О.Г. Швечкова
А.Н. Пылькин
Д.В. Марчев

БАЗОВЫЕ
КРИПТОГРАФИЧЕСКИЕ
АЛГОРИТМЫ ЗАЩИТЫ
ИНФОРМАЦИИ

УЧЕБНОЕ ПОСОБИЕ

Рекомендовано Научно-методическим советом
Федерального государственного бюджетного образовательного
учреждения высшего образования «Рязанский государственный
радиотехнический университет» в качестве учебного пособия
для студентов высших учебных заведений, обучающихся по направлениям
подготовки 2.09.03.04 «Программная инженерия»
и 2.09.03.03. «Прикладная информатика»
Рецензенты:
Минаев В.А. — д-р техн. наук, профессор кафедры «Защита информации» МГТУ им. Н.Э. Баумана;
Кузнецов А.Е. — д-р техн. наук, зам. директора НИИ обработки аэрокосмических изображений «Фотон» (г. Рязань)

Швечкова О.Г.,


ISBN 978-5-906923-83-7

В учебном пособии рассмотрены вопросы решения проблем информационной безопасности методами криптографической защиты информации. Исследованы и проанализированы цели защиты информации в свете существующей Доктрины информационной безопасности Российской Федерации. Изложены теоретические основы базовой криптографии, формализации и программной реализации классических криптографических алгоритмов. Представлены механизмы контроля и тестирования знаний и приемов практической реализации рассмотренных методов (в форме оценочных материалов).

Учебное пособие предназначено для подготовки специалистов по образовательным программам направлений 2.09.03.04 «Программная инженерия» и 2.09.03.03 «Прикладная информатика».

© Швечкова О.Г., Пылькин А.Н., Марчев Д.В., 2018
© КУРС, 2018

Подписано в печать 10.01.2018.
Формат 60×90/16. Бумага офсетная. Гарнитура Newton.
Печать цифровая. Усл. печ. л. 10,5.
Тираж 500 экз. Заказ № 1242
TK 683039-961731-100118
ООО Издательство «КУРС»
127273, Москва, ул. Олонецкая, д. 17А, офис 104.
Тел.: (495) 203-57-83. E-mail: kursizdat@gmail.com http://kursizdat.ru
ВВЕДЕНИЕ

В настоящее время сфера информатизации на базе стремительно развивающихся средств вычислительной техники и открытых сетей передачи данных является одним из фундаментальных факторов развития всего общества, включая решение социальных задач, глобальных экономических и промышленных процессов. Мощные вычислительные возможности и оперативность передачи информации не только оказали большое влияние на принципы ведения бизнеса, сложившиеся в большинстве традиционных отраслей, но и открыли новые направления развития предпринимательской деятельности.

Процессы информатизации общества становятся не только отдельным видом деятельности, но и высокотехнологичной научной базой для формирования нового вида ценнego товара и стратегического ресурса — информации. Количество циркулирующей в мировом пространстве информации, появление новых видов информационных воздействий и пользователей с различными направлениями целеполагания приводят к тому, что неуклонно возрастает роль обеспечения безопасности жизнедеятельности, и особенно роль безопасного информационного обмена. В настоящее время вопросы информационной безопасности выходят на доминирующие позиции, так как во всех отраслях науки и техники все современные технологические и информационные системы стали оперировать информацией, представленной в одной из форм — электронной (или цифровой).

Последние достижения человеческой мысли в области компьютерных технологий связаны с появлением не только персональных компьютеров, сетей передачи данных и электронных денег, но и таких понятий, как хакер, информационное оружие, компьютерные вирусы и т.п.

В данном контексте под информационной безопасностью подразумевается одно из ведущих направлений развития информационных технологий по обеспечению защищенности обрабатываемых, хранимых и передаваемых данных от незаконного ознакомления, преобразования и уничтожения (как крайний случай модификации), а также состояние защищенности информационных ресурсов от воздействий, направленных на нарушение их работоспособности.

Современные задачи обучения и формирования жизненной позиции молодых специалистов с учетом требований времени, информационно-технологического развития нашей страны и вызовов окру-
жающего мира требуют инновационного подхода к базовому образованию и изложению традиционных фундаментальных дисциплин. В данном контексте обучение основам информационной безопасности является непреложным требованием времени и значимым маркером зрелости базовой подготовки современного специалиста в области информатизации, гарантией качества и актуальности подготовки специалистов инженерных направлений.

Актуальные задачи инновационной стратегии образования четко обозначены в государственной политике формирования безопасной среды жизнеобеспечения во всех сферах существования нашего народа и общества.

Конкретные задачи подготовки современного специалиста являются отражением фундаментальных тезисов по обеспечению национальной безопасности нашего государства, изложенных в Доктрине информационной безопасности Российской Федерации.

Основные положения

Доктрины информационной безопасности РФ

6 декабря 2016 г. опубликован Указ президента РФ Владимира Путина об утверждении новой Доктрины информационной безопасности РФ.

Доктрина является документом стратегического планирования в сфере национальной безопасности. Действовавший ранее аналогичный документ от 9 сентября 2000 г. утратил свою силу.

