расшифровывания принятых данных.

Процесс зашифровывания заключается в генерации гаммы шифра и наложении полученной гаммы на исходный открытый текст обратимым образом, например, с использованием операции сложения по модулю 2.

Следует отметить, что перед зашифровыванием открытые данные разбивают на блоки $T_0^{(i)}$ одинаковой длины, обычно по 64 бита. Гамма

шифра вырабатывается в виде последовательности блоков $Г_ш^{(i)}$ аналогичной длины. Уравнение шифрования можно записать в виде $T_ш^{(i)} = Г_ш^{(i)} \oplus T_0^{(i)}$, $i = 1 \ \ldots \ M$, где $T_ш^{(i)}$ – i-тый блок шифротекста; $Г_ш^{(i)}$ – i-тый блок гаммы шифра; $T_0^{(i)}$ – i-тый блок открытого текста; M – количество блоков открытого текста.

Процесс расшифровывания сводится к повторной генерации гаммы шифра и наложению этой гаммы на принятые данные. Уравнение расшифровывания имеет вид $T_0^{(i)} = Г_ш^{(i)} \oplus T_ш^{(i)}$.

Получаемый этим методом шифротекст достаточно труден для раскрытия, поскольку теперь ключ является переменным. По сути дела гамма шифра должна изменяться случайным образом для каждого шифруемого блока. Если период гаммы превышает длину всего шифруемого текста и злоумышленнику неизвестна никакая часть исходного текста, то такой шифр можно раскрыть только прямым перебором всех вариантов ключа. В этом случае криптостойкость шифра определяется длиной ключа.

Ниже рассматриваются наиболее распространенные методы генерации гамм, которые могут быть использованы на практике.

Чтобы получить линейные последовательности элементов гаммы, длина которых превышает размер шифруемых данных, используются датчики псевдослучайных последовательностей (ПСП). На основе теории групп было разработано несколько типов таких датчиков.

В настоящее время наиболее доступными и эффективными являются конгруэнтные генераторы ПСП. Для этого класса генераторов можно сделать математически строгое заключение о том, какими свойствами обладают выходные сигналы этих генераторов с точки зрения периодичности и случайности. Рассмотрим, к примеру, генератор ПСП.

Одним из хороших конгруэнтных генераторов является линейный конгруэнтный датчик ПСП. Он вырабатывает последовательности псевдослучайных чисел $X(i)$, описываемые соотношением:

$$X_{(i+1)} \ = \ (A \cdot X_{(i)} + C) \ \mathrm{mod} \ m,$$

где A и C – константы, $X(0)$ – исходная величина, выбранная в качестве порождающего числа. Очевидно, что эти три величины и образуют ключ.

Такой датчик ПСП генерирует псевдослучайные числа с определенным периодом повторения, зависящим от выбранных значений A и C. Значение m обычно устанавливается равным $2n$, где n – длина машинного слова в битах. Датчик имеет максимальный период M до того, как генерируе-

мая последовательность начнет повторяться. По причине, отмеченной ранее, необходимо выбирать числа A и C такие, чтобы период M был максимальным. Как показано Д. Кнутом, линейный конгруэнтный датчик ПСП имеет максимальную длину M тогда и только тогда, когда C – нечетное, и $A = 1 \bmod 4$.

Генератор ПСП $X_t \in A = \{0, 1, ..., N-1\}$ может быть построен также на основе квадратичного рекуррентного соотношения:

$$X_{t+1} = \left(dX_t^2 + aX_t + c\right) \bmod N, \qquad t = 0,1,...$$

где $X_0, a, c, d \in A$ – параметры генератора.

Генератор имеет максимальный период, $T_{\max} = N$, когда:

1) c, N – взаимно простые числа;

2) $d, a-1$ – кратны p, p – любой нечетный простой делитель N;

3) d – четное число, причем:

$$d = \begin{cases} (a-1) \bmod 4, & \textit{если } N \textit{ кратно } 4; \\ (a-1) \bmod 4, & \textit{если } N \textit{ кратно } 2. \end{cases}$$

4) если N кратно 9, то либо $d \bmod 9 = 0$, либо $d \bmod 9 = 1$ и $cd \bmod 9 = 6$.

Генератор Эйхенауэра – Лена с обращением. Нелинейная ПСП Эйхенауэра – Лена с обращением определяется рекурсивным соотношением ($t = 0,1,...$):

$$X_{t+1} = \begin{cases} \left(aX_t^{-1} + c\right) \bmod N, & \textit{если } X_t \geq 1; \\ c, & \textit{если } X_t = 0. \end{cases}$$

где $X_t^{-1} \in A -$ обратный к X_t по mod$N$, т.е. $X_t^{-1} X_t = 1 \bmod N$;

$X_0, a, c \in A$ – параметры генератора.

Если $N = 2^q$, a, X_0 – нечетны, c – четно, то генератор имеет максимально возможный период $T_{\max} = 2^{q-1}$, когда:

$a = 1 \bmod, \qquad 4c = 2 \bmod 4.$

Для шифрования данных с помощью датчика ПСП может быть выбран ключ любого размера. Например, пусть ключ состоит из набора чисел $X(j)$ размерностью b, где $j = 1, 2, ..., n$. Тогда создаваемую гамму шифра G можно представить как объединение непересекающихся множеств $H(j)$.

Шифрование с помощью датчика ПСЧ является довольно распространенным криптографическим методом. Во многом качество шифра, по-

строенного на основе датчика ПСП, определяется не только и не столько характеристиками датчика, сколько алгоритмом получения гаммы. Один из фундаментальных принципов криптологической практики гласит, даже сложные шифры могут быть очень чувствительны к простым воздействиям.

4.4. Вопросы и задания для самопроверки

1. Какими способами можно обеспечить защиту передаваемой информации от несанкционированного доступа?

2. В чем различны криптография и криптоанализ?

3. Что называют шифрованием (расшифрованием)?

4. Что такое цифровая подпись?

5. Что такое криптостойкость?

6. Какую задачу решает имитостойкость?

7. Какие существуют модели каналов с криптографическим кодированием информации?

8. Какие требования предъявляются к современным криптосистемам?

9. Что является ключом в шифрующих таблицах?

10. Что является ключом в шифрующих таблицах при двойной перестановке?

11. В чем различия между аддитивной и аффинной системой Цезаря?

12. Какой основной недостаток систем шифрования Цезаря?

13. К какому типу шифров относится система шифрования Вижинера?

14. Какие достоинства имеет система Вижинера по отношению к системам Цезаря?

15. Какая криптосистема обеспечивает максимальную защиту информации?

16. Почему для большинства современных задач не рационально использование одноразовых шифров?

17. Что такое гамма шифра?

18. Чем ограничивается длина гаммы шифра?

19. На основе какого выражения можно сформировать псевдослучайную последовательность чисел?

20. В чем принципиальная разница между шифрованием методом гаммирования и одноразовой системой шифрования?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5
Криптографическое кодирование информации с использованием шифрующих таблиц и системы шифрования Цезаря

Теория для практического занятия представлена в разделе 4.2.

Перед выполнением тестовых заданий проводится опрос с использованием вопросов, представленных в разделе 4.4.

Тестовые задания:

1. В качестве ключа в шифрующих таблицах используют:

a) размер таблицы;

b) слово или фразу, задающие перестановки ;

c) особенности структуры таблицы;

d) ключ не используется.

2. К достоинствам системы шифрования Цезаря относятся:

a) большое число возможных ключей;

b) сохранение алфавитного порядка в последовательности заменяющих букв;

c) простота шифрования;

d) простота расшифрования.

3. Результат зашифровывания сообщения КОМПЬЮТЕР методом простой перестановки, размер таблицы 3×3 выглядит:

a) КПТОЬЕМЮР;

b) КТПОЕЬМРЮ;

c) ПКТЬОЕЮМР;

d) МЮРКПТОЬЕ.

4. Результат зашифровывания сообщения МЕСТО ВСТРЕЧИ МОСТ методом двойной перестановки (ключ для столбцов 2143, ключ для строк 4132) выглядит:

a) ОВСТМОСТРЕЧИМЕСТ;

b) ОМРМВОЕЕССЧСТТИТ;

c) ЕМТСВОТСЕРИЧОМТС;

d) ВОТСОМТСЕРИЧЕМТС.

5. Результат зашифровывания сообщения ПРИЕДУ ДЕСЯТОГО методом Цезаря ($K = 5$):

a) ФХНКЙШЙКЦДЧИУ;

b) УЧДЦКЙШЙКНЧФ;

c) ЙКЦДЧИУФХНКЙШ;

d) ШЙКНЧФУЧДЦКЙ.

Задачи

1. Зашифруйте методом простой перестановки (размер таблицы 4×4) следующие сообщения:

a) КОМПЬЮТЕРНЫЙ ДИСК;

b) ПОГОДА БУДЕТ ЯСНАЯ;

c) МОБИЛЬНЫЙ ТЕЛЕФОН;

d) ЭЛЕКТРИЧЕСТВА НЕТ.

2. Зашифруйте методом двойной перестановки (ключ для столбцов 4321, ключ для строк 2143) следующие сообщения:

a) ЗВУКОВОЕ ДАВЛЕНИЕ;

b) КАТУШКА ЗАЖИГАНИЯ;

c) КОСМИЧЕСКАЯ СВЯЗЬ;

d) МИКРОЭЛЕКТРОНИКА.

3. Зашифруйте методом Цезаря ($K = 7$) следующие сообщения:

a) ОРГАНИЗАЦИЯ ДОРОЖНОГО ДВИЖЕНИЯ;

b) КАСКАД УСИЛЕНИЯ;

c) ДЫРОЧНАЯ ПРОВОДИМОСТЬ;

d) ГАЛЬВАНИЧЕСКИЙ ЭЛЕМЕНТ.

4. Используя подстановку Цезаря (C_4), зашифруйте следующее сообщение:

КОЛИЧЕСТВО_ИНФОРМАЦИИ_В_ДИСКРЕТНОМ_СООБЩЕНИИ.

5. Используя подстановку Цезаря (C_5), определите исходный текст:

ЦУМЙЕТНКДУЧПХ_ЧУИУДПРВЬЕДНМДМЕПХ_ЧУИУ

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 6
Криптографическое кодирование информации с использованием аффинной системы подстановок Цезаря и алгоритма Вижинера

Перед выполнением тестовых заданий проводится опрос с использованием вопросов, представленных в разделе 4.4.

Теория для практического занятия представлена в разделе 4.3.

Тестовые задания:

1. Какие математические операции применяют в аффинной системе Цезаря?

a) сложения и вычитания;

b) сложения и умножения;

c) умножения и деления;

d) вычитания и деления.

2. К какой системе шифрования относится система Вижинера?

a) гаммирование;

b) одноалфавитные;

c) многоалфавитные;

d) шифры простой замены.

3. К какой системе шифрования относится аффинная система Цезаря?

a) гаммирование;

b) одноалфавитные;

c) многоалфавитные;

d) шифры простой замены.

4. Результат зашифровывания сообщения ОШИБКА НАЙДЕНА аффинной системой подстановок Цезаря ($m = 32$, $a = 7$, $b = 4$):

a) ЖМЪЛКДЯДГАЗЯД;

b) РГНУФДЖМАЙФОГ;

c) КУСШТКЫЭЪОЫК;

d) МТРПКЫДЦГМЫЙЕ.

5. Результат зашифровывания сообщения ДОЖДЯ НЕ БУДЕТ методом Вижинера (Ключ: КАМЕНЬ):

a) ГВВКЦТТРПФВШ;

b) ВУЛРЫФЩКЙЦРФ;

c) ООТЙМЗПБЯЙТМ;

d) ЛГНУЫВЖХЧМТА.

Задачи

1. Зашифруйте аффинной системой подстановок Цезаря ($m = 32$, $a = 13$, $b = 7$) следующие сообщения:

a) ПРЕХОДНАЯ ФУНКЦИЯ;

b) ТЕЛЕВИЗИОННЫЙ СТАНДАРТ;

c) РАМОЧНАЯ АНТЕННА;

d) ГИЛЬБЕРТОВО ПРОСТРАНСТВО.

2. Зашифруйте методом Вижинера (Ключ: ГИЛЬЗА) следующие сообщения:

a) ТЕОРИЯ ВЕРОЯТНОСТЕЙ;

b) БАКТЕРИЦИДНАЯ ЛАМПА;

c) ДАЛЬНЯЯ СВЯЗЬ;

d) КВАРЦЕВЫЙ РЕЗОНАТОР.

3. Зашифровать при помощи подстановки Вижинера сообщение: ЛИЧНОЕСООБЩЕНИЕДЛЯВСЕХ. Ключ: Текст

4. Расшифровать зашифрованное при помощи подстановки Вижинера сообщение: ЭКБМЪШХШСБКТКАШЦБНЕЬМПИГЛЩТЫ Ключ: Рефлекс.

5. Расшифровать зашифрованное при помощи подстановки Вижинера сообщение: ШТКЛХТЙВДЗЧТМПЧПШИСВЫПМ_ЦРРВЛОЙВЧМПЯ.

Ключ: Звезда

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 7
Криптографическое кодирование информации одноразовым шифром и гаммированием

Перед выполнением тестовых заданий проводится опрос с использованием вопросов, представленных в разделе 4.4.

Теория для практического занятия представлена в разделе 4.3.

Тестовые задания:

1. Какой размер должна иметь ключевая последовательность одноразовой системы шифрования?

a) длина ключа должна быть равна длине шифруемого сообщения;

b) длина ключа определяется пользователем;

c) длина ключа должна быть не менее 1/2 длины шифруемого сообщения;

d) длина ключа должна быть менее 1/2 длины шифруемого сообщения.

2. Гамма шифра используется для:

a) шифрования ключа;

b) расшифрования ключа;

c) шифрования сообщения;

d) расшифрования сообщения;

3. Возможности широкого использования одноразовой системы ограничены:

a) практическими аспектами реализации системы;

b) недостаточной криптостойкостью;

c) недостаточной имитостойкостью;

d) небольшим пространством возможных ключей.

4. Процедура расшифровывания для одноразовой системы описывается соотношением:

a) $X_{(i+1)} = (A \cdot X_{(i)} + C) \mod m$;

b) $C_k: j \to (j + k) \pmod m$;

c) $X_i = (Y_i - K_i) \bmod m$;

d) $X_{(i+1)} = (A \cdot X_{(i)} - C) \bmod m$.

5. Псевдослучайная последовательность чисел может быть сформирована на основе выражения:

a) $X_{(i+1)} = (A \cdot X_{(i)} + C) \bmod m$;

b) C_k: $j \to (j + k) \pmod m$;

c) $X_i = (Y_i - K_i) \bmod m$;

d) $X_{(i+1)} = (A \cdot X_{(i)} - C) \bmod m$.

Задачи

1. Используя одноразовый блокнот, зашифруйте сообщение: ВЗЛОМАТЬ СИСТЕМУ.

Ключ: Блуждающие_ключи.

2. Шифротекст выглядит так: ПНЮООЭВИШИУЯЮЩИРЬРЫКЛ. Определите исходный текст, используя одноразовый блокнот.

Ключ: Заканчивается_строка..

3. Шифротекст выглядит так: ДЧЦЩЪЯОИТЕТКМУБЙВ. Определите исходный текст, используя одноразовый блокнот.

Ключ: Мобильный_телефон

4. Зашифровать при помощи гаммирования сообщение, используя датчики ПСП: Ключ РАДИОТЕХНИК(T(0) = 0, A = 5, C = 3).

5. Зашифровать при помощи гаммирования сообщение, используя датчики ПСП: ПРОГРАММИСТГЕНИЙ (T(0) = 0, A = 5, C = 9).

6. Расшифровать зашифрованное при помощи гаммирования сообщение 0100100011000000010011110101010010000111100010000110110100 0100100101101101111100001100000000011001, используя датчики ПСП: (T(0) = 0, A = 5, C = 3).

7. Расшифровать зашифрованное при помощи гаммирования сообщение 010010001100000001001111010101001000011110001000011011, используя датчики ПСП: (T(0) = 0, A = 9, C = 3).

МОДУЛЬ 5. СОВРЕМЕННЫЕ СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

1. Американский стандарт шифрования DES.

2. Реализация функции шифрования в алгоритме DES. Алгоритм вычисления ключей.

3. Основные режимы работы алгоритма DES. Комбинирование блочных алгоритмов.

4. Логика построения шифра, структура ключевой информации стандарта ГОСТ 28147-89 и основной шаг криптопреобразования стандарта ГОСТ 28147-89.

5. Базовые циклы, основные режимы шифрования алгоритма ГОСТ 28147-89.

6. Вопросы и задания для самопроверки.

7. Практическое занятие № 8.

Цель модуля – изучение студентами симметричных криптосистем DES и ГОСТ 28147-89.

В результате изучения модуля студенты должны:

– знать процедуры зашифровывания и расшифровывания информации симметричными алгоритмами DES и ГОСТ 28147-89;

– знать достоинства и недостатки симметричных криптосистем;

– уметь использовать криптосистемы DES и ГОСТ 28147-89 для шифрования информации;

– иметь представление о размере ключа для надежного шифрования информации на современном уровне развития техники.

5.1. Американский стандарт шифрования DES

Стандарт шифрования данных DES (Data Encryption Standard), который ANSI называет Алгоритмом шифрования данных DEA (Data Encryption Algorithm), а ISO – DEA-1, за 20 лет стал мировым стандартом.

5.1.1. Разработка стандарта

В начале 70-х годов невоенные криптографические исследования были крайне редки. В этой области почти не публиковалось исследовательских работ. Большинство людей знали, что для своих коммуникаций военные используют специальную аппаратуру кодирования, но мало кто разбирался в криптографии как в науке. Заметными знаниями обладало Агентство национальной безопасности (National Security Agency, NSA), но оно даже не признавало публично своего собственного существования.

В 1972 году Национальное бюро стандартов (National Bureau of Standards, NBS), теперь называющееся Национальным институтом стандартов и техники (National Institute of Standards and Technology, NIST), выступило инициатором программы защиты линий связи и компьютерных данных. Одной из целей этой программы была разработка единого, стандартного криптографического алгоритма. Этот алгоритм мог бы быть проверен и сертифицирован, а использующие его различные криптографические устройства могли бы взаимодействовать. Он мог бы, к тому же, быть относительно недорогим и легкодоступным.

15 мая 1973 года в Federal Register NBS опубликовало требования к криптографическому алгоритму, который мог бы быть принят в качестве стандарта. В IBM существовала целая команда криптографов, работавшая в Кингстоне (Kingston) и Йорктаун Хайте (Yorktown Heights), в которую входили Рой Адлер (Roy Adler), Дон Копперсмит (Don Coppersmith), Хорст Фейстель (Horst Feistel), Эдна Кроссман (Edna Crossman), Алан Конхейм (Alan Konheim), Карл Майер (Carl Meyer), Билл Ноц (Bill Notz), Линн Смит (Lynn Smith), Уолт Тачмен (Walt Tuchman) и Брайант Такерман (Bryant Tuckerman).

Наконец, 17 марта 1975 года в Federal Register NBS опубликовало и подробности алгоритма, и заявление IBM о предоставлении неисключительной, бесплатной лицензии на алгоритм, а также предложило присылать комментарии по поводу данного алгоритма. В другой заметке в Federal Register, 1 августа 1975 года, различным организациям и широкой публике снова предлагалось прокомментировать предложенный алгоритм. Многие настороженно относились к участию «невидимой руки» NSA в разработке алгоритма. Боялись, что NSA изменит алгоритм, вставив в него потайную дверцу. Жаловались, что NSA уменьшило длину ключей с первоначальных 128 битов до 56. Жаловались на внутренние режимы работы алгоритма. Многие соображения NSA стали ясны и понятны в начале 90-х, но в 70-х они казались таинственными и тревожными.

Несмотря на критику, Стандарт шифрования данных DES 23 ноября 1976 года был принят в качестве федерального стандарта и разрешен к использованию на всех несекретных правительственных коммуникациях. Официальное описание стандарта, FIPS PUB 46, «Data Encryption Standard», было опубликовано 15 января 1977 года и вступило в действие шестью месяцами позже. FIPS PUB 81, «Modes of DES Operation» (Режимы работы DES), было опубликовано в 1980 году. FIPS PUB 74, «Guidelines for Implementing and Using the NBS Data Encryption Standard» (Руководство по реализации и использованию Стандарта шифрования данных NBS), появилось в 1981 году. NBS также опубликовало FIPS PUB 112, специфицируя

DES для шифрования паролей, и FIPS PUB 113, специфицируя DES для проверки подлинности компьютерных данных. (FIPS обозначает Federal Information Processing Standard.)

Эти стандарты были беспрецедентными. Никогда до этого оцененный NSA алгоритм не был опубликован. Возможно, эта публикация была следствием непонимания, возникшего между NSA и NBS. NSA считало, что DES будет реализовываться только аппаратно. В стандарте требовалась именно аппаратная реализация, но NBS опубликовало достаточно информации, чтобы можно было создать и программную реализацию DES. Не для печати NSA охарактеризовало DES как одну из своих самых больших ошибок. Если бы Агентство предполагало, что раскрытые детали позволят писать программное обеспечение, оно никогда бы не согласилось на это. Для оживления криптоанализа DES сделал больше, чем что-либо другое. Теперь для исследования был доступен алгоритм, который NSA объявило безопасным. Не случайно следующий правительственный стандарт алгоритма, Skipjack, был засекречен.

Американский национальный институт стандартов (American National Standards Institute, ANSI) одобрил DES в качестве стандарта для частного сектора в 1981 году (ANSI X3.92.), назвав его Алгоритмом шифрования данных (Data Encryption Algorithm, DEA). ANSI опубликовал стандарт режимов работы DEA (ANSI X3.106), похожий на документ NBS, и стандарт для шифрования в сети, использующий DES (ANSI X3.105).

ISO сначала проголосовала за введение DES, называемого в ее интерпретации DEA-1, в качестве международного стандарта, а затем приняла решение не заниматься стандартизацией криптографии. Однако в 1987 году группа ISO, занимающаяся международными стандартами в области оптовой торговли, применила DES в международном стандарте проверки подлинности и для управления ключами. DES также используется в качестве австралийского банковского стандарта.

В стандарте DES было оговорено, что он будет пересматриваться каждые пять лет. В 1983 году DES был повторно сертифицирован без всяких проблем. 6 марта 1987 года в Federal Register NBS попросило прокомментировать предложение на следующие пять лет. NBS предложило на обсуждение следующие три альтернативы: вновь подтвердить стандарт на следующие пять лет, отказаться от стандарта или пересмотреть применимость стандарта. После длительных споров DES был вновь утвержден в качестве правительственного стандарта США до 1992 года. В 1992 году альтернативы алгоритму DES все еще не было, и он был утвержден еще на 5 лет.

5.1.2. Обобщенная схема зашифровывания

Алгоритм DES использует комбинацию подстановок и перестановок. DES осуществляет зашифровывание 64-битовых блоков данных с помощью 64-битового ключа, в котором значащими являются 56 бит (остальные 8 бит - проверочные биты для контроля на четность). Расшифровывание в DES является операцией, обратной шифрованию, и выполняется путем повторения операций зашифровывания в обратной последовательности.

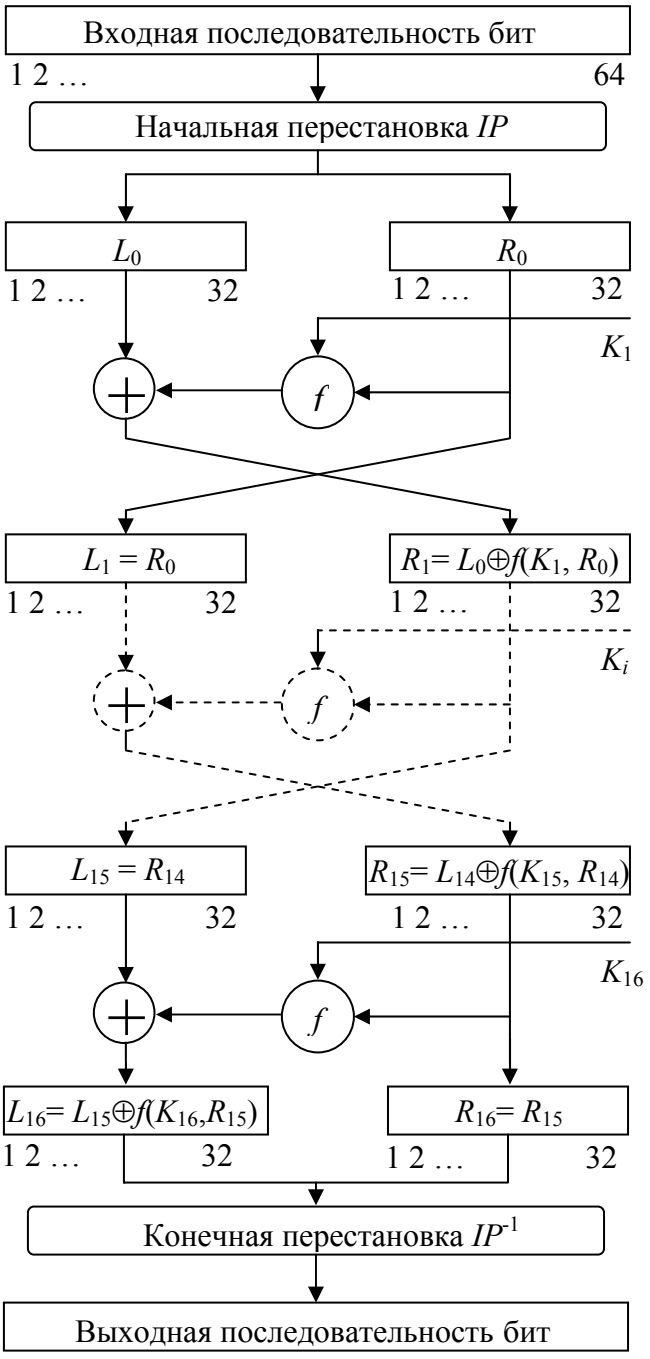


Рис. 5.1. Обобщенная схема зашифровывания в алгоритме DES

Обобщенная схема процесса шифрования в алгоритме DES (рис. 5.1) заключается в начальной перестановке бит 64-битового блока, шестнадцати циклах шифрования и, наконец, в конечной перестановке бит.

Следует отметить, что все приводимые таблицы являются стандартными и должны включаться в реализацию алгоритма DES в неизменном виде.

Все перестановки и коды в таблицах подобраны разработчиками таким образом, чтобы максимально затруднить процесс взлома шифра.

Пусть из файла исходного текста считан очередной 64-битовый блок T_0. Этот блок преобразуется с помощью матрицы начальной перестановки *IP* (табл. 5.1).

Таблица 5.1

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

Биты входного блока T_0 (64 бита) переставляются в соответствии с матрицей IP: бит 58 входного блока T_0 становится битом 1, бит 50 – битом 2 и т.д. Эту перестановку можно описать выражением $T_0 = IP(T)$. Полученная последовательность бит T_0 разделяется на две последовательности: L_0 – левые, или старшие, биты, R_0 – правые, или младшие, биты – каждая из которых содержит 32 бита.

Затем выполняется итеративный процесс шифрования, состоящий из 16 циклов. Пусть T_i – результат i-той итерации: $T_i = L_i R_i$, где $L_i = t_1 t_2 \dots t_{32}$ (первые 32 бита); $R_i = t_{33} t_{34} \dots t_{64}$ (последние 32 бита). Тогда результат i-той итерации описывается следующими формулами:

$$L_i = R_{i-1},\ i = 1,2,\dots,16;$$
$$R_i = L_{i-1} \oplus f\left(R_{i-1}, K_i\right),\ i = 1,2,\dots,16.$$

Функция f называется функцией шифрования. Ее аргументами являются последовательность R_{i-1}, получаемая на предыдущем шаге итерации, и 48-битовый ключ K_i, который является результатом преобразования 64-битового ключа шифра K. (Подробнее функция шифрования f и алгоритм получения ключа K описаны ниже.)

На последнем шаге итерации получают последовательности R_{16} и L_{16} (без перестановки местами), которые конкатенируются в 64-битовую последовательность $R_{16}L_{16}$.

По окончании шифрования осуществляется восстановление позиций бит с помощью матрицы обратной перестановки IP^{-1} (табл. 5.2).

Таблица 5.2

Матрица обратной перестановки

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

Процесс расшифровывания данных является инверсным по отношению к процессу шифрования. Все действия должны быть выполнены в обратном порядке. Это означает, что расшифровываемые данные сначала переставляются в соответствии с матрицей IP^{-1}, а затем над последовательностью бит $R_{16}L_{16}$ выполняются те же действия, что и в процессе зашифровывания, но в обратном порядке.

Итеративный процесс расшифровывания может быть описан следующими формулами:

$R_{i-1} = L_i,\ i = 1,2,\ldots,16;$

$L_{i-1} = R_i \oplus f\left(L_i, K_i\right),\ i = 1,2,\ldots,16.$

Таким образом, для процесса расшифровывания с переставленным входным блоком $R_{16}L_{16}$ на первой итерации используется ключ K_{16}, на второй итерации - K_{15} и т.д. На 16-й итерации используется ключ K_1. На последнем шаге итерации будут получены последовательности L_0 и R_0, которые конкатенируются в 64-битовую последовательность L_0R_0. Затем в этой последовательности 64 бита переставляются в соответствии с матрицей IP. Результат такого преобразования - исходная последовательность бит (расшифрованное 64-битовое значение).

5.2. Реализация функции шифрования в алгоритме DES. Алгоритм вычисления ключей

5.2.1. Функция шифрования в DES

Схема вычисления функции шифрования $f\left(R_{i-1}, K_{\mathrm{i}}\right)$ показана на рис. 5.2.

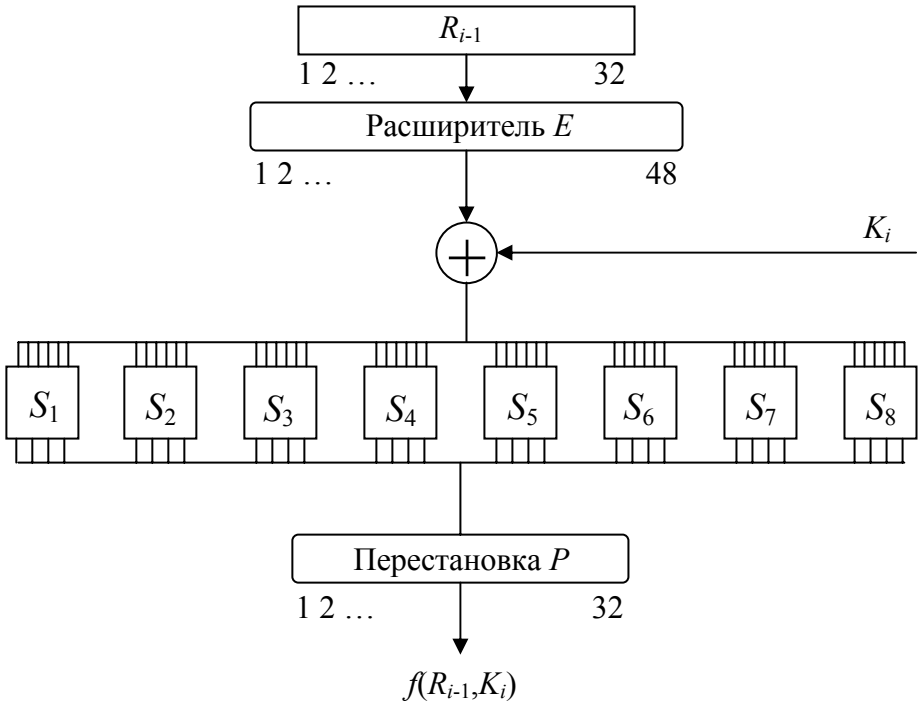


Рис. 5.2. Схема вычисления функции шифрования f

Для вычисления значения функции f используются:

– функция E (расширение 32 бит до 48);
– функция $S_1, S_2, \ldots, S_8$ (преобразование 6-битового числа в 4-битовое);
– функция P (перестановка бит в 32-битовой последовательности).

Приведем определения этих функций.

Аргументами функции шифрования f являются $R_{i\text{-}1}$ (32 бита) и K_i (48 бит). Результат функции $E\left(R_{i\text{-}1}\right)$ есть 48-битовое число. Функция расширения E, выполняющая расширение 32 бит до 48 (принимает блок из 32 бит и порождает блок из 48 бит), определяется табл. 5.3.

Таблица 5.3

Функция E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

В соответствии с табл. 5.3 первые три бита $E\left(R_{i\text{-}1}\right)$ – это биты 32, 1 и 2, а последние – 31, 32 и 1. Полученный результат (обозначим его $E\left(R_{i\text{-}1}\right)$) складывается по модулю 2 с текущим значением ключа K_i и затем разбивается на восемь 6-битовых блоков $B_1, B_2, \ldots, B_8 = E\left(R_{i-1}\right) \oplus K_i$.

Таблица 5.4

Функции *S*

		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_1	**0**	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	4	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	**0**	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	**0**	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	**0**	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	**0**	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	**0**	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	1	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	**0**	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	**0**	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Далее каждый из этих блоков используется как номер элемента в функциях – матрицах $S_1, S_2, \ldots, S_8$, содержащих 4-битовые значения (табл. 5.4).

Следует отметить, что выбор элемента в матрице S осуществляется достаточно оригинальным образом. Пусть на вход матрицы S поступает 6-битовый блок $B_j = b_1b_2b_3b_4b_5b_6$, тогда 2-битовое число b_1b_6 указывает номер строки матрицы, а 4-битовое число $b_2b_3b_4b_5$ - номер столбца. Например, если на вход матрицы S_1 поступает 6-битовый блок $B_1 = b_1b_2b_3b_4b_5b_6 = 100110_{(2)}$, то 2-битовое число $b_1b_6 = 10_{(2)} = 2_{(2)}$ указывает строку с номером 2 матрицы S_1, а 4-битовое число $b_2b_3b_4b_5 = 0011_{(2)} = 3_{(10)}$ указывает столбец с номером 3 матрицы S_1. Это означает, что в матрице S_1 блок $B_1 = 100110$ выбирает элемент на пересечении строки с номером 2 и столбца с номером 3, т.е. элемент $8_{(10)} = 1000_{(2)}$. Совокупность 6-битовых блоков $B_1, B_2, \ldots, B_8$ обеспечивает выбор 4-битового элемента в каждой из матриц $B_1, B_2, \ldots, B_8$.

В результате получаем $S_1(B_1), S_2(B_1), \ldots, S_8(B_1)$, т.е. 32-битовый блок (поскольку матрицы S содержат 4-битовые элементы). Этот 32-битовый блок преобразуется с помощью функции перестановки бит P (табл. 5.5).

Таблица 5.5

Функция P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Таким образом, функция шифрования

$$f(R_{i-1}, K_i) = P(S_1(B_1), \ldots, S_8(B_1)).$$

5.2.2. Алгоритм вычисления ключей

Как нетрудно заметить, на каждой итерации используется новое значение ключа K_i (длиной 48 бит). Новое значение ключа K_i вычисляется из начального ключа K (рис. 5.3).

Ключ K представляет собой 64-битовый блок с 8 битами контроля по четности, расположенными в позициях 8, 16, 24, 32, 40, 48, 56, 64. Для

удаления контрольных бит и подготовки ключа к работе используется функция *G* первоначальной подготовки ключа (табл. 5.6).

Таблица 5.6

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

Таблица 5.6 разделена на две части. Результат преобразования $G(\mathrm{K})$ разбивается на две половины C_0 и D_0, по 28 бит каждая. Первые четыре строки матрицы *G* определяют, как выбираются биты последовательности *C* (первым битом C_0 будет бит 57 ключа шифра, затем бит 49 и т.д., а последними битами – биты 44 и 36 ключа).

Следующие четыре строки матрицы *G* определяют, как выбираются биты последовательности D_0 (т.е. последовательность D_0 будет состоять из бит 63, 55, 47, ..., 12, 4 ключа шифра).

Как видно из табл. 5.6, для генерации последовательностей C_0 и D_0 не используются биты 8, 16, 24, 32, 40, 48, 56 и 64 ключа шифра. Эти биты не влияют на шифрование и могут служить для других целей (например, для контроля по четности). Таким образом, в действительности ключ шифра является 56-битовым.

После определения C_0 и D_0 рекурсивно определяются C_i и D_i, $i = 1,2,\ldots,16$. Для этого применяются операции циклического сдвига влево на один или два бита в зависимости от номера шага итерации, как показано в табл. 5.7.

Таблица 5.7

Таблица сдвигов для вычисления ключа

Итерация	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Сдвиг влево	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

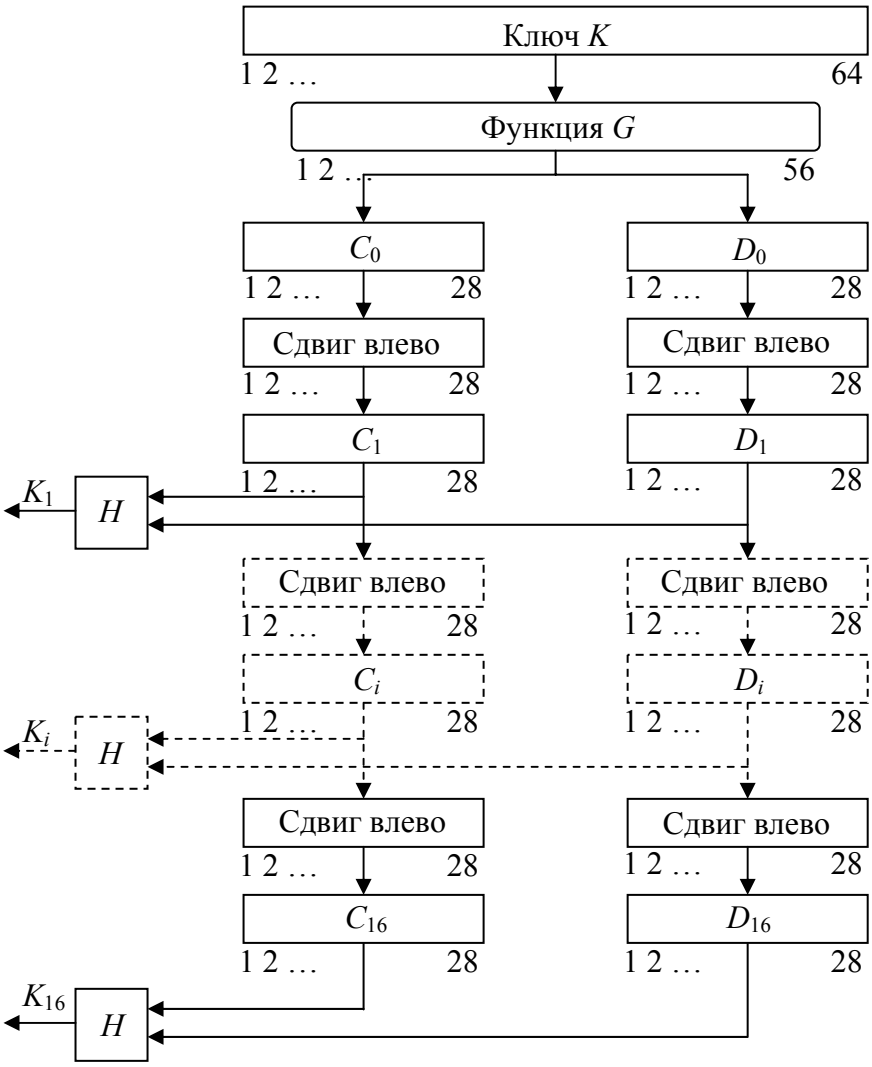


Рис. 5.3. Схема алгоритма вычисления ключей K_i

Операции сдвига выполняются для последовательностей C_i и D_i независимо. Например, последовательность C_3 получается посредством циклического сдвига влево на две позиции последовательности C_2, а последовательность D_3 – посредством сдвига влево на две позиции последовательности D_2, C_{16} и D_{16} получаются из C_{15} и D_{15} посредством сдвига влево на одну позицию.

Ключ K_i, определяемый на каждом шаге итерации, есть результат выбора конкретных бит из 56-битовой последовательности C_iD_i и их перестановки. Другими словами, ключ $K_i = H\left(C_iD_i\right)$, где функция H определяется матрицей, завершающей обработку ключа (табл. 5.8). Как следует из табл. 5.8, первым битом ключа K_i будет 14-й бит последовательности C_iD_i, вторым – 17-й бит, 47-м битом ключа K_i будет 29-й бит C_iD_i, а 48-м битом – 32-й бит C_iD_i.