Доктрина информационной безопасности РФ представляет собой систему официальных взглядов на обеспечение национальной безопасности государства в информационной сфере, под которой понимают совокупность информации, сайтов, сетей связи, а также государственных и частных компаний, обеспечивающих их работу. Главная стратегическая цель документа — защита жизненно важных интересов личности, общества и государства от внутренних и внешних угроз, связанных с применением информационных технологий в военно-политических целях.

В Доктрине перечислены основные информационные угрозы национальной безопасности России. Среди них:

- стремление «отдельных государств» использовать технологическое превосходство для доминирования в информационном пространстве;
- наращивание зарубежными странами возможностей по оказанию «информационно-психологического воздействия» на российское
население с целью внутриполитической дестабилизации и подрыва суверенитета РФ;
• увеличение в зарубежных СМИ числа материалов, содержащих «предвзятую оценку государственной политики РФ», дискриминация российских средств массовой информации за рубежом.

В качестве угрозы в документе также определено технологическое отставание РФ в сфере информационных технологий, высокий уровень зависимости от зарубежной компонентной базы и программного обеспечения, недостаточная эффективность отечественных научных исследований.

Отдельно в списке информационных угроз указывается на рост киберпреступности, в первую очередь в кредитно-финансовой сфере. Документ предусматривает пять основных сфер, в которых необходимо обеспечение информационной безопасности РФ.

Оборона
Состояние информационной безопасности в области обороны страны характеризуется увеличением масштабов применения отдельными государствами и организациями информационных технологий в военно-политических целях, в том числе для осуществления действий, противоречащих международному праву, направленных на подрыв суверенитета, политической и социальной стабильности, территориальной целостности Российской Федерации и ее союзников и представляющих угрозу международному миру, глобальной и региональной безопасности.

Госбезопасность
Состояние информационной безопасности в области государственной и общественной безопасности характеризуется постоянным повышением сложности, увеличением масштабов и ростом скоординированных компьютерных атак на объекты критической информационной инфраструктуры, усилиением разведывательной деятельности иностранных государств в отношении Российской Федерации, а также нарастанием угроз применения информационных технологий в целях нанесения ущерба суверенитету, территориальной целостности, политической и социальной стабильности Российской Федерации.

Экономика
Состояние информационной безопасности в экономической сфере характеризуется недостаточным уровнем развития конкурентоспособных информационных технологий и их использования для про-
изводства продукции и оказания услуг. Остается высоким уровень зависимости отечественной промышленности от зарубежных информационных технологий в части, касающейся электронной компонентной базы, программного обеспечения, вычислительной техники и средств связи, что обусловливает зависимость социально-экономического развития Российской Федерации от геополитических интересов зарубежных стран.

Наука, технологии и образование
Состояние информационной безопасности в области науки, технологий и образования характеризуется недостаточной эффективностью научных исследований, направленных на создание перспективных информационных технологий, низким уровнем внедрения отечественных разработок и недостаточным кадровым обеспечением в области информационной безопасности, а также низкой осведомленностью граждан в вопросах обеспечения личной информационной безопасности. При этом мероприятия по обеспечению безопасности информационной инфраструктуры, включая ее целостность, доступность и устойчивое функционирование, с использованием отечественных информационных технологий и отечественной продукции зачастую не имеют комплексной основы.

Стратегическая стабильность
Состояние информационной безопасности в области стратегической стабильности и равноправного стратегического партнерства характеризуется стремлением отдельных государств использовать технологическое превосходство для доминирования в информационном пространстве.

В каждой из них доктрина предусматривает несколько основных направлений противостояния информационным угрозам. В их числе:
• нейтрализация информационно-психологического воздействия, направленного на подрыв исторических основ и патриотических традиций, связанных с защитой Отечества» (оборона);
• повышение защищенности критически важной информационной инфраструктуры, противодействие экстремизму и размыванию «традиционных российских духовно-нравственных ценностей» (госбезопасность);
• инновационное развитие электронной промышленности, импортозамещение (экономика);
• разработка перспективных технологий (наука); развитие национальной системы управления российским сегментом Интернета (стратегическая стабильность).
В новом документе больший акцент сделан на опасности «информационно-психологического воздействия» на индивидуальное и общественное сознание граждан РФ со стороны иностранных спецслужб, а также террористических и экстремистских организаций.

В версии 2000 г. понятие «экстремистские организации» отсутствовало, а в качестве источников угроз были названы «диверсионно-подрывная деятельность иностранных специальных служб» и «деятельность международных террористических организаций».

Также в доктрине 2016 г. впервые отмечается тенденция к увеличению негативных оценок России в зарубежных СМИ. В предыдущей редакции документа говорилось лишь об «опасности зависимости духовной, экономической и политической сфер общественной жизни России от зарубежных информационных структур».

В соответствии с задачами современной образовательной системы авторами предлагается научно-практический подход с элементами промежуточного и обобщенного тестирования основных разделов изучаемого материала.