Таблица 5.8

Функция *H*

14	17	11	24	1	5
3	28	15	6	21	10
23	19	22	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

5.2.3. Реальные критерии проектирования

После появления публикаций о дифференциальном криптоанализе IBM раскрыла критерии проектирования S-блоков и P-блока. Критериями проектирования S-блоков являлись:

– У каждого S-блока 6 входных битов и 4 выходных бита. (Это самый большой размер, который мог быть реализован в одной микросхеме по технологии 1974 года.)

– Ни один выходной бит S-блока не должен быть слишком близок к линейной функции входных битов.

– Если зафиксировать крайние левый и правый биты S-блока, изменяя 4 средних бита, то каждый возможный 4-битовый результат получается только один раз.

– Если два входа S-блока отличаются только одним битом, результаты должны отличаться, по крайней мере, на 2 бита.

– Если два входа S-блока отличаются только двумя центральными битами, результаты должны отличаться по крайней мере на 2 бита.

– Если два входа S-блока отличаются двумя первыми битами, а последние их 2 бита совпадают, результаты не должны быть одинаковыми.

– Для любого ненулевого 6-битового отличия между входами, не более чем 8 из 32 пар входов могут приводить на выходе к одинаковому различию.

– Аналогичный предыдущему критерий, но для случая трех активных S-блоков.

Критериями проектирования P-блока являлись:

– 4 выходных бита каждого S-блока на этапе i распределены так, что 2 из них влияют на средние биты S-блоков на этапе $i + 1$, а другие 2 бита влияют на последние биты.

– 4 выходных бита каждого S-блока влияют на шесть различных S-блоков, никакие 2 не влияют на один и тот же S-блок.

– Если выходной бит одного S-блока влияет на средние биты другого S-блока, то выходной бит этого другого S-блока не может влиять на средние биты первого S-блока.

Сегодня совсем нетрудно генерировать S-блоки, но в начале 70-х это было нелегкой задачей. Программы, готовившие S-блоки, работали месяцами.

5.2.4. Слабые и полуслабые ключи

Из-за того, что первоначальный ключ изменяется при получении подключа для каждого этапа алгоритма, определенные первоначальные ключи являются слабыми. Вспомните, первоначальное значение расщепляется на две половины, каждая из которых сдвигается независимо. Если все биты каждой половины равны 0 или 1, то для всех этапов алгоритма используется один и тот же ключ. Это может произойти, если ключ состоит из одних 1, из одних 0, или если одна половина ключа состоит из одних 1, а другая – из одних 0. Кроме того, два слабых ключа обладают другими свойствами, снижающими их безопасность.

Четыре слабых ключа показаны в шестнадцатиричном виде в 1-й. (Не забывайте, что каждый восьмой бит – это бит четности.)

Значение слабого ключа (с битами четности)				Действительный ключ
0101	0101	0101	0101	0000000 0000000
1F1F	1F1F	OEOE	OEOE	0000000 FFFFFFF
EOEO	EOEO	F1F1	F1F1	FFFFFFF 0000000
FEFE	FEFE	FEFE	FEFE	FFFFFFF FFFFFFF

Кроме того, некоторые пары ключей при шифровании переводят открытый текст в идентичный шифротекст. Иными словами, один из ключей пары может расшифровать сообщения, зашифрованные другим ключом пары. Это происходит из-за метода, используемого DES для генерации подключей – вместо 16 различных подключей эти ключи генерируют только два различных подключа. В алгоритме каждый из этих подключей используется восемь раз. Эти ключи, называемые полуслабыми ключами, в шестнадцатиричном виде приведены в 0-й.

Полуслабые пары ключей DES

01FE	01FE	01FE	01FE	и	FE01	FE01	FE01	FE01
1FE0	1FE0	0EF1	0EF1	и	E01F	E01F	F10E	F10E
01E0	01E0	01F1	01F1	и	E001	E001	F101	F101
1FFE	IEEE	0EFE	0EFE	и	FE1F	FE1F	FE0E	FE0E
011F	011F	010E	010E	и	1F01	1F01	0E01	0E01
E0FE	E0FE	FIFE	FIFE	и	FEE0	FEE0	FEE1	FEE1

Ряд ключей генерирует только четыре подключа, каждый из которых четыре раза используется в алгоритме.

Возможно слабые ключи DES

1F	1F	01	01	0E	0E	01	01		E0	01	01	E0	Fl	01	01	Fl
01	1F	1F	01	01	0E	0E	01		FE	1F	01	E0	FE	0E	01	Fl
1F	01	01	1F	0E	01	01	0E		FE	01	1F	E0	FE	01	0E	Fl
01	01	1F	1F	01	01	0E	0E		E0	1F	1F	E0	Fl	0E	0E	Fl
E0	E0	01	01	Fl	Fl	01	01		FE	01	01	FE	FE	01	01	FE
FE	FE	01	01	FE	FE	01	01		E0	1F	01	FE	Fl	0E	01	FE
FE	E0	1F	01	FE	Fl	0E	01		E0	01	1F	FE	Fl	01	0E	FE
E0	FE	1F	01	Fl	FE	0E	01		FE	1F	1F	FE	FE	0E	0E	FE
FE	E0	01	1F	FE	Fl	01	0E		1F	FE	01	E0	0E	FE	01	Fl
E0	FE	01	1F	Fl	FE	01	0E		01	FE	1F	E0	01	FE	0E	Fl
EO	E0	1F	1F	Fl	Fl	0E	0E		1F	E0	01	FE	0E	Fl	01	FE
FE	FE	1F	1F	FE	FE	0E	0E		01	E0	1F	FE	01	Fl	0E	FE
FE	1F	E0	01	FE	0E	Fl	01		01	01	E0	E0	01	01	Fl	Fl
EO	1F	FE	01	Fl	0E	FE	01		1F	1F	E0	E0	0E	0E	Fl	Fl
FE	01	E0	1F	FE	01	Fl	0E		1F	01	FE	E0	0E	01	FE	Fl
EO	01	FE	1F	Fl	01	FE	0E		01	1F	FE	E0	01	0E	FE	Fl
01	E0	E0	01	01	Fl	Fl	01		1F	01	E0	FE	0E	01	Fl	FE
1F	FE	E0	01	0E	FE	F0	01		01	1F	E0	FE	01	0E	Fl	FE
1F	E0	FE	01	0E	Fl	FE	01		01	01	FE	FE	01	01	FE	FE
01	FE	FE	01	01	FE	FE	01		1F	1F	FE	FE	0E	0E	FE	FE
1F	E0	E0	1F	0E	Fl	Fl	0E		FE	FE	E0	E0	FE	FE	Fl	Fl
01	FE	E0	1F	01	FE	Fl	0E		E0	FE	FE	E0	Fl	FE	FE	Fl
01	E0	FE	1F	01	Fl	FE	0E		FE	E0	E0	FE	FE	Fl	Fl	FE
1F	FE	FE	1F	0E	FE	FE	0E		E0	E0	FE	FE	Fl	Fl	FE	FE

5.3. Основные режимы работы алгоритма DES. Комбинирование блочных алгоритмов

Чтобы воспользоваться алгоритмом DES для решения разнообразных криптографических задач, разработаны четыре рабочих режима:

– электронная кодовая книга ECB (Electronic Code Book);

– сцепление блоков шифра CBC (Cipher Block Chaining);
– обратная связь по шифротексту CPB (Cipher FeedBack);
– обратная связь по выходу OFB (Output FeedBack).

5.3.1. Режим Электронная кодовая книга

Длинный файл разбивают на 64-битовые отрезки (блоки) по 8 байт. Каждый из этих блоков шифруют независимо с использованием одного и того же ключа шифрования (рис. 5.4).

Основное достоинство – простота реализации. Недостаток – относительно слабая устойчивость против квалифицированных криптоаналитиков. Из-за фиксированного характера шифрования при ограниченной длине блока 64 бита возможно проведение криптоанализа «со словарем». Блок такого размера может повториться в сообщении вследствие большой избыточности в тексте на естественном языке. Это приводит к тому, что идентичные блоки открытого текста в сообщении будут представлены идентичными блоками шифротекста, что дает криптоаналитику некоторую информацию о содержании сообщения.

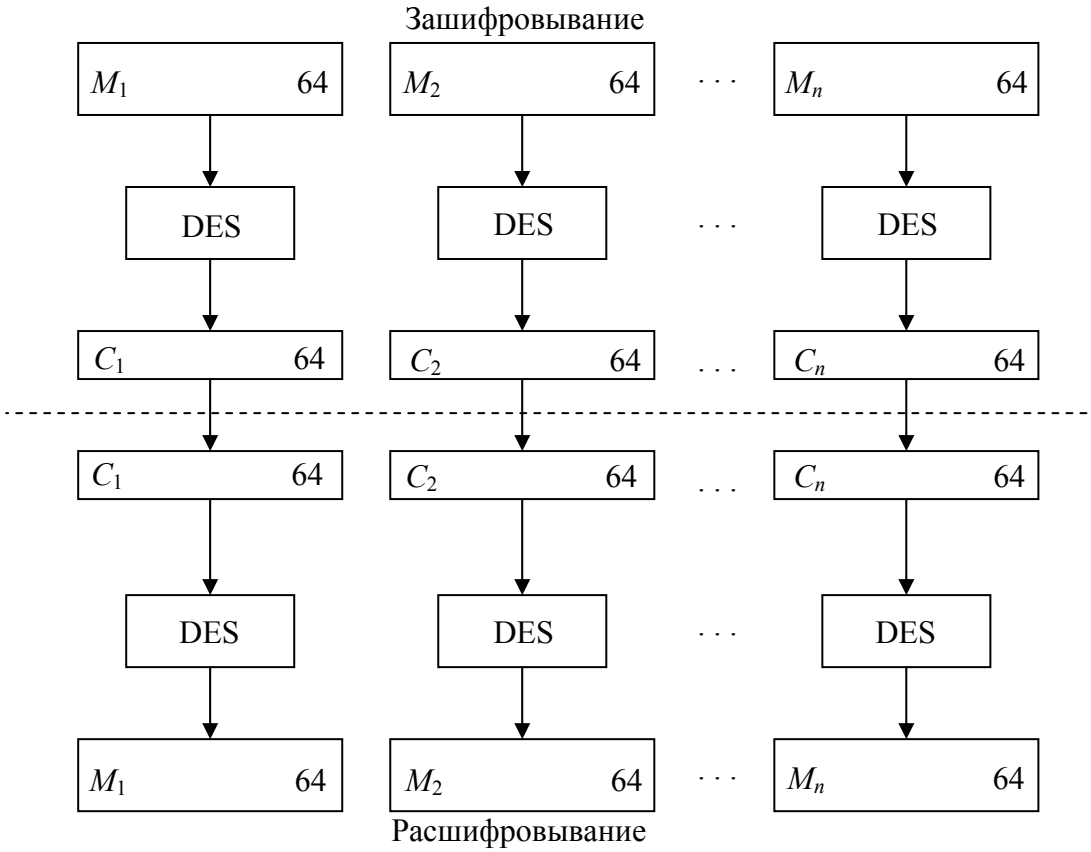


Рис. 5.4. Схема алгоритма DES в режиме электронной кодовой книги

5.3.2. Режим Сцепление блоков шифра

В этом режиме исходный файл M разбивается на 64-битовые блоки: $M = M_1 M_2 ... M_n$. Первый блок M_1 складывается по модулю 2 с 64-битовым начальным вектором IV, который меняется ежедневно и держится в секрете (рис. 5.5). Полученная сумма затем шифруется с использованием ключа DES, известного и отправителю, и получателю информации. Полученный 64-битовый шифр C_1 складывается по модулю 2 со вторым блоком текста, результат шифруется и получается второй 64-битовый шифр C_2 и т.д. Процедура повторяется до тех пор, пока не будут обработаны все блоки текста.

Таким образом, для всех $i = 1 \ldots n$ (n – число блоков) результат шифрования C определяется следующим образом: $C_i = \mathrm{DES}\left(M_i \oplus C_{i-1}\right)$, где $C_0 = \mathrm{IV}$ – начальное значение шифра, равное начальному вектору (вектору

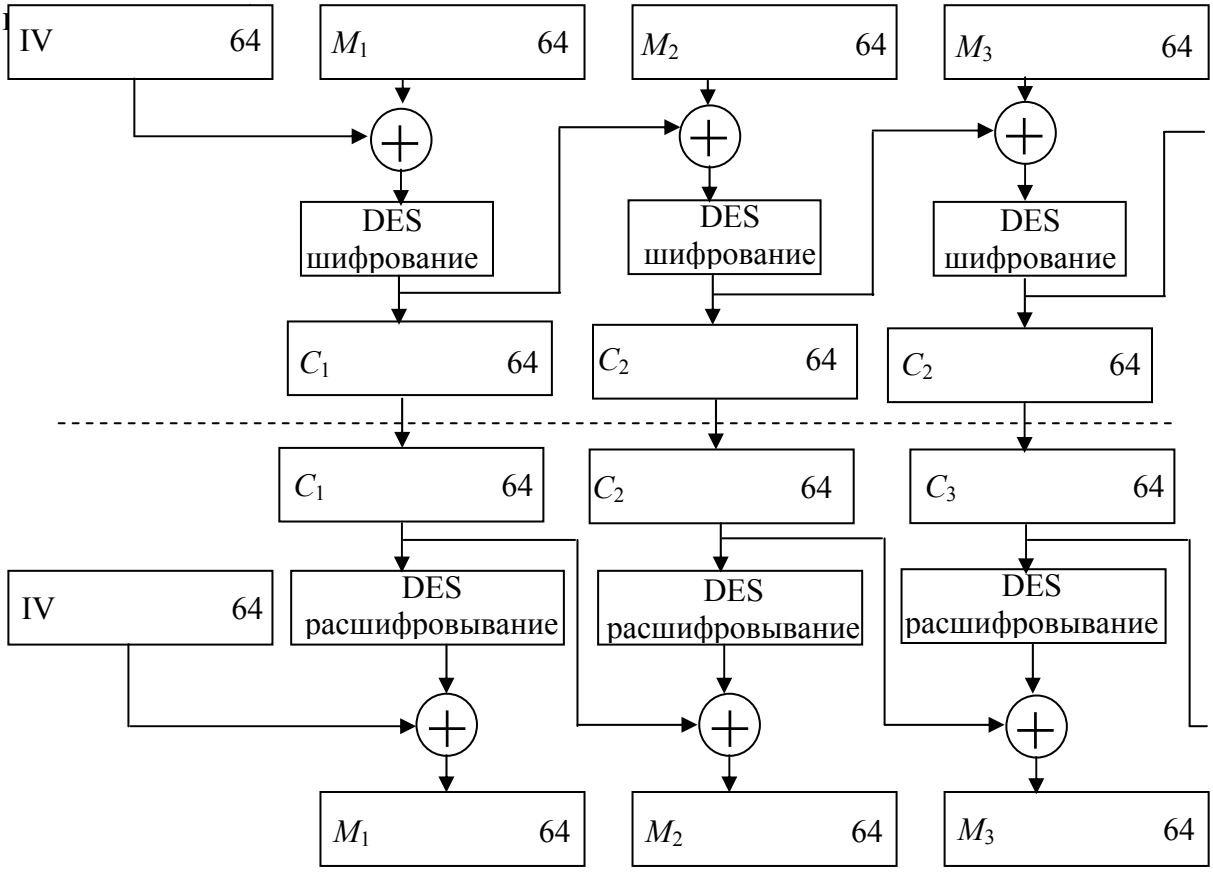


Рис. 5.5. Схема алгоритма DES в режиме сцепления блоков шифра

Очевидно, что последний 64-битовый блок шифротекста является функцией секретного ключа, начального вектора и каждого бита открытого текста независимо от его длины. Этот блок шифротекста называют кодом аутентификации сообщения (КАС).

Код КАС может быть легко проверен получателем, владеющим секретным ключом и начальным вектором, путем повторения процедуры, выполненной отправителем. Посторонний, однако, не может осуществить генерацию КАС, который воспринялся бы получателем как подлинный, чтобы добавить его к ложному сообщению, либо отделить КАС от истинного сообщения для использования его с измененным или ложным сообщением.

Достоинство данного режима в том, что он не позволяет накапливаться ошибкам при передаче.

Блок M_i является функцией только C_{i-1} и C_i. Поэтому ошибка при передаче приведет к потере только двух блоков исходного текста.

5.3.3. Режим Обратная связь по шифротексту

В этом режиме размер блока может отличаться от 64 бит (рис 5.6). Файл, подлежащий шифрованию (расшифровыванию), считывается последовательными блоками длиной k бит ($k = 1 \ldots 64$).

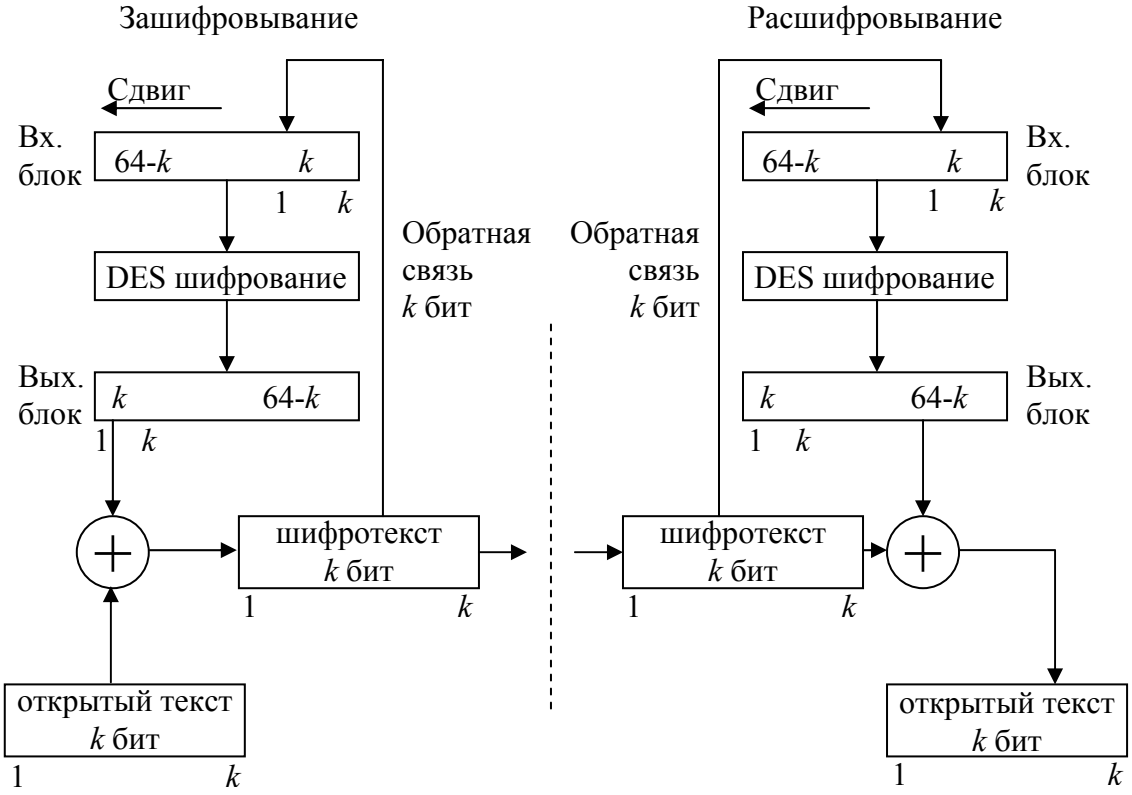


Рис. 5.6. Схема алгоритма DES в режиме обратной связи по шифротексту

Входной блок (64-битовый регистр сдвига) вначале содержит вектор инициализации, выровненный по правому краю. Предположим, что в результате разбиения на блоки мы получили n блоков длиной k бит каждый (остаток дописывается нулями или пробелами). Тогда для любого $i = 1 \ldots n$

блок шифротекста $C_i = M_i \oplus P_{i-1}$, где P_{i-1} обозначает k старших бит предыдущего зашифрованного блока.

Обновление сдвигового регистра осуществляется путем удаления его старших k бит и записи C_i в регистр. Восстановление зашифрованных данных также выполняется относительно просто: P_{i-1} и C_i вычисляются аналогичным образом и $M_i = C_i \oplus P_{i-1}$.

5.3.4. Режим Обратная связь по выходу

Этот режим тоже использует переменный размер блока и сдвиговый регистр, инициализируемый так же, как в режиме СРВ, а именно - входной блок вначале содержит вектор инициализации IV, выровненный по правому краю (рис. 5.7). При этом для каждого сеанса шифрования данных необходимо использовать новое начальное состояние регистра, которое должно пересылаться по каналу открытым текстом.

Положим $M = M_1 M_2 ... M_n$ для всех $i = 1 \ldots n$ $C_i = M_i \oplus P_i$, где P_i – старшие k бит операции $\mathrm{DES}(C_{i-1})$. Отличие от режима обратной связи по шифротексту состоит в методе обновления сдвигового регистра.

Это осуществляется путем отбрасывания старших k бит и дописывания справа P_i.

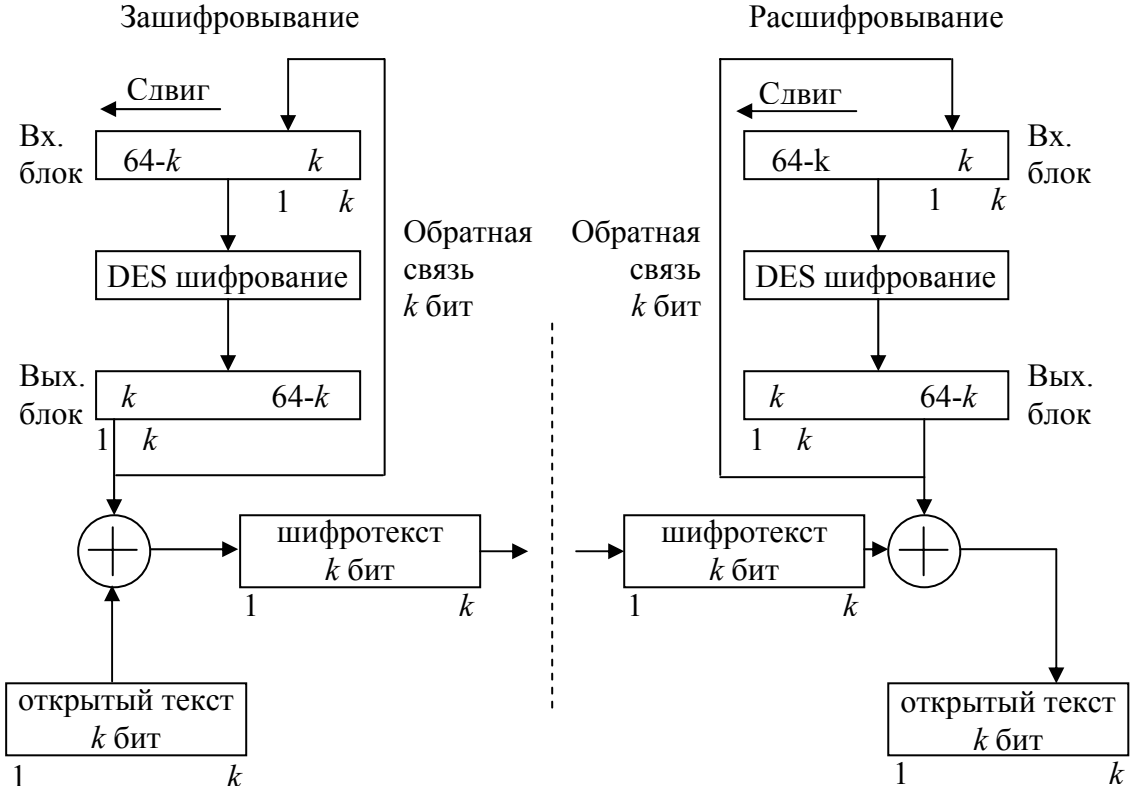


Рис. 5.7. Схема алгоритма DES в режиме обратной связи по выходу

5.3.5. Комбинирование блочных алгоритмов

В настоящее время блочный алгоритм DES считается относительно безопасным алгоритмом шифрования. Он подвергался тщательному криптоанализу в течение 20 лет, и самым практичным способом его взлома является метод перебора всех возможных вариантов ключа. Ключ DES имеет длину 56 бит, поэтому существует 2^{56} возможных вариантов такого ключа. Однако, с учетом вычислительных мощностей современных компьютеров, недалеко то время, когда поиск ключа DES методом полного перебора станет возможным для мощных в финансовом отношении государственных и коммерческих организаций. Возникает вопрос: нельзя ли использовать DES в качестве элемента для создания другого алгоритма с более длинным ключом?

В принципе, существует много способов комбинирования блочных алгоритмов для получения новых алгоритмов. Одним из таких способов комбинирования является многократное шифрование, т.е. использование блочного алгоритма несколько раз с разными ключами для шифрования одного и того же блока открытого текста. Двукратное шифрование блока открытого текста одним и тем же ключом не приводит к положительному результату. При использовании одного и того же алгоритма такое шифрование не влияет на сложность криптоаналитической атаки полного перебора.

Рассмотрим эффективность двукратного шифрования блока открытого текста с помощью двух разных ключей. Сначала шифруют блок P ключом K_1, а затем получившийся шифротекст $E_{K_1}(P)$ шифруют ключом K_2. В результате двукратного шифрования получают криптограмму $C = E_{K_2}(E_{K_1}(P))$. Расшифровывание является обратным процессом: $P = D_{K_1}(D_{K_2}(C))$.

Если блочный алгоритм обладает свойствами группы, то всегда найдется такой ключ K_3, что $C = E_{K_2}(E_{K_1}(P)) = E_{K_3}(P)$. Если же блочный алгоритм не является группой, то результирующий двукратно шифрованный блок текста окажется намного сложнее для взлома методом полного перебора вариантов. Вместо 2^n попыток, где n – длина ключа в битах, потребуется 2^{2n} попыток. В частности, если n = 64, то двукратно зашифрованный блок текста потребует 2^{128} попыток для нахождения ключа.

Однако Р. Меркль и М. Хеллман показали на примере DES, что, используя метод «обмена времени на память» и криптоаналитическую атаку «встреча посередине», можно взломать такую схему двукратного шифро-

вания за 2^n попыток. Правда, эта атака потребует очень большого объема памяти (для алгоритма с 56-битовым ключом потребуется 2^{56} 64-битовых блоков или 10^{17} бит памяти).

Более привлекательную идею предложил У. Тачмен. Суть этой идеи состоит в том, чтобы шифровать блок открытого текста P три раза с помощью двух ключей K_1 и K_2 (рис. 5.8). Процедура шифрования: $C = E_{K_1}(D_{K_2}(E_{K_1}(P)))$, т.е. блок открытого текста P сначала шифруется ключом K_1, затем расшифровывается ключом K_2 и окончательно зашифровывается ключом K_1. Этот режим иногда называют режимом EDE (encrypt – decrypt – encrypt). Введение в данную схему операции расшифровывания D_{K_2} позволяет обеспечить совместимость этой схемы со схемой однократного использования алгоритма DES. Если в схеме трехкратного использования DES выбрать все ключи одинаковыми, то эта схема превращается в схему однократного использования DES. Процедура расшифровывания выполняется в обратном порядке: $P = D_{K_1}(E_{K_2}(D_{K_1}(C)))$, т.е. блок шифротекста C сначала расшифровывается ключом K_1, затем зашифровывается ключом K_2 и окончательно расшифровывается ключом K_1.

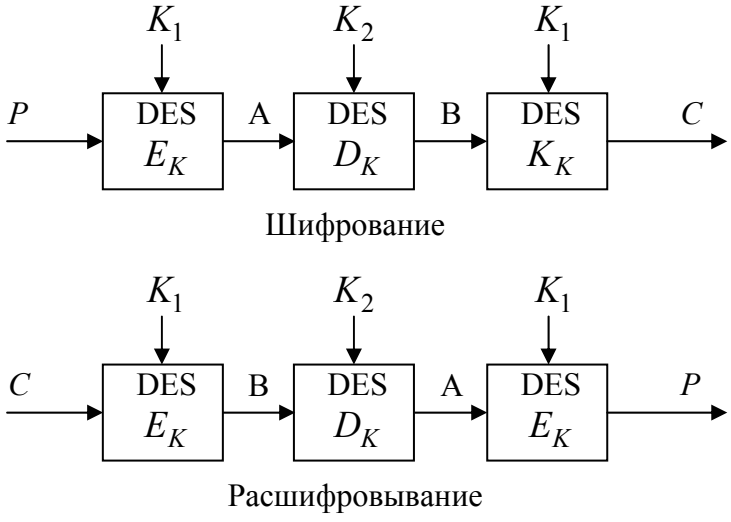


Рис. 5.8. Схемы трехкратного применения алгоритма DES с двумя разными ключами

Если исходный блочный алгоритм имеет n-битовый ключ, то схема трехкратного шифрования имеет $2n$-битовый ключ. Чередование ключей K_1 и K_2 позволяет предотвратить криптоаналитическую атаку «встреча посредине». При трехкратном шифровании можно применить три различных ключа. При этом возрастает общая длина результирующего ключа.

Процедуры шифрования и расшифровывания описываются выражениями: $C = E_{K_3}(D_{K_2}(E_{K_1}(P)))$, $P = D_{K_1}(E_{K_2}(D_{K_3}(C)))$. Трехключевой вариант имеет большую стойкость.

5.4. Логика построения шифра, структура ключевой информации и основной шаг криптопреобразования стандарта ГОСТ 28147-89

5.4.1. Введение в алгоритм

В качестве официального алгоритма криптографического преобразования данных для систем обработки информации в Республике Беларусь выбран алгоритм, стандартизованный в ГОСТ 28147-89. Он предназначен для аппаратной и программной реализации, удовлетворяет криптографическим требованиям и не накладывает ограничений на степень секретности защищаемой информации. Стандарт закреплен ГОСТом № 28147-89, принятом в 1989 году в СССР.

Элементы данных при рассмотрении данного алгоритма обозначаются заглавными латинскими буквами с наклонным начертанием (например, X). Через $|X|$обозначается размер элемента данных X в битах. Таким образом, если интерпретировать элемент данных X как целое неотрицательное число, можно записать следующее неравенство: $0 \le X < 2^{|X|}$.

Если элемент данных состоит из нескольких элементов меньшего размера, то обозначается следующим образом: $X = (X_0, X_1, ..., X_{n-1}) = X_0 \,||\, X_1 \,||\, ... \,||\, X_{n-1}$. Процедура объединения нескольких элементов данных в один называется конкатенацией данных и обозначается символом «||». Естественно, для размеров элементов данных должно выполняться следующее соотношение: $|X| = |X_0| + |X_1| + ... + |X_{n-1}|$. При задании сложных элементов данных и операции конкатенации составляющие элементы данных перечисляются в порядке возрастания старшинства. Иными словами, если интерпретировать составной элемент и все входящие в него элементы данных как целые числа без знака, то можно записать следующее равенство:

$$(X_0, X_1, ..., X_{n-1}) = X_0 \,||\, X_1 \,||\, ... \,||\, X_{n-1} =$$
$$= X_0 + 2^{|X_0|}(X_1 + 2^{|X_1|}(...(X_{n-1} + 2^{|X_{n-1}|} X_{n-1})...))$$

В алгоритме элемент данных может интерпретироваться как массив отдельных битов, в этом случае биты обозначаем той же самой буквой, что и массив, но в строчном варианте, как показано на следующем примере:

$$X = (x_0, x_1, ..., x_{n-1}) = x_0 + 2^1 \cdot x_1 + ... + 2^{n-1} \cdot x_{n-1}.$$

Если над элементами данных выполняется некоторая операция, имеющая логический смысл, то предполагается, что данная операция выполняется над соответствующими битами элементов.

Если внимательно изучить оригинал ГОСТ 28147-89, можно заметить, что в нем содержится описание алгоритмов нескольких уровней. На самом верхнем находятся практические алгоритмы, предназначенные для шифрования массивов данных и выработки для них имитовставки. Все они опираются на три алгоритма низшего уровня, называемые в тексте ГОСТа циклами. Эти фундаментальные алгоритмы можно назвать как базовые циклы, чтобы отличать их от всех прочих циклов. Они имеют следующие названия и обозначения, последние приведены в скобках:

– цикл зашифровывания ($32-З$);
– цикл расшифровывания ($32-Р$);
– цикл выработки имитовставки ($16-З$).

В свою очередь, каждый из базовых циклов представляет собой многократное повторение одной процедуры, называемой основным шагом криптопреобразования.

Таким образом, надо понять три следующие вещи:

а) что такое основной шаг криптопреобразования;

б) как из основных шагов складываются базовые циклы;

в) как из трех базовых циклов складываются все практические алгоритмы ГОСТа.

В ГОСТе ключевая информация состоит из двух структур данных. Помимо собственно ключа, необходимого для всех шифров, она содержит еще и таблицу замен. Ниже приведены основные характеристики ключевых структур ГОСТа.

1. Ключ является массивом из восьми 32-битовых элементов кода, далее он обозначается символом K: $K = \{K_i\}_{0 \le i < 2^{32}}$. В ГОСТе элементы ключа используются как 32-разрядные целые числа без знака: $0 \le K_i < 2^{32}$. Таким образом, размер ключа составляет $32 \cdot 8 = 256$ бит или 32 байта.

2. Таблица замен может быть представлена в виде матрицы размера 8×16, содержащей 4-битовые элементы, которые можно представить в виде целых чисел от 0 до 15. Строки таблицы замен называются узлами замен, они должны содержать различные значения, то есть каждый узел замен должен содержать 16 различных чисел от 0 до 15 в произвольном порядке. Таблица замен обозначается символом H: $H = \{H_{i,j}\}_{0 \le j < 15}^{0 \le i < 7}$, $0 \le H_{i,j} < 15$.

Таким образом, общий объем таблицы замен равен: 8 узлов × 16 элементов/узел ×4 бита/элемент = 512 бит или 64 байта.

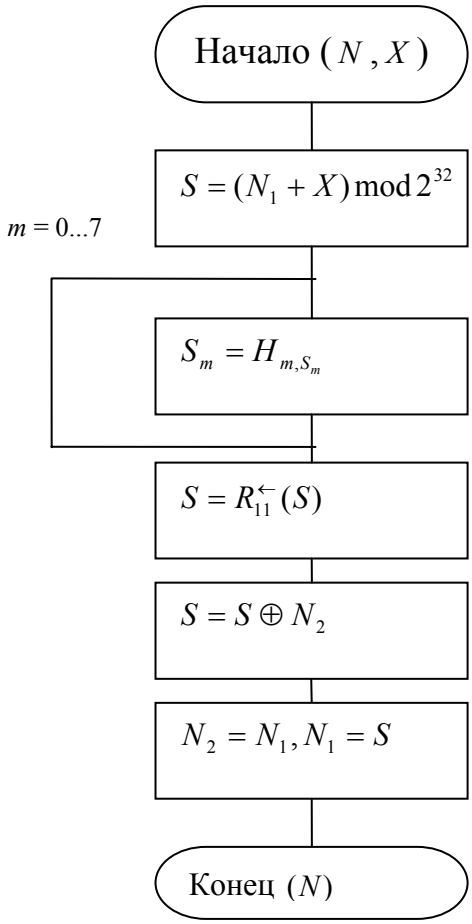


Рис. 5.9. Схема основного шага криптопреобразования алгоритма ГОСТ 28147-89

5.4.2. Основной шаг криптопреобразования

Основной шаг криптопреобразования по своей сути является оператором, определяющим преобразование 64-битового блока данных. Дополнительным параметром этого оператора является 32-битовый блок, в качестве которого используется какой-либо элемент ключа. Схема алгоритма основного шага приведена на рис. 5.9. Ниже даны пояснения к алгоритму основного шага:

Шаг 0.

Определяет исходные данные для основного шага криптопреобразования:

– N – преобразуемый 64-битовый блок данных, в ходе выполнения шага его младшая (N_1) и старшая (N_2) части обрабатываются как отдельные 32-битовые целые числа без знака. Таким образом, можно записать $N = N_1, N_2)$.

– X – 32-битовый элемент ключа.

Шаг 1.

Сложение с ключом. Младшая половина преобразуемого блока складывается по модулю 2^{32} с используемым на шаге элементом ключа, результат передается на следующий шаг.

Шаг 2.

Поблочная замена. 32-битовое значение, полученное на предыдущем шаге, интерпретируется как массив из восьми 4-битовых блоков кода: $S = (S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7)$.

Далее значение каждого из восьми блоков заменяется новым, которое выбирается по таблице замен следующим образом: значение блока S_i меняется на S_i-тый по порядку элемент (нумерация с нуля) i-того узла замен (т.е. i-той строки таблицы замен, нумерация также с нуля). Другими словами, в качестве замены для значения блока выбирается элемент из таблицы замен с номером строки, равным номеру заменяемого блока, и номером столбца, равным значению заменяемого блока как 4-битового целого неотрицательного числа. Теперь становится понятным размер таблицы замен: число строк в ней равно числу 4-битовых элементов в 32-битовом блоке данных, то есть восьми, а число столбцов равно числу различных значений 4-битового блока данных, равному как известно 24, шестнадцати.

Шаг 3.

Циклический сдвиг на 11 бит влево. Результат предыдущего шага сдвигается циклически на 11 бит в сторону старших разрядов и передается на следующий шаг. На схеме алгоритма символом $R_{11}^{\leftarrow}$ обозначена функция циклического сдвига своего аргумента на 11 бит влево, т.е. в сторону старших разрядов.

Шаг 4.

Побитовое сложение: значение, полученное на шаге 3, побитно складывается по модулю 2 со старшей половиной преобразуемого блока.

Шаг 5.

Сдвиг по цепочке: младшая часть преобразуемого блока сдвигается на место старшей, а на ее место помещается результат выполнения предыдущего шага.

Шаг 6.

Полученное значение преобразуемого блока возвращается как результат выполнения алгоритма основного шага криптопреобразования.

5.5. Базовые циклы, основные режимы шифрования алгоритма ГОСТ 28147-89

5.5.1. Базовые циклы криптографических преобразований

ГОСТ относится к классу блочных шифров, то есть единицей обработки информации в нем является блок данных. Следовательно, в нем определены алгоритмы для криптографических преобразований, то есть для зашифровывания, расшифровывания и «учета» в контрольной комбинации одного блока данных. Именно эти алгоритмы и называются базовыми циклами ГОСТа, что подчеркивает их фундаментальное значение для построения этого шифра.

Базовые циклы построены из основных шагов криптографического преобразования. В процессе выполнения основного шага используется только один элемент ключа, в то время как ключ ГОСТ содержит восемь таких элементов. Следовательно, чтобы ключ был использован полностью, каждый из базовых циклов должен многократно выполнять основной шаг с различными его элементами. Вместе с тем в каждом базовом цикле все элементы ключа должны быть использованы одинаковое число раз, по соображениям стойкости шифра это число должно быть больше одного.

Базовые циклы заключаются в многократном выполнении *основного шага* с использованием разных элементов ключа и отличаются друг от друга только числом повторения шага и порядком использования ключевых элементов. Ниже приведен этот порядок для различных циклов.

1. Цикл зашифрования 32-З:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_0, K_1, K_2, K_3, K_4,$$
$$K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

2. Цикл расшифрования 32-Р:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0, K_7, K_6, K_5, K_4,$$
$$K_3, K_2, K_1, K_0, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

3. Цикл выработки имитовставки 16-З:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7.$$

Каждый из циклов имеет собственное буквенно-цифровое обозначение, соответствующее шаблону «$n - X$», где первый элемент обозначения (n), задает число повторений основного шага в цикле, а второй элемент

обозначения (X), буква, задает порядок зашифровывания («$З$») или расшифровывания («$Р$») в использовании ключевых элементов.

Цикл расшифровывания должен быть обратным циклу зашифровывания, то есть последовательное применение этих двух циклов к произвольному блоку должно дать в итоге исходный блок, что отражается следующим соотношением: $Ц_{32-Р}(Ц_{32-З}(T)) = T$, где T – произвольный 64-битовый блок данных, $Ц_X(T)$ – результат выполнения цикла X над блоком данных T. Для выполнения этого условия для алгоритмов, подобных ГОСТу, необходимо и достаточно, чтобы порядок использования ключевых элементов соответствующими циклами был взаимно обратным. В справедливости записанного условия для рассматриваемого случая легко убедиться, сравнив приведенные выше последовательности для циклов $32-З$ и $32-Р$.