Анализ угроз безопасности информации и информационных систем, тенденции развития информационных технологий дают все основания сделать вывод о постоянном возрастании роли криптографических методов при решении общих вопросов защиты информации в целом, задач аутентификации в распределенных системах, обеспечения секретности данных при их передаче по открытым каналам связи, обеспечения юридической значимости результатов информационного обмена. Специалистами в области информационной безопасности криптографической защите уделяется особое внимание, считая ее наиболее надежным, а в некоторых ситуациях — единственно возможным средством защиты.

В рамках изучаемой дисциплины «Информационная безопасность» наряду с освоением фундаментальных и математических основ рассматриваемого предмета значительное внимание уделяется разделу криптографии, где в соответствии с известной общепринятой классификацией алгоритмов шифрования рассматриваются как классические алгоритмы, так и самые современные, применимые как за рубежом, так и в нашей стране.

При изложении дисциплины «Информационная безопасность» наряду с освоением фундаментальных и математических основ рассматриваемого предмета значительное внимание уделяется разделу криптографии, при изучении которого предлагается следующий подход, положенный в основу представленной методики.
Данный подход может рассматриваться только в контексте решения большой общей проблемы защиты информации и информационных систем и реализуется в виде учебно-практического пособия, разделенного концептуально на два уровня.

Данное учебно-практическое пособие является первой частью издания и посвящено изучению базовых понятий и общих положений криптографии и классических алгоритмов, основным угрозам современного информационного обмена, роли современной криптографии при решении общей задачи защиты информации. Подробно представлена классификация криптографических алгоритмов.

Изучение каждого из алгоритмов представлено в виде отдельной темы, содержащей соответствующий теоретический материал, нормализацию и программную реализацию задачи. Программная реализация адаптирована для учебных целей. С целью проведения промежуточного контроля разработана система контрольных вопросов и тестовых задач.

В завершение курса по изучению классических криптографических алгоритмов проводится результирующее тестирование по всем разделам представленного учебного пособия.
Тема 1

ОСНОВНЫЕ ПОЛОЖЕНИЯ ТЕОРИИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

1.1. Понятие информационной безопасности. Задачи обеспечения информационной безопасности

В зависимости от контекста и области применения существует несколько определений понятия информационной безопасности.

В Законе РФ «Об участии в международном информационном обмене» информационная безопасность определяется как состояние защищенности информационной среды общества, обеспечивающее ее формирование, использование и развитие в интересах граждан, организаций, государства.

В Доктрине информационной безопасности РФ указывается, что информационная безопасность характеризует состояние защищенности национальных интересов в информационной сфере, определяемых совокупностью сбалансированных интересов личности, общества и государства.

В «Концепции информационной безопасности сетей связи и общего пользования РФ» даны два определения этого понятия.

Информационная безопасность — это свойство сетей связи общего пользования противостоять возможности реализации нарушителем угрозы информационной безопасности.

Информационная безопасность — свойство сетей связи общего пользования сохранять неизменными характеристики информационной безопасности в условиях возможных воздействий нарушителя.

В зависимости от конкретных угроз безопасности и специальных задач по защите информационных систем и пользовательской информации понятие информационной безопасности трактуется с акцентом на специальных решениях или общем подходе к проблеме.

Информационная безопасность — это защищенность информации и поддерживающей ее инфраструктуры от случайного или преднамеренного воздействия естественного или искусственного характера, которое может нанести ущерб владельцам или пользователям информации.
Под защитой информации понимается комплекс мероприятий, направленных на обеспечение информационной безопасности.

Согласно ГОСТ 50922-2006 защита информации — это деятельность, направленная на предотвращение утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию.

Основными задачами защиты пользовательской информации являются:

• обеспечение конфиденциальности информации;
• обеспечение целостности информации;
• обеспечение достоверности информации;
• обеспечение оперативности доступа к информации;
• обеспечение юридической значимости информации, представленной в виде электронного документа;
• обеспечение неотслеживаемости действий клиента.

Под конфиденциальностью следует понимать свойство информации быть доступной только ограниченному кругу пользователей информационной системы, в которой циркулирует данная информация.

К сожалению, практическая реализация мер по обеспечению конфиденциальности современных информационных систем имеет определенные проблемы. Во-первых, сведения о технических каналах утечки информации по совершенно понятным причинам являются закрытыми, так что большинство пользователей лишено возможности составить представление о потенциальных рисках. Во-вторых, на пути пользовательской криптографии как основного средства обеспечения конфиденциальности стоят многочисленные законодательные препоны и технические проблемы.

Целостность информации условно подразделяется на статическую и динамическую.

Статическая целостность информации предполагает неизменность информационных объектов от их исходного состояния, определяемого автором или источником информации.

Динамическая целостность информации включает вопросы корректного выполнения действий с информационными потоками, например анализ потока сообщений для выявления некорректных потоков, контроль правильности передачи сообщений, подтверждение отдельных сообщений и др.

Целостность — гарантия того, что информация сейчас существует в ее исходном виде, т.е. при ее хранении или передаче не было произведено несанкционированных изменений.
Под целостностью понимается свойство информации или программного обеспечения сохранять свою структуру и (или) содержание в процессе передачи и (или) хранения.