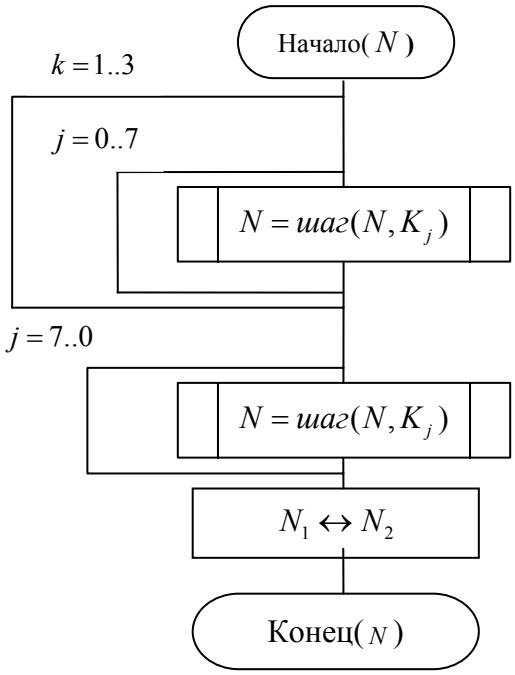


Рис. 5.10. Схема цикла зашифровывания $32-З$

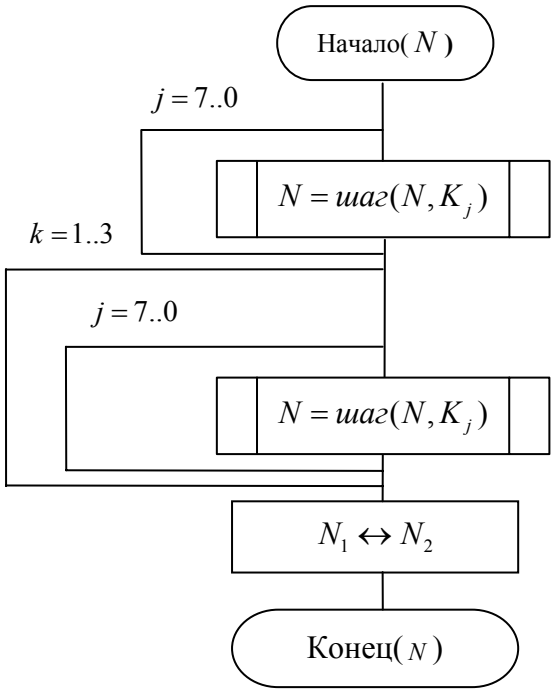


Рис. 5.11. Схема цикла расшифровывания $32-Р$

Из двух взаимно обратных циклов любой может быть использован для зашифровывания, тогда второй должен быть использован для расшифровывания данных, однако стандарт ГОСТ 28147-89 закрепляет роли за циклами и не предоставляет пользователю права выбора в этом вопросе.

Цикл выработки имитовставки вдвое короче циклов шифрования, порядок использования ключевых элементов в нем такой же, как в первых 16 шагах цикла зашифровывания, поэтому этот порядок в обозначении цикла кодируется той же самой буквой «*З*».

Схемы базовых циклов приведены на рисунках 5.10 – 5.12. Каждый из них принимает в качестве аргумента и возвращает в качестве результата 64-битовый блок данных, обозначенный на схемах *N*. Символ Шаг (*N, X*) обозначает выполнение основного шага криптопреобразования для блока *N* с использованием ключевого элемента *X*. Между циклами шифрования и вычисления имитовставки есть еще одно отличие: в конце базовых циклов шифрования старшая и младшая часть блока результата меняются местами, это необходимо для их взаимной обратимости.

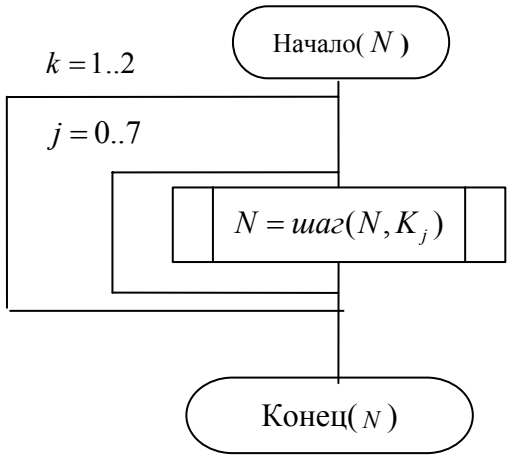


Рис. 5.12. Схема цикла выработки имитовставки 16 – *З*

5.5.2. Основные режимы шифрования

ГОСТ 28147-89 предусматривает три следующих режима шифрования данных:

- простая замена,
- гаммирование,
- гаммирование с обратной связью,

и один дополнительный режим выработки имитовставки.

В любом из этих режимов данные обрабатываются блоками по 64 бита, на которые разбивается массив, подвергаемый криптографическому преобразованию. Однако в двух режимах гаммирования есть возможность обработки неполного блока данных размером меньше 8 байт, что сущест-

венно при шифровании массивов данных с произвольным размером, который может быть не кратным 8 байтам.

Обозначения, используемые на схемах:

$T_o, T_{Ш}$ – массивы соответственно открытых и зашифрованных данных;

$T_i^0, T_i^{Ш}$ – i-тые по порядку 64-битовые блоки соответственно открытых и зашифрованных данных : $T_o = (T_1^o, T_2^o, ..., T_n^o)$, $T_{Ш} = (T_1^{Ш}, T_2^{Ш}, ..., T_n^{Ш})$, $1 \le i \le n$,последний блок может быть неполным: $|T_i^o| = |T_i^{Ш}| = 64$ при $1 \le i \le n, 1 \le |T_n^o| = |T_n^{Ш}| \le 64$;

n – число 64-битовых блоков в массиве данных;

$Ц_X$ – функция преобразования 64-битового блока данных по алгоритму базового цикла «X».

Простая замена

Зашифровывание в данном режиме заключается в применении цикла $32-3$ к блокам открытых данных, расшифровывание – цикла $32-P$ к блокам зашифрованных данных. Это наиболее простой из режимов, 64-битовые блоки данных обрабатываются в нем независимо друг от друга. Схемы алгоритмов зашифровывания и расшифровывания в режиме простой замены приведены на рис. 5.13 – 5.14.

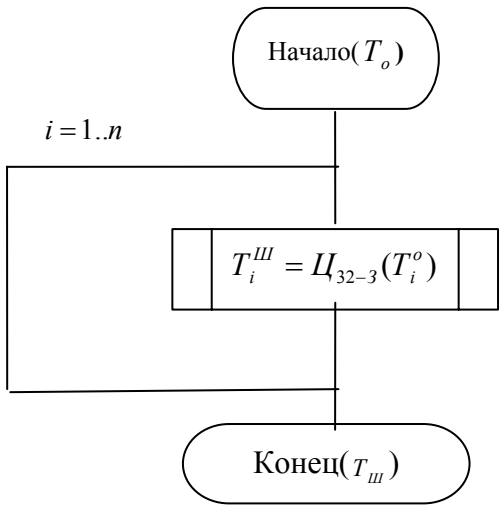


Рис. 5.13. Схема цикла зашифровывания данных в режиме простой замены

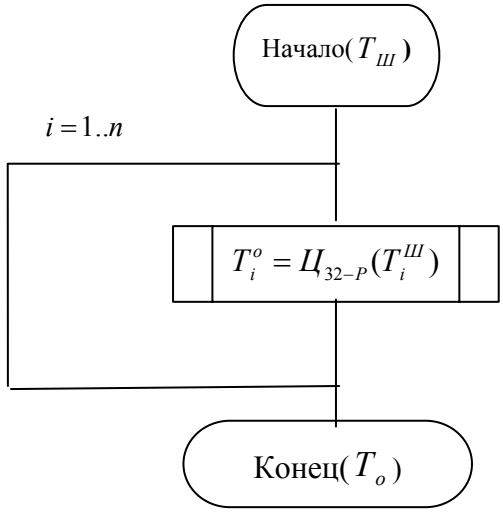


Рис. 5.14. Алгоритм расшифровывания данных в режиме простой замены

Размер массива открытых или зашифрованных данных, подвергающийся соответственно зашифровыванию или расшифровыванию, должен быть кратен 64 битам: $|T_o| = |T_{Ш}| = 64 \cdot n$, после выполнения операции размер полученного массива данных не изменяется.

Режим шифрования простой заменой имеет следующие особенности:

1. Так как блоки данных шифруются независимо друг от друга и от их позиции в массиве, при зашифровывании двух одинаковых блоков открытого текста получаются одинаковые блоки шифртекста и наоборот. Отмеченное свойство позволит криптоаналитику сделать заключение о тождественности блоков исходных данных, если в массиве зашифрованных данных ему встретились идентичные блоки, что является недопустимым для серьезного шифра.

2. Если длина шифруемого массива данных не кратна 8 байтам или 64 битам, возникает проблема, чем и как дополнять последний неполный блок данных массива до полных 64 бит. Очевидные решения типа «дополнить неполный блок нулевыми битами» или «дополнить неполный блок фиксированной комбинацией нулевых и единичных битов» могут при определенных условиях дать в руки криптоаналитика возможность методами перебора определить содержимое этого самого неполного блока, и этот факт означает снижение стойкости шифра. Кроме того, длина шифротекста при этом изменится, увеличившись до ближайшего целого, кратного 64 битам, что часто бывает нежелательным.

ГОСТ предписывает использовать режим простой замены исключительно для шифрования ключевых данных.

Гаммирование

Гаммирование решает обе упомянутые проблемы: во-первых, все элементы гаммы различны для реальных шифруемых массивов и, следовательно, результат зашифровывания даже двух одинаковых блоков в одном массиве данных будет различным. Во-вторых, хотя элементы гаммы и вырабатываются одинаковыми порциями в 64 бита, использоваться может и часть такого блока с размером, равным размеру шифруемого блока.

Гамма для этого режима получается следующим образом: с помощью некоторого алгоритмического рекуррентного генератора последовательности чисел (РГПЧ) вырабатываются 64-битовые блоки данных, которые далее подвергаются преобразованию по циклу $32-3$, то есть зашифровыванию в режиме простой замены, в результате получаются блоки гаммы. Алгоритмы зашифровывания и расшифровывания в режиме гаммирования идентичны.

РГПЧ, используемый для выработки гаммы, является рекуррентной функцией:

$$\Omega_{i+1} = f(\Omega_i),$$

где Ω_i – элементы рекуррентной последовательности, f – функция преобразования. Следовательно, неизбежно возникает вопрос о его инициали-

зации, то есть об элементе Ω_0. В действительности, этот элемент данных является параметром алгоритма для режимов гаммирования, на схемах он обозначен как S, и называется в криптографии синхропосылкой, а в ГОСТе – начальным заполнением одного из регистров шифрователя. Разработчики ГОСТа решили использовать для инициализации РГПЧ не непосредственно синхропосылку, а результат ее преобразования по циклу $32-3$: $\Omega_0 = Ц_{32-3}(S)$. Последовательность элементов, вырабатываемых РГПЧ, целиком зависит от его начального заполнения, то есть элементы этой последовательности являются функцией своего номера и начального заполнения РГПЧ: $\Omega_i = f_i(\Omega_0)$, где $f_i(X) = f(f_{i-1}(X))$, $f_0(X) = X$. С учетом преобразования по алгоритму простой замены добавляется еще и зависимость от ключа:

$$Г_i = Ц_{32-3}(\Omega_i) = Ц_{32-3}(f_i(\Omega_0)) = Ц_{32-3}(f_i(Ц_{32-3}(S))) = \varphi_i(S,K),$$

где $Г_i$ – i-тый элемент гаммы, K – ключ.

Таким образом, последовательность элементов гаммы для использования в режиме гаммирования однозначно определяется ключевыми данными и синхропосылкой. Естественно, для обратимости процедуры шифрования в процессах зашифровывания и расшифровывания должна использоваться одна и та же синхропосылка. Из требования уникальности гаммы, невыполнение которого приводит к катастрофическому снижению стойкости шифра, следует, что для шифрования двух различных массивов данных на одном ключе необходимо обеспечить использование различных синхропосылок. Это приводит к необходимости хранить или передавать синхропосылку по каналам связи вместе с зашифрованными данными, хотя в отдельных особых случаях она может быть предопределена или вычисляться особым образом, если исключается шифрование двух массивов на одном ключе.

Теперь рассмотрим РГПЧ, используемый в ГОСТе для генерации элементов гаммы. Прежде всего, надо отметить, что к нему не предъявляются требования обеспечения каких-либо статистических характеристик вырабатываемой последовательности чисел. РГПЧ спроектирован разработчиками ГОСТа исходя из необходимости выполнения следующих условий:

– период повторения последовательности чисел, вырабатываемой РГПЧ, не должен сильно (в процентном отношении) отличаться от максимально возможного при заданном размере блока значения 2^{64};

– соседние значения, вырабатываемые РГПЧ, должны отличаться друг от друга в каждом байте, иначе задача криптоаналитика будет упрощена;

– РГПЧ должен быть достаточно просто реализуем как аппаратно, так и программно на наиболее распространенных типах процессоров, большинство из которых, как известно, имеют разрядность 32 бита.

Исходя из перечисленных принципов создатели ГОСТа спроектировали РГПЧ, имеющий следующие характеристики:

– в 64-битовом блоке старшая и младшая части обрабатываются независимо друг от друга:

$$\Omega_i = (\Omega_i^0, \Omega_i^1), |\Omega_i^0| = |\Omega_i^1| = 32, \Omega_{i+1}^0 = \hat{f}(\Omega_i^0), \Omega_{i+1}^1 = \tilde{f}(\Omega_i^1)$$

фактически, существуют два независимых РГПЧ для старшей и младшей частей блока;

– рекуррентные соотношения для старшей и младшей частей следующие:

$$\Omega_{i+1}^0 = (\Omega_i^0 + C_1) \bmod 2^{32}, \text{ где } C_1 = 1010101_{16};$$

$$\Omega_{i+1}^1 = (\Omega_i^1 + C_2 - 1) \bmod (2^{32} - 1) + 1, \text{ где } C_2 = 1010104_{16}.$$

Нижний индекс в записи числа означает его систему счисления, таким образом, константы, используемые на данном шаге, записаны в 16-ричной системе счисления. Период повторения последовательности для младшей части составляет 2^{32}, для старшей части $2^{32} - 1$, для всей последовательности период составляет $2^{32} \cdot (2^{32} - 1)$.

Схема алгоритма шифрования в режиме гаммирования приведена на рис. 5.15, ниже изложены пояснения к схеме:

Шаг 0.

Определяет исходные данные для основного шага криптопреобразования:

– $T_{0(Ш)}$ – массив открытых (зашифрованных) данных произвольного размера, подвергаемый процедуре зашифровывания (расшифровывания), по ходу процедуры массив подвергается преобразованию порциями по 64 бита;

– S – синхропосылка, 64-битовый элемент данных, необходимый для инициализации генератора гаммы;

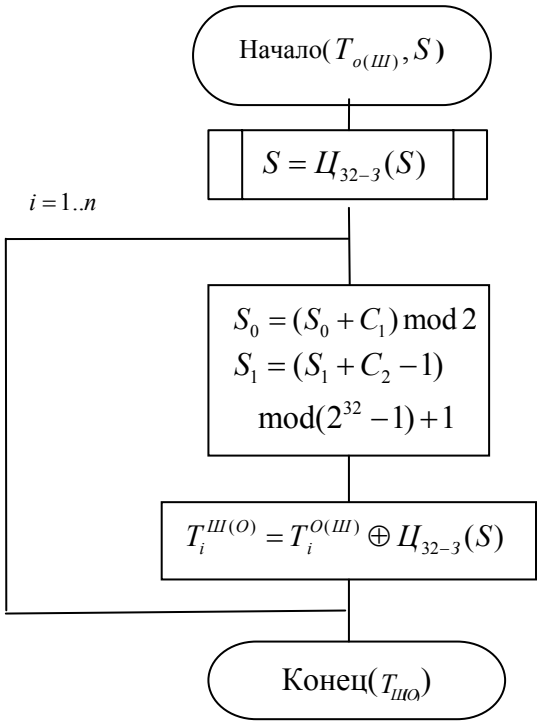


Рис. 5.15. Алгоритм зашифровывания (расшифровывания) данных в режиме гаммирования

Шаг 1.

Начальное преобразование синхропосылки, выполняемое для ее «рандомизации», то есть для устранения статистических закономерностей, присутствующих в ней, результат используется как начальное заполнение РГПЧ.

Шаг 2.

Один шаг работы РГПЧ, реализующий его рекуррентный алгоритм. В ходе данного шага старшая(S_1) и младшая (S_0) части последовательности данных вырабатываются независимо друг от друга.

Шаг 3.

Гаммирование. Очередной 64-битовый элемент, выработанный РГПЧ, подвергается процедуре зашифровывания по циклу $32 - З$, результат используется как элемент гаммы для зашифровывания (расшифровывания) очередного блока открытых (зашифрованных) данных того же размера.

Шаг 4.

Результат работы алгоритма – зашифрованный (расшифрованный) массив данных.

Ниже перечислены особенности гаммирования как режима шифрования.

1. Одинаковые блоки в открытом массиве данных дадут при зашифровывании различные блоки шифртекста, что позволит скрыть факт их идентичности.

2. Поскольку наложение гаммы выполняется побитно, шифрование неполного блока данных легко выполнимо как шифрование битов этого неполного блока, для чего используются соответствующие биты блока гаммы. Так, для зашифровывания неполного блока в 1 бит можно использовать любой бит из блока гаммы.

3. Синхропосылка, использованная при зашифровывании, каким-то образом должна быть передана для использования при расшифровывании. Это может быть достигнуто следующими путями:

– хранить или передавать синхропосылку вместе с зашифрованным массивом данных, что приведет к увеличению размера массива данных при зашифровывании на размер синхропосылки, то есть на 8 байт;

– использовать предопределенное значение синхропосылки или вырабатывать ее синхронно источником и приемником по определенному закону, в этом случае изменение размера передаваемого или хранимого массива данных отсутствует.

Оба способа дополняют друг друга, и в тех редких случаях, где не работает первый, наиболее употребительный из них, может быть использован второй. Второй способ имеет гораздо меньшее применение, поскольку сделать синхропосылку предопределенной можно только в том случае, если на данном комплекте ключевой информации шифруется заведомо не более одного массива данных, что бывает в редких случаях. Генерировать синхропосылку синхронно у источника и получателя массива данных также не всегда представляется возможным, поскольку требует жесткой привязки к чему-либо в системе.

В режиме гаммирования биты массива данных шифруются независимо друг от друга. Таким образом, каждый бит шифротекста зависит от соответствующего бита открытого текста и, естественно, порядкового номера бита в массиве: $t_i^{Ш} = t_i^{O} \oplus \gamma_i = f(t_i^{O}, i)$. Из этого вытекает, что изменение бита шифротекста на противоположное значение приведет к аналогичному изменению бита открытого текста на противоположный:

$$\overline{t}_i^{Ш} = t_i^{Ш} \oplus 1 = (t_i^{O} \oplus \gamma_1) \oplus 1 = (t_i^{o} \oplus 1) \oplus \gamma_i = \overline{t}_i^{O} \oplus \lambda_i,$$

где $\overline{t}$ обозначает инвертированное по отношению к t значение бита ($\overline{0} = 1, \overline{1} = 0$).

Данное свойство дает злоумышленнику возможность, воздействуя на биты шифротекста, вносить предсказуемые и даже целенаправленные изменения в соответствующий открытый текст, получаемый после его расшифровывания, не обладая при этом секретным ключом.

Гаммирование с обратной связью

Данный режим очень похож на режим гаммирования и отличается от него только способом выработки элементов гаммы – очередной элемент гаммы вырабатывается как результат преобразования по циклу 32-З предыдущего блока зашифрованных данных, а для зашифровывания первого блока массива данных элемент гаммы вырабатывается как результат преобразования по тому же циклу синхропосылки. Этим достигается зацепление блоков – каждый блок шифротекста в этом режиме зависит от соответствующего и всех предыдущих блоков открытого текста. Поэтому данный режим иногда называется гаммированием с зацеплением блоков. На стойкость шифра факт зацепления блоков не оказывает никакого влияния. Схема алгоритмов зашифровывания и расшифровывания в режиме гаммирования с обратной связью приведена на рис. 5.16.