Рассматривая вопрос передачи информации в виде сообщений через сеть, можно прийти к заключению, что каждое сообщение по своему смысловому содержанию образует некоторый класс. Другими словами, смысл конечного сообщения останется таким же, как и начального, даже если форма представления информации в электронном виде изменится. Таким образом, каждое сообщение на русском языке будет иметь свой класс эквивалентности, и для данного случая свойство сохранения целостности информации можно сформулировать следующим образом: переданное сообщение $X$ считается сохранившим целостность, если полученное в результате передачи сообщение $X'$ принадлежит классу эквивалентности сообщения $X$.

Достоверность — свойство информации, выражающееся в строгой принадлежности объекту, который является ее источником, либо тому объекту, от которого эта информация принята.

Оперативность — способность информации или некоторого информационного ресурса быть доступным для конечного пользователя в соответствии с его временными потребностями.

Юридическая значимость означает, что документ обладает юридической силой. С этой целью субъекты, которые нуждаются в подтверждении юридической значимости передаваемого сообщения, договариваются о повсеместном принятии некоторых атрибутов информации, выражающих ее способность быть юридически значимой. Данное свойство информации особенно актуально в системах электронных платежей, где осуществляется операция по переводу денежных средств.

Неотслеживаемость — способность совершать некоторые действия в информационной системе незаметно для других объектов. Актуальность данного требования стала очевидной благодаря появлению таких понятий, как электронные деньги и Internet-banking. Так, для авторизации доступа к электронной платежной системе пользователь должен предоставить сведения, однозначно его идентифицирующие.

При анализе проблематики, связанной с информационной безопасностью, необходимо учитывать специфику данного аспекта деятельности, состоящую в том, что информационная безопасность есть составная часть информационных технологий — области, развивающейся беспрецедентно высокими темпами. Здесь важны не столько отдельные решения (законы, учебные курсы, программно-технические изделия), находящиеся на современном уровне, сколько
механизмы генерации новых решений, позволяющие жить в темпе технического прогресса, постоянно изменяющегося круга задач и достаточно агрессивного внешнего окружения.

К сожалению, современная технология программирования не позволяет создавать безошибочные программы, что не способствует быстрому развитию средств обеспечения информационной безопасности. Следует исходить из того, что необходимо конструировать надежные системы (информационной безопасности) с привлечением ненадежных компонентов (программ). В принципе это возможно, но требует соблюдения определенных архитектурных принципов и контроля состояния защищенности на всем протяжении жизненного цикла информационных систем (ИС).

В таких условиях системы информационной безопасности должны уметь противостоять разнообразным атакам, как внешним, так и внутренним, атакам автоматизированным и скоординированным. Иногда нападение длится доли секунды; порой прощупывание уязвимых мест ведется медленно и растягивается на часы, так что подозрительная активность практически незаметна. Целью злоумышленников может быть нарушение всех составляющих информационной безопасности — доступности, целостности или конфиденциальности.

1.2. Угрозы информационной безопасности

В общем случае под угрозой принято понимать потенциально возможное событие, действие, процесс или явление, которые могут привести к нанесению ущерба чьим-либо интересам.

Попытка реализации угрозы называется атакой, субъект, предпринимающий такую попытку, — злоумышленником. Потенциальные злоумышленники называются источниками угрозы.

Угроза информационной безопасности — это возможность реализации воздействия на информацию, приводящего к нарушению конфиденциальности, целостности или доступности этой информации, а также возможность воздействия на компоненты системы, приводящего к их утрате, уничтожению или сбою функционирования.

Чаще всего угроза является следствием наличия уязвимых мест в защите информационных систем (таких, например, как возможность доступа посторонних лиц к критически важному оборудованию или ошибки в программном обеспечении).

Промежуток времени от момента, когда появляется возможность использовать слабое место, и до момента, когда пробел ликвидируется, называется окном опасности, ассоциированным с данным уяз-
вым местом. Пока существует окно опасности, возможны успешные атаки на ИС.

При эксплуатации программного обеспечения (ПО), в случае обнаружения неисправностей и появления средств использования ошибки открывается окно опасности, которое ликвидируется при наложении «заплат», исправляющих соответствующие погрешности.

Для большинства уязвимых мест окно опасности существует сравнительно долго (несколько дней, иногда — недель), поскольку за это время должны произойти следующие события:

- должно стать известно о средствах использования пробела в защите;
- должны быть выпущены соответствующие заплаты;
- заплаты должны быть установлены в защищаемой ИС.

Подчеркнем, что само понятие «угроза» в разных ситуациях частую трактуется по-разному. Например, для подчеркнутой открытой организации угроз конфиденциальности может просто не существовать — вся информация считается общедоступной; однако в большинстве случаев нелегальный доступ представляется серьезной опасностью. Иными словами, угрозы, как и все в ИБ, зависят от интересов субъектов информационных отношений (и от того, какой ущерб является для них неприемлемым).

На практике чаще всего используется следующая основная классификация угроз, основывающаяся на трех введенных ранее базовых свойствах защищаемой информации.

1. Угрозы нарушения конфиденциальности информации, в результате реализации которых информация становится доступной субъекту, не располагающему полномочиями для ознакомления с ней.

2. Угрозы нарушения целостности информации, к которым относится любое злонамеренное искажение информации, обрабатываемой с использованием ИС.