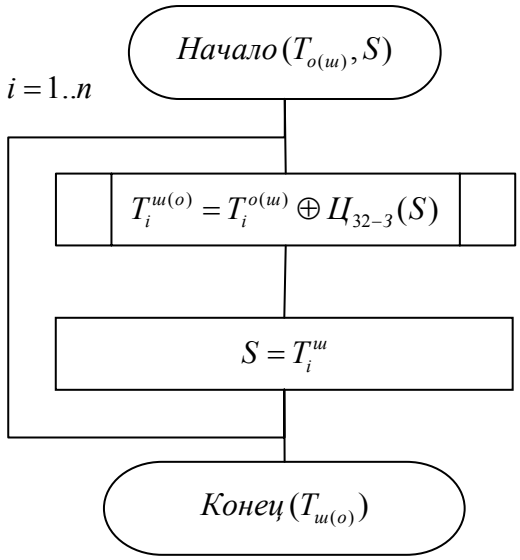


Рис. 5.16. Алгоритм зашифровывания (расшифровывания) данных в режиме гаммирования с обратной связью

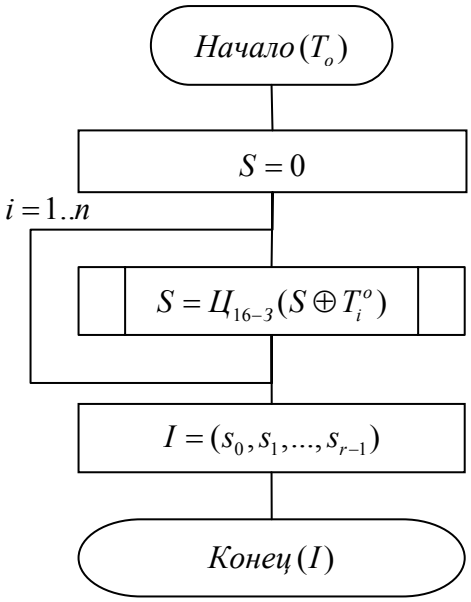


Рис. 5.17. Алгоритм выработки имитовставки для массива данных

Шифрование в режиме гаммирования с обратной связью обладает теми же особенностями, что и шифрование в режиме обычного гаммирования, за исключением влияния искажений шифротекста на соответствующий открытый текст. Для сравнения запишем функции расшифровывания блока для обоих упомянутых режимов:

$T_i^o = T_i^{ш} \oplus Г_i$, гаммирование;

$T_i^o = T_i^{ш} \oplus Ц_{32-З}(T_{i-1}^{ш})$, гаммирование с обратной связью.

Выработка имитовставки к массиву данных

Для решения задачи обнаружения искажений в зашифрованном массиве данных с заданной вероятностью в ГОСТе предусмотрен дополнительный режим криптографического преобразования – выработка имитовставки. Имитовставка – это контрольная комбинация, зависящая от открытых данных и секретной ключевой информации. Целью использования имитовставки является обнаружение всех случайных или преднамеренных изменений в массиве информации. Проблема, изложенная в предыдущем пункте, может быть успешно решена с помощью добавления к шифрованным данным имитовставки. Для потенциального злоумышленника две следующие задачи практически неразрешимы, если он не владеет ключевой информацией:

– вычисление имитовставки для заданного открытого массива информации;

– подбор открытых данных под заданную имитовставку.

Схема алгоритма выработки имитовставки приведена на рис. 5.17. В качестве имитовставки берется часть блока, полученного на выходе, обычно – 32 его младших бита.

В шифре ГОСТ используется 256-битовый ключ, и объем ключевого пространства составляет 2^{256}. Ни на одной из существующих в настоящее время или предполагаемых к реализации в недалеком будущем ЭВМ общего применения нельзя подобрать ключ за время, меньшее многих сотен лет. Стандарт проектировался с большим запасом и по стойкости на много порядков превосходит американский стандарт DES с его реальным размером ключа в 56 бит и объемом ключевого пространства всего 2^{56}.

5.6. Вопросы и задания для самопроверки

1. К какому типу криптосистем относится алгоритм DES?

2. Представьте обобщенную схему зашифровывания DES.

3.Сколько различных ключей существует в алгоритме DES?

4. Сколько циклов шифрования в алгоритме DES?

5. Представьте обобщенную схему расшифровывания DES.

6. Какого размера ключ использует функция шифрования алгоритма DES?

7. Для решения какой задачи используется расширитель при реализации функции шифрования?

8. Представьте структурную схему реализации функции шифрования алгоритма DES.

9. Сколько ключей необходимо при шифровании 64-битового блока алгоритмом DES?

10. Представьте схему алгоритма вычисления ключей для DES.

11. Поясните принцип работы блока замены алгоритма DES.

12. Какие существуют режимы работы алгоритма DES?

13. Какой размер блока открытого текста в алгоритме DES?

14. Каким методом можно повысить криптостойкость алгоритма DES?

15. К какому типу криптосистем относится алгоритм ГОСТ 28147-89?

16. Сколько различных ключей существует в алгоритме ГОСТ 28147-89?

17. Какие основные режимы (базовые циклы) работы предусматривает алгоритм ГОСТ 28147-89?

18. Какой размер имеет таблица замен в ГОСТ 28147-89?

19. Представьте схему основного шага криптопреобразования алгоритма ГОСТ 28147-89.

20. Представьте порядок использования ключевых элементов в циклах зашифровывания и расшифровывания ГОСТ 28147-89.

21. Представьте схему цикла зашифровывания (расшифровывания) алгоритма ГОСТ 28147-89.

22. Поясните отличие режима гаммирования от гаммирования с обратной связью в ГОСТ 28147-89.

23. Сколько циклов используется в алгоритме ГОСТ 28147-89 для зашифровывания и расшифровывания информации?

24. Какой номер подключа, используемого на 9-м цикле расшифровывания информации в алгоритме ГОСТ 28147-89?

25. Какие основные достоинства и недостатки симметричных криптосистем?

26. Приведите качественное сравнение алгоритмов шифрования DES и ГОСТ 28147-89 и обоснуйте свои выводы.

5.7. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 8
Симметричные системы шифрования DES и ГОСТ 28147-89

Теория для практического занятия представлена в модуле 5.

Перед выполнением тестовых заданий проводится опрос с использованием вопросов, представленных в разделе 5.6.

Тестовые задания:

1. Симметричной криптосистема называется потому, что:

a) ключ, используемый в процессе зашифровывания, симметричен ключу, используемому в процессе расшифровывания;

b) в процессе зашифровывания и расшифровывания используется один и тот же ключ;

c) шифротекст обладает внутренней симметрией;

d) в процессе зашифровывания и расшифровывания используются разные ключи.

2. Размер блока информации обрабатываемого алгоритмом DES равен:

a) 32 бита;

b) 56 бит;

c) 64 бита;

d) 128 бит.

3. Количество циклов в алгоритме DES равно:

a) 15;

b) 4;

c) 16;

d) 1.

4. Разрядность подключей K_i, используемых на каждом цикле алгоритма DES равна:

a) 32 бита;

b) 48 бит;

c) 64 бита;

d) 56 бит.

5. Количество различных ключей в алгоритме DES равно:

a) 56;

b) 2^{64};

c) 2^{56};

d) 2^{48}.

6. Размер блока информации обрабатываемого алгоритмом ГОСТ 28147-89 равен:

a) 48 бит;

b) 64 бита;

c) 128 бит;

d) 256 бит.

7. Количество различных ключей в алгоритме ГОСТ 28147-89 равно:

a) 2^{127};

b) 2^{128};

c) 2^{64};

d) 1152921504606846976.

8. Разрядность подключей K_i, используемых на каждом цикле алгоритма ГОСТ 28147-89, равна:

a) 32 бита;

b) 48 бит;

c) 64 бита;

d) 56 бит.

9. Количество циклов в алгоритме ГОСТ 28147-89 равно:

a) 15;

b) 32;

c) 16;

d) 64.

10. Номер подключа K_i, используемого на десятом цикле процедуры зашифровывания в алгоритме ГОСТ 28147-89, равен:

a) 1;

b) 2;

c) 6;

d) 0;

Задачи

1. Переведите число 3^{43} в двоичную систему счисления.

2. Пусть каждая из 16 первых букв русского алфавита (абвгдежзийклмноп) имеет четырехразрядный двоичный код, соответствующий ее номеру от 0 до 15, т.е. *а* – 0000_2, *б* – 0001_2, …, *п* – 1111_2. Составьте из этих букв произвольное сообщение состоящее из 32 букв, затем разбейте полученное сообщение на блоки длиной 64 бита. Значения полученных блоков запишите в десятичной системе счисления.

3. Найдите сумму по модулю 2 следующих пар чисел:

a) 224489301 и 28973675;

b) 3479913811 и 2301120149;

c) 3040958609 и 2781188359;

d) 3075166647 и 3785852425.

4. Найдите сумму по модулю 3 следующих пар чисел:

a) 3496 и 3718;

b) 3668 и 1419;

c) 5563 и 6482;

d) 6379 и 1215.

5. Найдите сумму по модулю 2^{32} следующих пар чисел:

a) 3037741847 и 1257225448;

b) 2706981523 и 1587985773;

c) 2597745569 и 1697221728;

d) 13862897145741766693.

6. Найдите 4-битовое число на выходе блока замены S_2 (функции шифрования алгоритма DES), если на вход подать число:

a) 15;

b) 62;

c) 23;

d) 45.

7. Найдите 64-битовое число на выходе блока перестановки *IP*, алгоритма DES, если на вход подано число:

a) 16175076002153763172;

b) 11870809655824700300;

c) 14436987034089088762;

d) 12727938706001950544.

МОДУЛЬ 6. АСИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

1. Построения систем с открытым ключом. Алгоритмы рюкзака.
2. Алгоритм RSA. Шифрование и дешифрование RSA.
3. Криптосистема Эль-Гамаля. Алгоритм Рабина. Комбинированный метод шифрования.
4. Вопросы и задания для самопроверки.
5. Практическое занятие № 9.

Цель модуля – изучение студентами ассиметричных криптосистем и особенностей их использования при криптографическом кодировании.

В результате изучения модуля студенты должны:

– знать процедуры зашифровывания и расшифровывания информации асимметричными алгоритмами RSA, Эль-Гамаля и Рабина;

– знать достоинства и недостатки асимметричных криптосистем;

– уметь использовать криптосистемы RSA, Эль-Гамаля и Рабина для шифрования информации;

– иметь представление об алгоритмах рюкзака, а также комбинированных методах шифрования.

6.1. Построения систем с открытым ключом. Алгоритмы рюкзака

6.1.1. Особенности и построение криптосистем с открытым ключом

Как бы ни были сложны и надежны криптографические системы, их слабое место при практической реализации – проблема распределения ключей. Для того чтобы был возможен обмен конфиденциальной информацией между двумя субъектами ИС, ключ должен быть сгенерирован одним из них, а затем каким-то образом опять же в конфиденциальном порядке передан другому. Т.е. в общем случае для передачи ключа опять же требуется использование какой-то криптосистемы.

Для решения этой проблемы на основе результатов, полученных классической и современной алгеброй, были предложены системы с открытым ключом. Можно сказать, что криптосистема с открытым ключом определяется тремя алгоритмами: генерации ключей, шифрования и расшифровывания. Алгоритм генерации ключей открыт, всякий может подать ему на вход случайную строку r надлежащей длины и получить пару ключей ($K(o)$, $K(c)$). Один из ключей (например, $K(o)$) публикуем, он называется

открытым, а второй $K(c)$, называемый секретным (закрытым), хранится в тайне. Алгоритмы шифрования $E_{k(0)}$ и расшифровывания $D_{k(c)}$ таковы, что для любого открытого текста M: $D_{k(c)}(E_{k(o)}(M)) = M$.

Открытый ключ доступен любому, кто желает послать сообщение адресату. Исходный текст шифруется открытым ключом адресата и передается ему. Зашифрованный текст, в принципе, не может быть расшифрован тем же открытым ключом.

Дешифрование сообщения возможно только с использованием закрытого ключа, который известен только самому адресату.

Криптографические системы с открытым ключом используют так называемые необратимые или односторонние функции, которые обладают следующим свойством: при заданном значении x относительно просто вычислить значение $f(x)$, однако если $y = f(x)$, то нет простого пути для вычисления значения x.

В самом определении необратимости присутствует неопределенность. Под необратимостью понимается не теоретическая необратимость, а практическая невозможность вычислить обратное значение, используя современные вычислительные средства за обозримый интервал времени.

Поэтому, чтобы гарантировать надежную защиту информации к системам с открытым ключом (СОК) предъявляются два важных и очевидных требования:

1) преобразование исходного текста должно быть необратимым и исключать его восстановление на основе открытого ключа;

2) определение закрытого ключа на основе открытого также должно быть невозможным на современном технологическом уровне. При этом желательна точная нижняя оценка сложности (количества операций) раскрытия шифра.

Алгоритмы шифрования с открытым ключом получили широкое распространение в современных информационных системах. Все предлагаемые сегодня криптосистемы с открытым ключом опираются на один из следующих типов необратимых преобразований:

– разложение больших чисел на простые множители;
– вычисление логарифма в конечном поле;
– вычисление корней алгебраических уравнений.

Здесь же следует отметить, что алгоритмы криптосистемы с открытым ключом можно использовать в трех назначениях:

– как самостоятельные средства защиты передаваемых и хранимых данных;

– как средства для распределения ключей;
– средства для цифровой подписи документов.

Концепция криптографии с открытыми ключами была выдвинута Уитфилдом Диффи (Whitfield Diffie) и Мартином Хеллманом (Martin Hellman), и независимо Ральфом Мерклом (Ralph Merkle). Их вкладом в криптографию было убеждение, что ключи можно использовать парами – ключ шифрования и ключ дешифрирования – и что может быть невозможно, получить один ключ из другого. Диффи и Хеллман впервые представили эту идею на Национальной компьютерной конференции (National Computer Conference) 1976 года. С 1976 года было предложено множество криптографических алгоритмов с открытыми ключами. Многие из них небезопасны. Из тех, которые являются безопасными, многие непригодны для практической реализации. Либо они используют слишком большой ключ, либо размер полученного шифротекста намного превышает размер открытого текста.

Немногие алгоритмы являются и безопасными, и практичными. Некоторые из этих безопасных и практичных алгоритмов подходят только для распределения ключей. Другие подходят для шифрования (и для распределения ключей). Третьи полезны только для цифровых подписей. Только три алгоритма хорошо работают как при шифровании, так и для цифровой подписи: RSA, EIGamal, Rabin. Все эти алгоритмы медленны. Они шифруют и дешифрируют данные намного медленнее, чем симметричные алгоритмы. Обычно их скорость недостаточна для шифрования больших объемов данных.

6.1.2. Сущность алгоритма рюкзака

Первым алгоритмом для обобщенного шифрования с открытым ключом стал алгоритм рюкзака, разработанный Ральфом Мерклом и Мартином Хеллманом. Он мог быть использован только для шифрования, хотя позднее Ади Шамир адаптировал систему для цифровой подписи. Безопасность алгоритмов рюкзака опирается на проблему рюкзака, NP-полную проблему. Хотя позже было обнаружено, что этот алгоритм небезопасен, его стоит изучить, так как он демонстрирует возможность применения NP-полной проблемы в криптографии с открытыми ключами.

Проблема рюкзака несложна. Дана куча предметов различной массы, можно ли положить некоторые из этих предметов в рюкзак так, чтобы масса рюкзака стала равна определенному значению? Более формально, дан набор значений M_1, M_2,. . . M_n и сумма S, вычислить значения b_i, такие что

$$S = b_1M_1 + b_2M_2 + ... + b_nM_n,$$

b_i может быть либо нулем, либо единицей. Единица показывает, что предмет кладут в рюкзак, а ноль – что не кладут.

Например, массы предметов могут иметь значения 1, 5, 6, 11, 14 и 20. Вы можете упаковать рюкзак так, чтобы его масса стала равна 22, использовав массы 5, 6 и 11. Невозможно упаковать рюкзак так, чтобы его масса была равна 24. В общем случае время, необходимое для решения этой проблемы, с ростом количества предметов в куче растет экспоненциально.

В основе алгоритма рюкзака Меркла – Хеллмана лежит идея шифровать сообщение как решение набора проблем рюкзака. Предметы из кучи выбираются с помощью блока открытого текста, по длине равного количеству предметов в куче (биты открытого текста соответствуют значениям *b*), а шифротекст является полученной суммой. Пример шифротекста, зашифрованного с помощью проблемы рюкзака, показан ниже:

Открытый текст	1 1 1 0 0 1	0 1 0 1 1 0	0 0 0 0 0 0	0 1 1 0 0 0
Рюкзак	1 5 6 11 14 20	1 5 6 11 14 20	1 5 6 11 14 20	1 5 6 11 14 20
Шифротекст	1+5+6+20=32	5+11+14=30	0=0	5+6=11

На самом деле существуют две различные проблемы рюкзака, одна решается за линейное время, а другая, как считается, – нет. Легкую проблему можно превратить в трудную. Открытый ключ представляет собой трудную проблему, которую легко использовать для шифрования, но невозможно для дешифрирования сообщений.

Закрытый ключ является легкой проблемой, давая простой способ дешифрировать сообщения. Тому, кто не знает закрытый ключ, придется попытаться решить трудную проблему рюкзака.

6.1.3. Сверхвозрастающие рюкзаки

Что такое легкая проблема рюкзака? Если перечень масс представляет собой сверхвозрастающую последовательность, то полученную проблему рюкзака легко решить. Сверхвозрастающая последовательность – это последовательность, в которой каждой член больше суммы всех предыдущих членов. Например, последовательность {1, 3, 6, 13, 27, 52} является сверхвозрастающей, а {1, 3, 4, 9, 15, 25} – нет.

Решение сверхвозрастающего рюкзака найти легко. Возьмите полный вес и сравните его с самым большим числом последовательности. Если полный вес меньше, чем это число, то его не кладут в рюкзак. Если полный вес больше или равен этому числу, то оно кладется в рюкзак.

Уменьшим массу рюкзака на это значение и перейдем к следующему по величине числу последовательности. Будем повторять, пока процесс не закончится. Если полный вес уменьшится до нуля, то решение найдено.

Например, пусть полный вес рюкзака – 70, а последовательность весов {2, 3, 6, 13, 27, 52}. Самый большой вес, 52, меньше 70, поэтому кладем 52 в рюкзак. Вычитая 52 из 70, получаем 18. Следующий вес, 27, больше 18, поэтому 27 в рюкзак не кладется, вес 13 меньше 18, поэтому кладем 13 в рюкзак. Вычитая 13 из 18, получаем 5. Следующий вес, 6, больше 5, поэтому 6 не кладется в рюкзак. Продолжение этого процесса покажет, что и 2, и 3 кладутся в рюкзак, и полный вес уменьшается до 0, что сообщает о найденном решении. Если бы это был блок шифрования методом рюкзака Меркла – Хеллмана, открытый текст, полученный из значения шифротекста 70, был бы равен 110101.

Не сверхвозрастающие, или нормальные, рюкзаки представляют собой трудную проблему – быстрого алгоритма для них не найдено. Единственно известным способом определить, какие предметы кладутся в рюкзак, является методическая проверка возможных решений, пока вы не наткнетесь на правильное. Самый быстрый алгоритм, принимая во внимание различную эвристику, имеет экспоненциальную зависимость от числа возможных предметов. Добавьте к последовательности весов еще один член, и найти решение станет вдвое труднее. Это намного труднее сверхвозрастающего рюкзака, где, если вы добавите один предмет к последовательности, поиск решения увеличится на одну операцию.

Алгоритм Меркла – Хеллмана основан на этом свойстве. Закрытый ключ является последовательностью весов проблемы сверхвозрастающего рюкзака. Открытый ключ – это последовательность весов проблемы нормального рюкзака с тем же решением. Меркл и Хеллман, используя модульную арифметику, разработали способ преобразования проблемы сверхвозрастающего рюкзака в проблему нормального рюкзака.

6.1.4. Создание открытого ключа из закрытого

Чтобы получить нормальную последовательность рюкзака, возьмем сверхвозрастающую последовательность рюкзака, например, {2, 3, 6, 13, 27, 52}, и умножим по модулю m все значения на число n. Значение модуля должно быть больше суммы всех чисел последовательности, например, 105. Множитель должен быть взаимно простым числом с модулем, например, 31. Нормальная последовательность рюкзака имеет вид:

$2 \times 31 \bmod 105 = 62$
$3 \times 31 \bmod 105 = 93$
$6 \times 31 \bmod 105 = 81$
$13 \times 31 \bmod 105 = 88$
$27 \times 31 \bmod 105 = 102$
$52 \times 31 \bmod 105 = 37$.