3. Угрозы нарушения доступности информации, возникающие в тех случаях, когда доступ к некоторому ресурсу ИС для легальных пользователей блокируется.

1.2.1. Основные угрозы конфиденциальности

Конфиденциальную информацию можно разделить на предметную и служебную. Служебная информация (например, пароли пользователей) не относится к определенной предметной области, в информационной системе она играет техническую роль, но ее раскрытие особенно опасно, поскольку оно чревато получением несанкционированного доступа ко всей информации, в том числе предметной.
Перехват данных — очень серьезная угроза, и если конфиденциальность действительно является критичной, а данные передаются по многим каналам, их защита может оказаться весьма сложной и дорогостоящей. Технические средства перехвата хорошо проработаны, доступны, прости в эксплуатации, а установить их, например, на кабельную сеть может кто угодно, так что эту угрозу нужно принимать во внимание по отношению не только к внешним, но и к внутренним коммуникациям.

Кражи оборудования являются угрозой не только для резервных носителей, но и для компьютеров, особенно портативных.

К неприятным угрозам, от которых трудно защититься, можно отнести и злоупотребление полномочиями. На многих типах систем привилегированный пользователь (например, системный администратор) способен прочитать любой (незашифрованный) файл, получить доступ к почте любого пользователя и т.п.

1.2.2. Основные угрозы целостности

На втором месте по размерам ущерба (после непреднамеренных ошибок и упущений) стоят кражи и подлоги. Можно предположить, что реальный ущерб от подобных противоправных действий с использованием персональных компьютеров был намного больше, чем известно в реальности, поскольку многие организации по понятным причинам скрывают такие инциденты; не вызывает со мнений, что в наши дни ущерб от такого рода действий возрастает многократно.

В большинстве случаев виновниками оказываются штатные сотрудники организаций, отлично знакомые с режимом работы и мерами защиты. Это еще раз подтверждает опасность внутренних угроз, хотя упоминается о них значительно реже, чем о внешних.

Ранее указывалось различие между статической и динамической целостностью. С целью нарушения статической целостности злоумышленник (как правило, штатный сотрудник) может:

• ввести неверные данные;
• изменить данные.

Иногда изменяются содержательные данные, иногда — служебная информация.

Угрозой целостности является не только фальсификация или изменение данных, но и отказ от совершенных действий. Если нет средств обеспечить неотказуемость, компьютерные данные не могут рассматриваться в качестве доказательства.
Потенциально уязвимы с точки зрения нарушения целостности не только данные, но и программы. Внедрение рассмотренного выше вредоносного ПО — пример подобного нарушения.

Угрозами динамической целостности являются нарушение атомарности транзакций, переупорядочение, кража, дублирование данных или внесение дополнительных сообщений (сетевых пакетов и т.п.). Соответствующие действия в сетевой среде называются активным прослушиванием.

1.2.3. Основные угрозы доступности

Угрозы доступности могут выглядеть достаточно примитивно — как повреждение или даже разрушение оборудования (в том числе носителей данных). Такое повреждение может вызываться естественными причинами. К сожалению, находящиеся в массовом использовании источники бесперебойного питания не защищают от мощных кратковременных импульсов и случаи выгорания оборудования — не редкость.

Общеизвестно, что периодически необходимо производить резервное копирование данных. Однако, даже если это предложение выполняется, резервные носители зачастую хранят небрежно, не обеспечивая их защиту от вредного воздействия окружающей среды. И когда требуется восстановить данные, оказывается, что эти самые носители никак не желают читаться.

1.3. Роль криптографических протоколов в задаче обеспечения информационной безопасности

Основу обеспечения информационной безопасности в информационно-телекоммуникационных системах составляют криптографические методы и средства защиты информации. Следует учесть, что наиболее надежную защиту можно обеспечить только с помощью комплексного подхода, т.е. решение задачи должно представлять собой совокупность организационно-технических и криптографических мероприятий.

В основе криптографических методов лежит понятие криптографического преобразования информации, производимого по определенным математическим законам с целью исключить доступ к данной информации посторонних пользователей, а также с целью обеспечения невозможности бесконтрольного изменения информации со стороны тех же самых лиц.
Применение криптографических методов защиты обеспечивает решение основных задач информационной безопасности. Этого можно добиться путем реализации следующих криптографических методов защиты как пользовательской и служебной информации, так и информационных ресурсов в целом:

- **шифрование** всего информационного трафика, передающегося через открытые сети передачи данных, и отдельных сообщений;
- **криптографическая аутентификация** устанавливающих связь разноуровневых объектов (имеются в виду уровни модели взаимодействия открытых систем);
- **защита несущего данные трафика** средствами имитозащиты (защиты от навязывания ложных сообщений) и электронно-цифровой подписи с целью обеспечения целостности и достоверности передаваемой информации;
- **шифрование данных**, представленных в виде файлов либо хранящихся в базе данных;
- **контроль целостности программного обеспечения** путем применения криптографически стойких контрольных сумм;
- **применение электронно-цифровой подписи** для обеспечения юридической значимости платежных документов; применение заменяющей цифровой подписи для обеспечения неотслеживаемости действий клиента в платежных системах, основанных на понятии электронных денег.