Итого – {62, 93, 81, 88, 102, 37}.

Сверхвозрастающая последовательность рюкзака является закрытым ключом, а нормальная последовательность рюкзака – открытым.

6.2. Алгоритм RSA. Шифрование и дешифрирования RSA

6.2.1. Разработка алгоритма

Из всех предложенных за эти годы алгоритмов с открытыми ключами RSA проще всего реализовать и этот алгоритм многие годы противостоит интенсивному криптоанализу. Хотя криптоанализ ни доказал, ни опроверг безопасность RSA, он, по сути, обосновывает уровень доверия к алгоритму. Безопасность RSA основана на трудности разложения на множители больших чисел. Открытый и закрытый ключи являются функциями двух больших (100 – 200 десятичных разрядов или даже больше) простых чисел. Предполагается, что восстановление открытого текста по шифротексту и открытому ключу эквивалентно разложению на множители двух больших чисел.

Несмотря на довольно большое число различных СОК, наиболее популярна криптосистема RSA, разработанная в 1977 году и получившая название в честь ее создателей: Рона Ривеста (Rivest), Ади Шамира (Shamir) и Леонарда Адлемана (Adleman).

Они воспользовались тем фактом, что нахождение больших простых чисел в вычислительном отношении осуществляется легко, но разложение на множители произведения двух таких чисел практически невыполнимо. Доказано (теорема Рабина), что раскрытие шифра RSA эквивалентно такому разложению. Поэтому для любой длины ключа можно дать нижнюю оценку числа операций для раскрытия шифра, а с учетом производительности современных компьютеров оценить и необходимое на это время. Время выполнения наилучших из известных алгоритмов разложения при $n > 10^{145}$ на сегодняшний день выходит за пределы современных технологических возможностей.

Возможность гарантированно оценить защищенность алгоритма RSA стала одной из причин популярности этой СОК на фоне десятков других схем. Поэтому алгоритм RSA используется в банковских компьютерных сетях, особенно для работы с удаленными клиентами (обслуживание кредитных карточек).

6.2.2. Математическая база алгоритма

Малая теорема Ферма

Если p – простое число, то

$$x^{p-1} = 1 \pmod{p}$$

для любого x, простого относительно p.

Функцией Эйлера $\varphi(n)$ называется число положительных целых, меньших n и простых относительно n.

Если $n = pq$, (p и q – отличные друг от друга простые числа), то

$$\varphi(n) = (p-1)(q-1).$$

Если $n = pq$, (p и q – отличные друг от друга простые числа) и x – простое относительно p и q, то

$$x^{\varphi(n)} = 1 (\mathrm{mod}\ n).$$

Следствие. Если $n = pq$, (p и q – отличные друг от друга простые числа) и e – простое относительно $\varphi(n)$, то отображение

$$E_{e,n}: x \to x^e \pmod{n}$$

является взаимно однозначным на Z_n.

Если e – простое относительно $\varphi(n)$, то существует целое d, такое, что

$$ed = 1 \pmod{\varphi(n)}$$

6.2.3. Шифрование и дешифрирования RSA

Действия получателя криптограммы B:

1. Получатель B генерирует два произвольных больших простых числа p и q. Эти числа должны быть примерно одинаковыми, размерностью 100 – 200 десятичных разрядов. Они должны быть секретными.

2. Получатель B вычисляет значение модуля $n = p \cdot q$ и функции Эйлера $\varphi(n) = (p-1)\cdot(q-1)$ и выбирает значение открытого ключа K_0 с соблюдением условий: $1 < K_0 \le \varphi(n)$, $(K_0, \varphi(n)) = 1$, т.е. K_0 и $\varphi(n)$ должны быть взаимно простыми.

3. Получатель B вычисляет значение секретного ключа K_C (обратного числа к числу K_0 по модулю $\varphi(n)$):

$$K_C = \left(K_0^{-1}\right) \bmod \varphi(n).$$

4. В посылает A пару чисел n, K_0 по открытому каналу.

Действия отправителя криптограммы A:

1. Разбивает исходный текст M на блоки M_i, $i = 1,2,\ldots,m$, т.е. $M = M_1, M_2, \ldots, M_m$. Величина $M_i < n$. Т.е. если p и q 100-разрядные простые числа, то n будет содержать около 200 разрядов, и каждый блок сообщения m должен быть около 200 разрядов в длину. Если нужно зашифровать фиксированное число блоков, их можно дополнить несколькими нулями слева, чтобы гарантировать, что блоки всегда будут меньше n.

2. Шифрует каждое число M_i по формуле $C_i = \left(M_i^{K_0}\right) \bmod n$ и отправляет криптограмму $C = C_1, C_2, \ldots, C_m$.

Получатель B, получив криптограмму, расшифровывает каждый блок секретным ключом K_C, $M_i = \left(C_i^{K_C}\right) \bmod n$, и восстанавливает весь текст $M = M_1, M_2, \ldots, M_m$.

Пример. Шифрование сообщения «CAB»

Для простоты вычислений будут использоваться небольшие числа.

Действия получателя B:

1. Выбирает $p = 3$ и $q = 11$.

2. Вычисляет модуль $n = p \cdot q = 3 \cdot 11 = 33$.

3. Вычисляет значение функции Эйлера для $N = 33$:

$\varphi(n) = \varphi(33) = (p - 1)(g - 1) = 2 \cdot 10 = 20$.

Выбирает в качестве открытого ключа K_0 произвольное число с учетом выполнения условий: $1 < K_0 \le 20$, НОД $(K_0, 20) = 1$. Пусть $K_0 = 7$.

4. Вычисляет значение секретного ключа K_C, используя расширенный алгоритм Евклида при сравнении $K_C \equiv 7^{-1} \pmod{20}$. Решение дает $K_C = 3$.

5. Пересылает A пару чисел ($n = 33$, $K_0 = 7$).

Действия получателя криптограммы A:

6. Представляет шифруемое сообщение как последовательность целых чисел. Пусть буква A представляется как число 1, буква B – как число 2, буква C – как число 3. Тогда сообщение CAB можно представить как последовательность чисел 312, т.е. $M_1 = 3$, $M_2 = 1$, $M_3 = 2$.

7. Шифрует текст, представленный в виде последовательности чисел M_1, M_2, M_3, используя ключ $K_0 = 7$, и $N = 33$, по формуле $C_i = M_i^{K_0} \pmod{N} = M_i^7 \pmod{33}$.

Получает:

$C_1 = 3^7 \pmod{33} = 2187 \pmod{33} = 9$,

$C_2 = 1^7 \pmod{33} = 1 \pmod{33} = 1$,

$C_3 = 2^7 \pmod{33} = 128 \pmod{33} = 29$.

Отправляет *B* криптограмму $C_1, C_2, C_3 = 9, 1, 29$.

Действия *B*:

8. Расшифровывает принятую криптограмму C_1, C_2, C_3, используя секретный ключ $K_C = 3$, по формуле $M_i = C_i^{K_C} (\mathrm{mod}\, N) = C_i^3 (\mathrm{mod}\, 33)$.

Получает:

$M_1 = 9^3 \pmod{33} = 729 \pmod{33} = 3$,

$M_2 = 1^3 \pmod{33} = 1 \pmod{33} = 1$,

$M_3 = 29^3 \pmod{33} = 24389 \pmod{33} = 2$.

Таким образом, восстановлено исходное сообщение: CAB

Существует вариант криптосистемы RSA, в которой вместо функции Эйлера используется функция Кармайкла λ, где $\lambda(n)$ – наименьшее целое *t*, такое что для любого целого *x*, взаимно простого с *n*, выполняется $x^t = 1 \bmod n$. Если *n* выбирается так, как описано выше, то $\lambda(n) = \mathrm{НОК}\,(P - 1, Q - 1)$.

6.2.4. Аспекты практической реализации и безопасности

Покажем, что при расшифровывании восстанавливается исходный текст. Согласно обобщению Эйлером малой теоремы Ферма: если НОД $(a, n) = 1$ и $a^{\varphi(n)+1} \equiv a \bmod n$. Открытый K_0 и закрытый K_C ключи в алгоритме связаны соотношением $K_0 \cdot K_C \equiv 1 \bmod \varphi(n)$, или $K_0 \cdot K_C = k \cdot \varphi(n) + 1$ для некоторого целого k. Таким образом, процесс шифрования, а затем расшифровывания некоторого сообщения M_i выглядит следующим образом:

$$((M_i^{K_0}) \bmod n)^{K_C} \bmod n = (M_i^{K_0 \cdot K_C}) \bmod n = (M_i^{k \cdot \varphi(n)+1}) \bmod n = M_i.$$

В процессе применения RSA злоумышленник может иметь: C_i, K_0, n – и организовать дешифрирование двумя способами:

1. По C_i, K_0, n получить M_i. Для этого он решает задачу вычисления M_i из уравнения $C_i = M_i^{K_0} \bmod n$. Эта задача вычислительно трудна.

2. По n вычислить P, Q, затем найти $\varphi(n)$ и вычислить $K_C = \left(K_0^{-1}\right) \bmod \varphi(n)$ и дешифрировать сообщение $M_i = C_i^{K_C} \bmod n$.

Однако задача разложения большого числа на простые множители вычислительно сложна.

Пользователи A и B должны быстро осуществлять все вычисления: вычислять K_0, шифровать и расшифровывать.

Вычисление K_0 с использованием алгоритма Евклида - довольно быстрый процесс и не представляет трудности. Зашифровывание и расшифровывание – возведение большого числа в большую степень – требует определенных затрат времени, но, с учетом наличия быстрых алгоритмов и быстродействия современных компьютеров, это приемлемая процедура.

В настоящее время алгоритм RSA активно реализуется как в виде самостоятельных криптографических продуктов, так и в качестве встроенных средств.

Важная проблема практической реализации – генерация больших простых чисел. Решение задачи «в лоб» – генерация случайного большого числа n (нечетного) и проверка его делимости на множители. В случае неуспеха следует взять $n + 2$ и так далее.

Другая проблема – ключи какой длины следует использовать? В 1994 г. было факторизовано число со 129 десятичными цифрами. Это удалось осуществить математикам А. Ленстра и М. Манасси посредством организации распределенных вычислений на 1600 компьютерах, объединенных сетью, в течение восьми месяцев. По мнению А. Ленстра и М. Манасси, их работа компрометирует криптосистемы RSA и создает большую угрозу их дальнейшим применениям. Теперь разработчикам криптоалгоритмов с открытым ключом на базе RSA приходится избегать применения чисел длиной менее 200 десятичных разрядов. Самые последние публикации предлагают применять для этого числа длиной не менее 250 – 300 десятичных разрядов.

Была сделана попытка расчета оценок безопасных длин ключей асимметричных криптосистем на ближайшие 20 лет, исходя из прогноза развития компьютеров и их вычислительной мощности, а также возможного совершенствования алгоритмов факторизации. Эти оценки (табл. 6.1) даны для трех групп пользователей (индивидуальных пользователей, корпораций и государственных организаций), в соответствии с различием требований к их информационной безопасности. Конечно, данные оценки следует рассматривать как сугубо приблизительные, как возможную тенденцию изменений безопасных длин ключей асимметричных криптосистем со временем.

Таблица 6.1

Оценка длин ключей для асимметричных криптосистем, бит

Год	Отдельные пользователи	Корпорации	Государственные организации
1995	768	1280	1536
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

Третий немаловажный аспект реализации RSA – вычислительный. Ведь приходится использовать аппарат длинной арифметики. Если используется ключ длиной k бит, то для операций по открытому ключу требуется O (k^2) операций, по закрытому ключу – O (k^3) операций, а для генерации новых ключей требуется O (k^4) операций. Самая «быстрая» аппаратная реализация обеспечивает скорости в 100 раз больше компьютерной. По сравнению с тем же алгоритмом DES, RSA требует в тысячи раз большее время.

Алгоритм RSA входит в стандарт шифрования ISO9796. Сам алгоритм RSA запатентован только в США. Французское и австралийское банковские сообщества приняли RSA в качестве стандарта.

6.3. Криптосистема Эль-Гамаля. Алгоритм Рабина. Комбинированный метод шифрования

6.3.1. Криптосистема Эль-Гамаля

Схема Эль-Гамаля, предложенная в 1985 г., может быть использована как для шифрования, так и для цифровых подписей. Данная система является альтернативой RSA и при равном значении ключа обеспечивает ту же криптостойкость. В отличие от RSA метод Эль-Гамаля основан на проблеме дискретного логарифма. Если возводить число в степень в конечном поле достаточно легко, то восстановить аргумент по значению (то есть найти логарифм) довольно трудно.

Множество параметров системы включает простое число p и целое число g, степени которого по модулю p порождают большое число элементов Z_p.

Для того чтобы генерировать пару ключей (открытый ключ – секретный ключ), сначала выбирают некоторое большое простое число P и большое целое число G, причем $G < P$. Числа P и G могут быть распространены среди группы пользователей. Затем выбирают случайное целое число X, причем $X < P$. Число X является секретным ключом и должно

храниться в секрете. Далее вычисляют $Y = G^X \bmod P$. Число Y является открытым ключом.

Для того чтобы зашифровать сообщение M, выбирают случайное целое число $1 < K < P - 1$ такое, что числа K и $(P - 1)$ являются взаимно простыми. Затем вычисляют числа $a = G^K \bmod P$, $b = Y^K M \bmod P$. Пара чисел (a, b) является шифротекстом. Заметим, что длина шифротекста вдвое больше длины исходного открытого текста M.

Для того чтобы расшифровать шифротекст (a, b), вычисляют

$$M = b/a^X \bmod P.$$

Поскольку $a^X \equiv G^{KX} \bmod P$, $b/a^X \equiv Y^K M/a^X \equiv G^{KX} M/G^{KX} \equiv M \pmod P$, то соотношение справедливо.

Пример. Выберем $P = 11$, $G = 2$, секретный ключ $X = 8$

Вычисляем $Y = G^X \bmod P = 2^8 \bmod 11 = 256 \bmod 11 = 3$.

Итак, открытый ключ $Y = 3$.

Пусть сообщение $M = 5$. Выберем некоторое случайное число $K = 9$. Убедимся, что НОД $(K, P - 1) = 1$. Действительно, НОД (9, 10) = 1. Вычисляем пару чисел a и b: $a = G^K \bmod P = 2^9 \bmod 11 = 512 \bmod 11 = 6$, $b = Y^K M \bmod P = 3^9 \cdot 5 \bmod 11 = 19683 \cdot 5 \bmod 11 = 9$. Получим шифротекст $(a, b) = (6, 9)$.

Выполним расшифровывание этого шифротекста. Вычисляем сообщение M, используя секретный ключ X: $M = b/a^X \bmod P = 9/6^8 \bmod 11$. Выражение $M \equiv 9/6^8 \bmod 11$ можно представить в виде $6^8 \cdot M \equiv 9 \bmod 11$ или $1679616 \cdot M \equiv 9 \bmod 11$. Решая данное сравнение, находим $M = 5$.

В реальных схемах шифрования необходимо использовать в качестве модуля P большое целое простое число, имеющее в двоичном представлении длину 512…1024 бит.

6.3.2. Алгоритм Рабина

Безопасность схемы Рабина (Rabin) опирается на сложность поиска квадратных корней по модулю составного числа. Эта проблема аналогична разложению на множители. Рассмотрим одну из реализаций этой схемы.

Сначала выбираются два простых числа p и q, конгруэнтных $3 \bmod 4$. Эти простые числа являются закрытым ключом, а их произведение $n = pq$ – открытым ключом.

Для шифрования сообщения M (M должно быть меньше n), просто вычисляется

$$C = M^2 \bmod n.$$

Дешифрирование сообщения также несложно. Так как получатель знает p и q, он может решить две конгруэнтности с помощью китайской теоремы об остатках. Вычисляется

$$m_1 = C^{(p+1)/4} \bmod p$$
$$m_2 = (p - C^{(p+1)/4}) \bmod p$$
$$m_3 = C^{(q+1)/4} \bmod q$$
$$m_4 = (q - C^{(q+1)/4}) \bmod q$$

Затем выбираются целые числа $a = q(q^{-1} \bmod p)$ и $b = p(p^{-1} \bmod q)$. Четырьмя возможными решениями являются:

$$M_1 = (am_1 + bm_3) \bmod n$$
$$M_2 = (am_1 + bm_4) \bmod n$$
$$M_3 = (am_2 + bm_3) \bmod n$$
$$M_4 = (am_2 + bm_4) \bmod n$$

Один из четырех результатов, M_1, M_2, M_3, M_4, равно M. Если сообщение написано по-английски, выбрать правильное M нетрудно. С другой стороны, если сообщение является потоком случайных битов (скажем, для генерации ключей или цифровой подписи), способа определить, какое M правильное, нет. Одним из способов решить эту проблему служит добавление к сообщению перед шифрованием известного заголовка.

Хью Вильямс (Hugh Williams) переопределил схему Рабина, чтобы устранить эти недостатки. В его схеме p и q выбираются так, чтобы

$$p \equiv 3 \bmod 8$$
$$q \equiv 7 \bmod 8$$
$$N = pq$$

Кроме того, используется небольшое целое число S, для которого $J(S,N) = -1$ (J – это символ Якоби). N и S опубликовываются. Секретным ключом является k, для которого

$$k = 1/2(1/4(p-1)(q-1)+1)$$

Для шифрования сообщения M вычисляется c_1 такое, что $J(M,N) = (-1)^{c_1}$. Затем вычисляется $M' = (S^{c_1} * M) \bmod N$. Как и в схеме Рабина, $C = M'^2 \bmod N$. И $c_2 = M' \bmod 2$. Окончательным шифротекстом сообщения является тройка:

$$(C, c_1, c_2).$$

Для дешифрирования C получатель вычисляет M'' с помощью

$$C^k \equiv \pm M'' (\bmod N).$$

Правильный знак M'' определяет c_2. Наконец

$$M = (S^{c_1} * (-1)^{c_1} * M'') \bmod N .$$

6.3.3. Комбинированный метод шифрования

Главным достоинством криптосистем с открытым ключом является их потенциально высокая безопасность: нет необходимости ни передавать, ни сообщать кому бы то ни было значения секретных ключей, ни убеждаться в их подлинности. В симметричных криптосистемах существует опасность раскрытия секретного ключа во время передачи. Однако алгоритмы, лежащие в основе криптосистем с открытым ключом, имеют следующие недостатки:

– генерация новых секретных и открытых ключей основана на генерации новых больших простых чисел, а проверка простоты чисел занимает много процессорного времени;

– процедуры зашифровывания и расшифровывания, связанные с возведением в степень многозначного числа, достаточно громоздки.

Поэтому быстродействие криптосистем с открытым ключом обычно в сотни и более раз меньше быстродействия симметричных криптосистем с секретным ключом.

Комбинированный (гибридный) метод шифрования позволяет сочетать преимущества высокой секретности, присущие асимметричным криптосистемам с открытым ключом, с преимуществами высокой скорости работы, присущими симметричным криптосистемам с секретным ключом. При таком подходе криптосистема с открытым ключом применяется для зашифровывания, передачи и последующего расшифровывания только секретного ключа симметричной криптосистемы. А симметричная криптосистема применяется для зашифровывания и передачи исходного открытого текста. В результате криптосистема с открытым ключом не заменяет симметричную криптосистему с секретным ключом, а лишь дополняет ее, позволяя повысить в целом защищенность передаваемой информации. Такой подход иногда называют схемой электронного цифрового конверта.

Если пользователь *A* хочет передать зашифрованное комбинированным методом сообщение *M* пользователю *B*, то порядок его действий будет таков:

1. Создать (например, сгенерировать случайным образом) симметричный ключ, называемый в этом методе сеансовым ключом K_S.

2. Зашифровать сообщение *M* на сеансовом ключе K_S.

3. Зашифровать сеансовый ключ K_S на открытом ключе K_0 пользователя *B*.

4. Передать по открытому каналу связи в адрес пользователя *B* зашифрованное сообщение вместе с зашифрованным сеансовым ключом.

Действия пользователя *B* при получении зашифрованного сообщения и зашифрованного сеансового ключа должны быть обратными.

5. Расшифровать на своем секретном ключе K_C сеансовый ключ K_S.