В общем случае взаимодействие объектов (субъектов) подобных систем всегда сопровождается соблюдением некоторых договоренностей, называемых протоколом.

Формально протоколом следует считать последовательность действий объектов (субъектов) для достижения определенной цели. В свою очередь, криптографическими протоколами будем называть те, в которых участники для достижения определенной цели используют криптографические преобразования информации.

Перечислим основные задачи обеспечения информационной безопасности, которые решаются с помощью криптографических протоколов:

- **обмен ключевой информацией** с последующей установкой защищенного обмена данными. При этом не существует никаких предположений, общались ли предварительно между собой стороны, обменивающиеся ключами (например, без использования криптографических протоколов невозможно было создать системы распределения ключевой информации в распределенных сетях передачи данных);
- **аутентификация сторон**, устанавливающих связь;
авторизация пользователей при доступе к телекоммуникационным и информационным службам.

Стремительное развитие криптографических протоколов в большей степени стимулируется развитием систем электронных платежей, интеллектуальных карт, появлением электронных денег и т.д. Таким образом, прогнозируется не только увеличение числа компаний, занимающихся бизнесом в сети, и их общего оборота, но и реальное возрастание среднего дохода, приходящегося на одну такую компанию. Поскольку на сегодняшний день основным криптографическим средством защиты информации в Интернете являются протоколы, можно констатировать, что развитие подобных средств защиты коммерческих тайн будет продолжаться и в количественном, и в качественном отношении.

Многогранность применения криптографических протоколов в решении задач обеспечения информационной безопасности как в локальных информационных системах, так и в распределенных информационных системах приводит к необходимости детального рассмотрения их основных типов, вопросов практического применения таких протоколов и построения на их основе специальных информационных систем.

Учитывая, что основой любого криптопротокола являются так называемые криптоалгоритмы, рассмотрим вопросы построения и практического применения основных типов подобных механизмов. Кроме хорошо известных и повсеместно используемых зарубежных криптоалгоритмов, исследуются отечественные разработки в области информационной безопасности и стандарты на криптоалгоритмы.

1.4. Общие сведения классической криптографии

1.4.1. Основные понятия и определения классической криптографии

Прежде чем перейти к рассмотрению криптографических протоколов, а также к их практическому применению, необходимо уделить внимание вопросам, которые в рамках криптографии давно признаются классическими, а именно — основам построения систем засекреченной связи.

Под системой засекреченной связи будем понимать систему передачи информации, в которой смысл передаваемой информации скрывается с помощью криптографической обработки, т.е. фактом факта передачи информации не учитывается наличие какой-либо системы.
засекреченной связи — использование алгоритмов шифрования как основного средства сохранения конфиденциальности.

Шифрование — процесс криптографического преобразования множества открытых сообщений во множество закрытых сообщений.

Расшифрование — процесс криптографического преобразования закрытых сообщений в открытые.

Дешифрование — процесс нахождения открытого сообщения, соответствующего заданному закрытому, при неизвестном криптографическом преобразовании.

Абстрактно систему засекреченной связи можно описать как множество отображений множества открытых сообщений в множество закрытых. Выбор конкретного типа преобразования определяется ключом расшифрования (или зашифрования). Отображения должны обладать свойством взаимооднозначности, т.е. при расшифровании должен получаться единственный результат, совпадающий с первоначальным открытым сообщением (рис. 1.1).

![Diagram of encryption and decryption](image)

Рис. 1.1. Общая структура системы засекреченной связи

Ключи зашифрования и расшифрования могут в общем случае быть различными, хотя для простоты рассуждений (применительно к этой главе) предположим, что они идентичны. Множество, из которого выбираются ключи, называется ключевым пространством.

Совокупность процессов зашифрования, множества открытых сообщений, множества возможных закрытых сообщений и ключевого пространства называется алгоритмом зашифрования.

Совокупность процессов расшифрования, множества возможных закрытых сообщений, множества открытых сообщений и ключевого пространства называется алгоритмом расшифрования.

Работу системы засекреченной связи можно описать следующим образом.

1. Из ключевого пространства выбирается ключ зашифрования $K$ и отправляется по надежному каналу передачи.
2. К открытому сообщению $C$, предназначенному для передачи, применяют конкретное преобразование $F_k$, определяемое ключом $K$, для получения зашифрованного сообщения $M: M = F_k(C)$.

3. Полученное зашифрованное сообщение $M$ пересылают по каналу передачи данных.

4. На принимающей стороне к полученному сообщению $M$ применяют конкретное преобразование $D_k$, определяемое из всех возможных преобразований ключом $K$, для получения открытого сообщения $C: C = D_k(M)$.

Наличие потенциального противника приводит к тому, что система засекреченной связи может быть скомпрометирована. На практике обеспечение надежности функционирования подобной системы сводится к стойкости используемых алгоритмов шифрования, лежащих в основе всех операций. Это обусловлено тем, что стойкость всей системы не может быть выше стойкости алгоритмов шифрования, однако может быть и гораздо ниже, хотя бы в силу технической реализации самой системы.