6. С помощью полученного сеансового ключа K_S расшифровать и прочитать сообщение *M*.

При использовании комбинированного метода шифрования можно быть уверенным в том, что только пользователь *B* сможет правильно расшифровать ключ K_S и прочитать сообщение *M*.

Таким образом, при комбинированном методе шифрования применяются криптографические ключи как симметричных, так и асимметричных криптосистем. Очевидно, выбор длин ключей для каждого типа криптосистемы следует осуществлять таким образом, чтобы злоумышленнику было одинаково трудно атаковать любой механизм защиты комбинированной криптосистемы. Структурная схема комбинированной системы представлена на рис. 6.1.

В табл. 6.2 приведены распространенные длины ключей симметричных и асимметричных криптосистем, для которых трудность атаки полного перебора примерно равна трудности факторизации соответствующих модулей асимметричных криптосистем.

Таблица 6.2

Длины ключей для симметричных и асимметричных криптосистем при одинаковой их криптостойкости

Длина ключа симметричной криптосистемы, бит	Длина ключа асимметричной криптосистемы, бит
56	384
64	512
80	768
112	1792
128	2304

Комбинированный метод шифрования является наиболее рациональным, объединяя в себе высокое быстродействие симметричного шифрования и высокую криптостойкость, гарантируемую системами с открытым ключом.

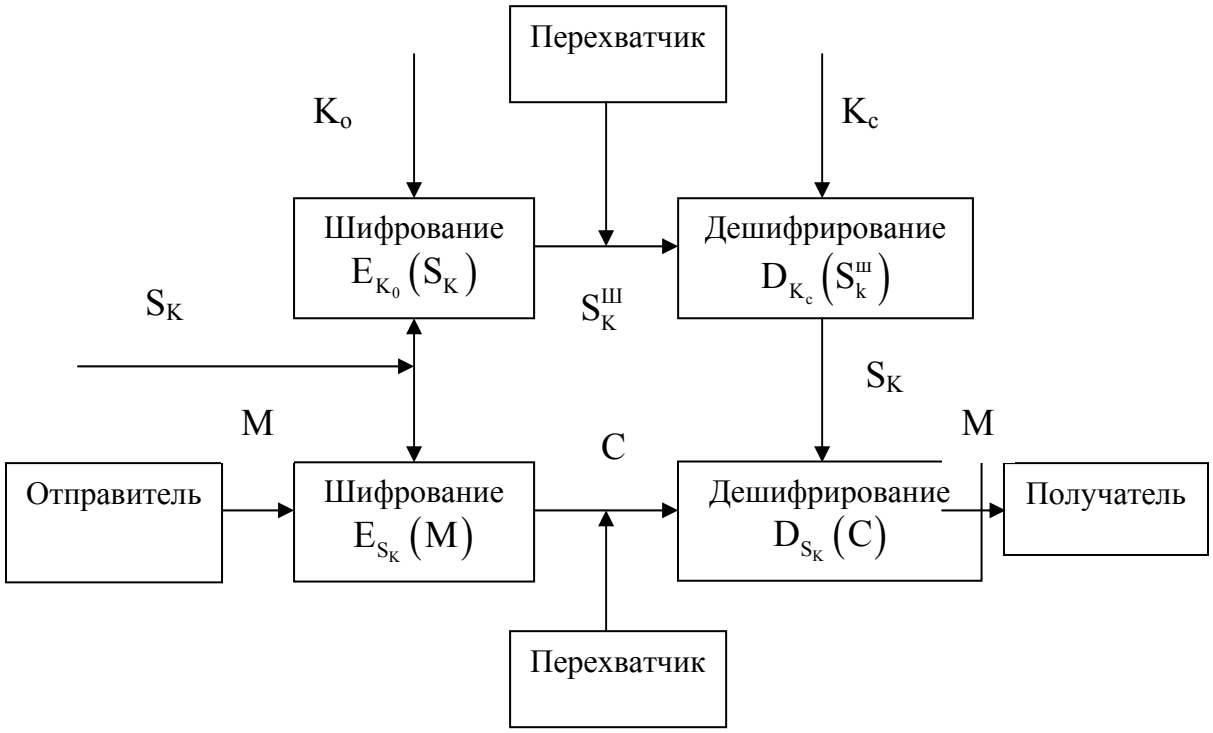


Рис. 6.1. Схема комбинированного метода шифрования

6.4. Вопросы и задания для самопроверки

1. Какие требования предъявляются к асимметричным криптосистемам?
2. В чем заключается сущность алгоритма рюкзака?
3. Как получить нормальную последовательность из сверхвозрастающей?
4. Каким образом формируются открытый и закрытый ключи в алгоритме рюкзака?
5. Приведите пример шифрования и дешифрирования информации с помощью алгоритма рюкзака.
6. Кто создал алгоритм RSA?
7. Какой длины должен использоваться ключ в алгориме RSA?
8. Сформулируйте малую теорему Ферма.
9. Дайте определение функции Эйлера.
10. Как находится модуль *n* в криптосистеме RSA?
11. На чем основана криптостойкость RSA?
12. Как связаны открытый и секретный ключи в алгоритме RSA?
13. Сформулируйте последовательность действий получателя и отправителя сообщения при использовании алгоритма RSA.
14. Приведите пример шифрования сообщения алгоритмом RSA.
15. Какими способами противник может организовать дешифрирование RSA?

16. Какие размеры модуля n рекомендуется использовать в алгоритме RSA?

17. Что такое дискретный логарифм?

18. Как соотносится размер шифротекста и открытого текста в алгоритме Эль-Гамаля?

19. В чем заключается недостаток алгоритма Эль-Гамаля?

20. Приведите пример шифрования алгоримом Эль-Гамаля.

21. На чем основана безопасность алгоритма Рабина?

22. В чем сущность модификации алгоритма Рабина?

23. Приведите пример шифрования алгоримом Рабина и его модификацией?

24. Что такое цифровой конверт?

25. Что Вы понимаете под комбинированным методом шифрования?

26. Сформулируйте алгоритм работы комбинированной системы шифрования.

27. Представьте структурную схему комбинированной криптосистемы.

28. Каковы преимущества комбинированных систем шифрования?

6.5. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 9
Ассиметричные системы шифрования

Теория для практического занятия представлена в модуле 6.

Перед выполнением тестовых заданий проводится опрос с использованием вопросов, представленных в разделе 6.4.

1. Асимметричной криптосистема называется потому, что:

a) в процессе зашифровывания и расшифровывания используются разные ключи;

b) процесс зашифровывания непохож на процесс расшифровывания;

c) в процессе зашифровывания и расшифровывания используется один и тот же ключ;

d) шифротекст не обладает внутренней симметрией.

2. Функция Эйлера – $\varphi(n)$ находит:

a) количество чисел больше n, которые являются взаимно простыми с n;

b) количество простых чисел меньше n;

c) количество чисел меньше n, которые не являются взаимно простыми с n;

d) количество простых чисел больше n;

e) количество чисел меньше n, которые являются взаимно простыми с n.

3. Какой тип необратимых преобразований используют агоритмы шифрования с открытым ключом?

a) разложение больших чисел на простые множители;

b) разложение простых чисел на множители;

c) вычисление логарифма в конечном поле;

d) вычисление корней алгебраических уравнений.

4. Алгоритмы криптосистемы с открытым ключом можно использовать:

a) как самостоятельные средства защиты передаваемых и хранимых данных;

b) как средства для распределения ключей;

c) как средства для сжатия информации;

в) как средства для цифровой подписи документов.

5. Как связаны открытый и секретный ключи в алгоритме RSA?

a) являются обратными числами по модулю n;

b) никак не связаны;

c) их произведение равно n;

d) являются обратными числами по модулю $\varphi(n)$.

6. Как связаны открытый и секретный ключи в алгоритме Эль-Гамаля?

a) являются обратными числами;

b) функцией дискретного логарифма;

c) никак не связаны;

d) функцией sin.

7. Как связаны открытый и секретный ключи в алгоритме Рабина?

a) являются обратными числами по модулю n;

b) никак не связаны;

c) открытый ключ равен произведению закрытых;

d) являются обратными числами по модулю $\varphi(n)$.

8. Алгоритмы, лежащие в основе криптосистем с открытым ключом, имеют недостатки:

a) генерация новых секретных и открытых ключей основана на генерации новых больших простых чисел, а проверка простоты чисел занимает много процессорного времени;

b) невозможно оценить криптостойкость ассимметричных систем;

c) процедуры зашифровывания и расшифровывания, связанные с возведением в степень многозначного числа, достаточно громоздки;

d) низкая криптостойкость по сравнению с симметричными системами.

9. Какое количество ключей исползуется в комбинированной системе шифрования?

a) 2;

b) 3;

c) 4;

d) 5.

10. В комбинированной системе шифрования по открытому каналу связи передается:

a) зашифрованный открытый ключ;

b) зашифрованный сеансовый ключ;

c) зашифрованное сообщение;

d) зашифрованный открытый ключ и зашифрованное сообщение.

Задачи

1. Найдите значение функции Эйлера – $\varphi(n)$ следующих чисел:

a) 15;

b) 72;

c) 311;

d) 128.

2. Зашифруйте методом RSA ($K_O = 91$, $n = 323$) следующие числа:

a) 35;

b) 94;

c) 248;

d) 236.

3. Расшифруйте методом RSA ($K_C = 3$, $n = 33$) следующие числа:

a) 294;

b) 531;

c) 7;

d) 111.

4. Зашифруйте методом Эль-Гамаля ($P = 43$, $G = 33$, $Y = 26$, $K = 11$) следующие числа:

a) 16;

b) 7;

c) 31;

d) 23.

5. Расшифруйте методом Эль-Гамаля ($P = 47$, $X = 21$) следующие варианты шифротекста (a,b):

a) $(45,20)$;

b) $(45,11)$;

c) $(45,14)$;

d) $(45,29)$.

ЛИТЕРАТУРА

1. Баричев С.Г., Гончаров В.В., Серов Р.Е. Основы современной криптографии. – М.: Горячая линия – Телеком, 2001. – 152 с.

2. Гоппа В.Д. Введение в алгебраическую теорию информации. – М.: Наука, Физматлит, 1995. – 112 с.

3. Колмогоров А.Н. Теория информации и теория алгоритмов. – М.: Наука, 1987. – 304 с.

4. Лидовский В.В. Теория информации: Учебное пособие. – М.: МАТИ, 2003. – 112 с.

5. Лукин Е.С. Прикладная теория информации: Учеб. пособие для студентов специальности «Информатика». – Мн.: БГУИР, 2002. – 42 с.

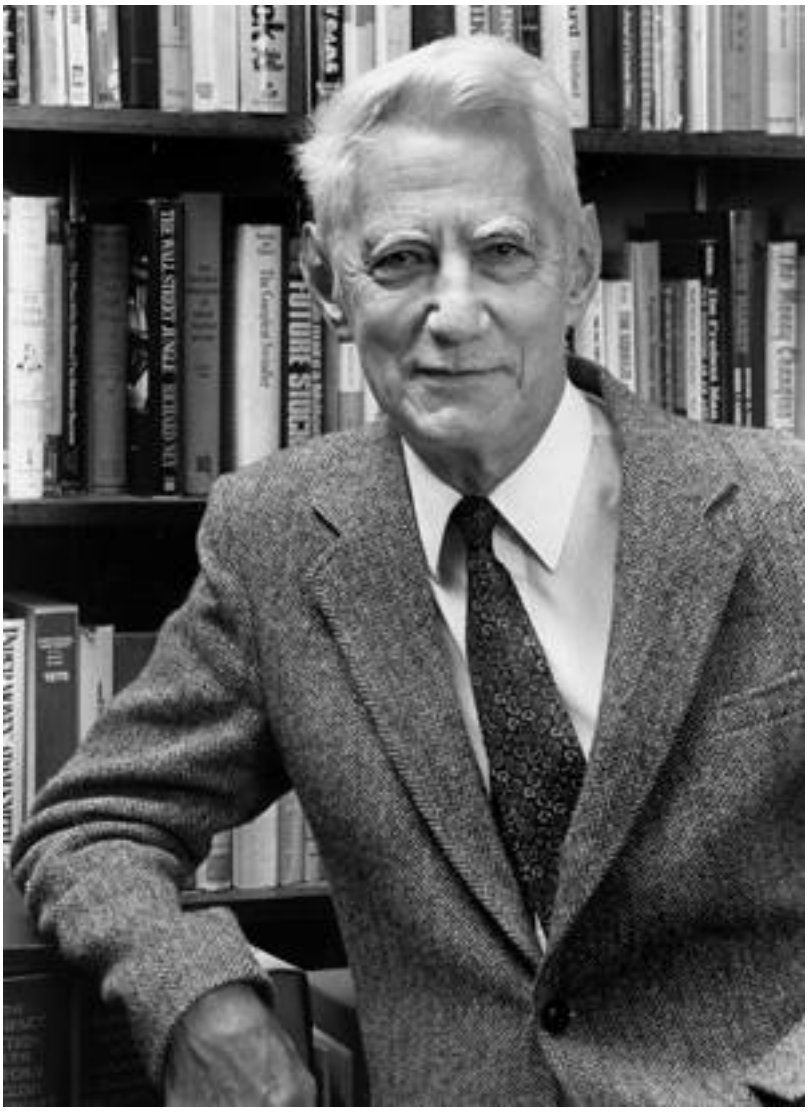
6. Молдовян А.А., Молдовян Н.А., Советов Б.А. Криптография. – СПб.: Лань, 2000.

7. Основы теории передачи информации: Учеб. пособие. Ч. I. Экономное кодирование / В.И. Шульгин. – Харьков: Нац. аэрокосм. ун-т «Харьк. авиац. ин-т», 2003.

8. Салома А. Криптография с открытым ключом: Пер. с англ. – М.: Мир, 1995. – 318 с.

9. Сорока Н.И., Кривинченко Г.А. Теория передачи информации: Конспект лекций для студентов специальности Т.11.01.00 «Автоматическое управление в технических системах». – Мн.: БГУИР, 1998. – 88 с.

10. MacKay D. Information Theory, Inference, and Learning Algorithms: Cambridge University Press, 2003. – 640 p.



Клод Шеннон (Claude Elmwood Shannon) родился в 1916 году и вырос в городе Гэйлорде штата Мичиган. Еще в детские годы Клод познакомился как с детальностью технических конструкций, так и с общностью математических принципов. Он постоянно возился с детекторными приемниками и радиоконструкторами и решал математические задачки и головоломки.

Будучи студентом Мичиганского университета, который он окончил в 1936 году, Клод специализировался одновременно и в математике, и в электротехнике. Летом 1938 года он занимался исследовательской работой в Массачусетсе и осенью был переведен с отделения электротехники на отделение математики, где начал работу над докторской диссертацией. Его начальник, Ванневэр Буш, стал в это время президентом Института Карнеги в Вашингтоне; одно из подразделений этого института, находящееся в

Колд Спринг Харбор (Cold Spring Harbor, N.Y.), занималось тогда генетикой, и он посоветовал Шеннону заняться с точки зрения алгебры проблемой хранения генетической информации. Шеннон провел там лето 1939 года, работая с генетиком Барбарой Баркс (Barbara Burks) над диссертацией, которую он назвал «Алгебра в теоретической генетике» (руководителем диссертации со стороны M.I.T был профессор Фрэнк Л. Хичкок (Frank L. Hitchcock), занимавшийся алгеброй).

Примерно в это же время Шеннон занимался разработкой идей в области вычислительных машин и систем связи. В письме от 16 февраля 1939 года он писал о зависимости между временем, пропускной способностью, шумом и искажениями в системах связи, а также о разработке вычислительных систем для выполнения символических математических операций.

Весной 1940 года он, наконец, защитил диссертации и получил звания магистра электротехники и доктора математики; летом он занимался дальнейшими исследованиями в области коммутирующих электрических цепей в Лабораториях Белла, разработав новый метод их проектирования, позволявший существенно сократить число контактов в них. Результаты этой работы были опубликованы в статье «Разработка двухконечных коммутирующих цепей» («The Synthesis of Two-Terminal Switching Circuits»).

Академический год 1940 – 1941 гг. он провел в Принстоне под руководством Германа Вейла (Hermann Weyl), начав серьезно работать над своими идеями относительно теории информации и эффективных систем связи.

Торнтон С. Фрай (Thornton C. Fry), глава отделения математики в Лабораториях Белла, был в это время членом комитета по разработке систем управления зенитным огнем – страна вооружалась в связи с европейской войной; он предложил Шеннону также поработать на оборону. Вернувшись в Лаборатории, Шеннон присоединился к группе, разрабатывающей устройства для обнаружения самолетов и ракет противника и наведения зенитных орудий; задача эта была актуальной в связи с созданием в Германии ракет Фау-1 и Фау-2. Без этих систем наведения потери Англии в войне были бы существенно большими.

Шеннон провел 15 лет в Лабораториях Белла в достаточно хорошем окружении – в это время там работали многие первоклассные математики, такие как Джон Пирс (John Pierce), известный своей работой в области спутниковой связи, Гарри Найквист (Harry Nyquist), много сделавший в теории обнаружения сигналов, Хендрик Бод (Hendrik Bode), занимавшийся обратной связью, создатели транзистора Браттин, Бардин и Шокли (Brattain, Bardeen и Shockley), Джордж Стибиц (George Stibitz), создавший первый (1938 год) релейный компьютер; Барни Оливер (Barney Oliver), выдающийся инженер, и другие.

Все эти годы Шеннон работал в различных областях, главным образом – в теории информации, началом которой послужила его статья «Математическая теория связи» («Mathematical Theory of Communication»). В этой статье было показано, что любой источник информации – телеграфный ключ, говорящий человек, телекамера и так далее – имеет «темп производства информации», который можно измерить в битах в секунду. Каналы связи имеют «пропускную способность», измеряемую в тех же единицах; информация может быть передана по каналу тогда и только тогда, когда пропускная способность не меньше темпа поступления информации. Эта статья по теории связи обычно считается наиболее весомым вкладом Шеннона в науку.

Занятия Шеннона проблемами информации и шума имели множество различных приложений. К примеру, в статье «Теория защищенной связи» («Communication Theory of Secrecy Systems») он связал криптографию с проблемой передачи информации по зашумленному каналу (роль шума в этом случае играет ключ криптосистемы). Эта работа привела в дальнейшем к тому, что Шеннон был назначен консультантом правительства США по вопросам криптографии.

Другой задачей, которой он занимался совместно с Е.Ф. Муром (E.F. Moore), было повышение надежности релейных цепей путем введения избыточного числа элементов (каждый из которых ненадежен). Эта задача, опять же, сводится к передаче информации по зашумленному каналу.

В более легком стиле выдержана его статья в области вычислительной техники «Программирование компьютера для игры в шахматы» («Programming a Computer for Playing Chess») 1950 года. В то время компьютеры были медленными, и программирование их было достаточно сложным; с тех пор создано множество шахматных программ, однако большинство из них и сейчас основаны на идеях этой работы.

В 1956 году Шеннон покинул Bell Labs и со следующего года стал профессором Массачусетского технологического института, откуда ушел на пенсию в 1978 году. В числе его студентов был, в частности, Марвин Мински и другие известные ученые, работавшие в области искусственного интеллекта.

Шеннон заложил основание и для современного кодирования с коррекцией ошибок, без которого не обходится сейчас ни один дисковод для жестких дисков или система потокового видео, и, возможно, многие продукты, которым еще только предстоит увидеть свет.

В 1965 году Шеннон по приглашению учавствовал в работе инженерной конференции в России.

Клод Шеннон скончался в 2001 году от болезни Альцгеймера на 84 году жизни.

Учебное издание

ЭЛЕМЕНТЫ ТЕОРИИ ИНФОРМАЦИИ

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС
для студентов специальности 1-40 01 01
«Программное обеспечение информационных технологий»

Составитель
БОГУШ Рихард Петрович

Редактор Ю.М. Казакевич

Дизайн обложки И.С. Васильевой

Подписано в печать 28.03.06. Формат 60×84 1/16. Гарнитура Таймс. Бумага офсетная.
Отпечатано на ризографе. Усл. печ. л. 9,28. Уч.-изд. л. 8,07. Тираж 60. Заказ 388

Издатель и полиграфическое исполнение:
Учреждение образования «Полоцкий государственный университет»

ЛИ № 02330/0133020 от 30.04.04 ЛП № 02330/0133128 от 27.05.04

211440, г. Новополоцк, ул. Блохина, 29