В данном теоретическом разделе рассматриваются такие базовые понятия, как требования к алгоритмам шифрования, вопросы компрометации алгоритмов в случае неограниченных возможностей противника (времени, вычислительных средств, количества перехваченных зашифрованных сообщений), стойкость алгоритмов шифрования.

1.4.2. Стойкость алгоритмов шифрования

Для каждого открытого сообщения существует априорная вероятность выбора, поскольку механизм выбора открытых сообщений можно представить как некоторый вероятностный процесс. Аналогично выбор каждого ключа тоже имеет априорную вероятность. Противник, перехватывающий зашифрованные сообщения, может вычислить апостериорные вероятности как появления открытого сообщения, так и вероятность появления ключа.

Предположим, что противнику известны все криптографические преобразования, используемые в системе засекреченной связи, а также ключевое пространство. В результате перехвата некоторого объема зашифрованных сообщений и вычисления апостериорных вероятностей противник поймет, что им будет соответствовать единственное решение об использовании ключа или передаче открытого сообщения (точка единственности принятия решения), удовлетворяющего данным вероятностям. Понятно, что подобный вывод вполне может привести к раскрытию системы противником.
Под раскрытием системы засекреченной связи или алгоритма шифрования следует понимать одну из следующих планируемых противником операций, направленных на достижение этой цели:

- полное раскрытие. Противник находит путем вычислений секретный ключ системы;
- нахождение эквивалентного алгоритма. Противник находит алгоритм, функционально эквивалентный алгоритму шифрования, не имея при этом представления об используемом секретном ключе;
- нахождение открытого сообщения. Противник находит открытое сообщение, соответствующее одному из перехваченных зашифрованных;
- частичное раскрытие. Противник получает частичную информацию об используемом ключе или об открытом сообщении.

Криптостойкостью называется характеристика шифра, определяющая его стойкость к расшифрованию без знания ключа (т.е. криптоатаке). Показатель криптостойкости — главный параметр любой криптосистемы.

В качестве показателя криптостойкости можно выбрать:

- количество всех возможных ключей или вероятность подбора ключа за заданное время с заданными ресурсами;
- количество операций или время (с заданными ресурсами), необходимое для взлома шифра с заданной вероятностью;
- стоимость вычисления ключевой информации или исходного текста.

Однако следует понимать, что эффективность защиты информации криптографическими методами зависит не только от криптостойкости шифра, но и от множества других факторов, включая вопросы реализации криптосистем в виде устройств или программ. При анализе криптостойкости шифра необходимо учитывать и человеческий фактор. Например, подкуп конкретного человека, в руках которого сосредоточена необходимая информация, может стоить на несколько порядков дешевле, чем создание суперкомпьютера для взлома шифра.

Стойкость криптографического алгоритма необходимо рассматривать относительно пары атака—цель, где под целью противника понимается планируемая угроза. В мировой литературе проработана классификация различных типов атак на криптографические алгоритмы:

- атака с известным шифротекстом (ciphertext-only attack). Предполагается, что противник знает криптосистему, т.е. алгоритмы
шифрования, но не знает секретный ключ. Кроме того, ему известен лишь набор перехваченных криптограмм;

• **атака с известным открытым текстом** (known plaintext attack). То же, что предыдущая, но противник получает в свое распоряжение еще некоторый набор криптограмм и соответствующих им открытых текстов;

• **простая атака с выбором открытого текста** (chosen-plaintext attack). Противник имеет возможность выбрать необходимое количество открытых текстов и получить соответствующие им криптограммы;

• **адаптивная атака с выбором открытого текста** (adaptive-chosen-plaintext attack). В этом случае противник имеет возможность выбирать открытые тексты с учетом того, что криптограммы всех предыдущих открытых текстов ему известны;

• **атака с выбором шифротекста** (chosen-ciphertext attack). Противник имеет возможность выбрать необходимое количество криптограмм и получить соответствующие им открытые тексты;

• **адаптивная атака с выбором шифротекста** (adaptive-chosen-ciphertext attack). Противник, выбирая очередную криптограмму, знает все открытые тексты, соответствующие предыдущим криптограммам;

• **атака с выбором текста** (chosen-text attack). Противник имеет возможность выбирать как криптограммы (и дешифровывать их), так и открытые тексты (и зашифровывать их);

• **атака с выбором ключа** (chosen-key attack). Противник знает не сами ключи, а некоторые различия между ними.

В этом перечне атаки представлены по мере возрастания их силы. Анализируя атаки и их цели, можно прийти к выводу, что наибольшей стойкостью алгоритм обладает в том случае, когда он способен противостоять самой сильной атаке, проводимой противником, при условии, что он преследует самую слабую из возможных целей атаки.

Теоретически существуют абсолютно стойкие алгоритмы шифрования. Математическое доказательство данного факта было предложено в работах К. Шеннона. Для того чтобы алгоритм считался абсолютно стойким, он должен удовлетворять следующим условиям:

• длина ключа и длина открытого сообщения должны быть одинаковы;

• ключ должен использоваться только один раз;

• выбор ключа из ключевого пространства должен осуществляться равномерно.
Данные требования приводят к тому, что абсолютно стойкие алгоритмы с практической точки зрения являются труднореализуемыми. Например, осуществление первого и второго условий приводит к тому, что необходимо иметь запас ключей большой длины, что практически невыполнимо.

Рассмотрим самые распространенные на сегодняшний день причины осуществления успешных атак на алгоритмы шифрования:
• наличие статистической структуры исторически сложившихся языков. То есть существуют определенные символы или комбинации символов, наиболее часто встречающиеся в естественной речи;
• наличие вероятных слов. Речь идет о словах или выражениях, появление которых можно ожидать в перехваченном сообщении. Так, в деловой переписке присутствуют шаблонные слова; в английском языке, например, наиболее часто встречаются «and», «the», «are» и т.д.

Следует учесть, что существует ряд методов, позволяющих сделать зашифрованные сообщения практически непригодными для статистического анализа и анализа посредством вероятных слов. К ним относятся:
• рассеивание. Влияние одного символа открытого сообщения распространяется на множество символов зашифрованного сообщения. Этот метод хотя и приводит к увеличению количества ошибок, однако с его помощью удается скрыть статистическую структуру открытого сообщения;
• запутывание. Развитием принципа рассеивания стал принцип запутывания, в котором влияние одного символа ключа распространяется на множество символов зашифрованного сообщения;
• перемешивание. Принцип перемешивания основывается на использовании особых преобразований исходного сообщения, в результате чего вероятные последовательности как бы рассеиваются по всему пространству возможных открытых сообщений.

Примерами изложенных выше методов могут служить стандарты шифрования, такие как DES (Data Encryption Standard) и ГОСТ 28147-89.

1.4.3. Основные методы криптографической защиты информации

Криптографические методы предотвращения угроз (криптографические алгоритмы) в компьютерной системе являются наиболее эффективными способами защиты информационных систем. При этом под криптографическим преобразованием информации понимается
такое преобразование исходной информации, в результате которого она становится недоступной для ознакомления и использования лицами, не имеющими на это полномочий.

По виду воздействия на исходную информацию эти методы подразделяют на четыре группы: кодирование, сжатие-расширение, стеганография, шифрование-дешифрование.

Содержанием процесса кодирования информации является замена смысловых конструкций исходной информации (слов, предложений) кодами. В качестве кодов могут использоваться сочетания букв, цифр, букв и цифр. При кодировании и обратном преобразовании используют специальные таблицы или словари. Кодирование информации целесообразно применять в системах с ограниченным набором смысловых конструкций. Такой вид криптографического преобразования применим, например, в командных линиях автоматизированных систем управления. Недостатком кодирования конфиденциальной информации является необходимость хранения и распространения кодировочных таблиц, которые требуется часто менять, чтобы избежать раскрытия кодов статистическими методами обработки перехваченных сообщений.

Основным видом криптографического преобразования информации в компьютерной системе является шифрование или дешифрование. Под шифрованием понимается преобразование открытой информации в зашифрованную информацию (шифратекс) или обратное преобразование зашифрованной информации в открытую. Процесс преобразования открытой информации в закрытую получил название зашифрование, а обратный процесс — расшифрование.

Методом шифрования (шифром) называется совокупность обратимых преобразований открытой информации в закрытую в соответствии с алгоритмом шифрования.

Атака на шифр — это процесс расшифрования закрытой информации без знания ключа и, возможно, при отсутствии сведений об алгоритме шифрования.

Современные методы шифрования должны отвечать следующим требованиям.

1. Стойкость шифра к противостоянию криптоанализу (криптостойкость) должна быть такой, чтобы вскрытие его могло быть осуществлено только путем решения задачи полного перебора ключей.

2. Криптостойкость обеспечивается не секретностью алгоритма шифрования, а секретностью ключа.

3. Шифратекс не должен существенно превосходить по объему исходную информацию.
4. Ошибки, возникающие при шифровании, не должны приводить к искажениям и потерям информации.
5. Время шифрования не должно быть большим.
6. Стоимость шифрования должна быть согласована со стоимостью закрываемой информации.

Криптостойкость шифра является основным показателем его эффективности. Она измеряется временем или стоимостью средств, необходимых для криптоанализа при получении исходной информации по шифротексту, при условии, что ему неизвестен ключ.

Сохранить в секрете широко используемый алгоритм шифрования практически невозможно, поэтому он не должен иметь скрытых слабых мест, которыми могли бы воспользоваться криптоаналитики. Если это условие выполняется, то криптостойкость шифра определяется длиной ключа, так как единственный путь вскрытия зашифрованной информации — перебор комбинаций ключа и выполнение алгоритма расшифрования. Таким образом, время и средства, затрачиваемые на анализ шифра, зависят от длины ключа и сложности алгоритма шифрования.

1.4.4. Классификация криптографических алгоритмов

Все методы шифрования могут быть классифицированы по различным признакам. Эти методы можно подразделить на две основные группы: методы шифрования с симметричными ключами и системы шифрования с открытыми ключами.

Методы шифрования с симметричными ключами включают в себя:

- замены — прямой замены, моноалфавитной замены, методы полиалфавитной замены, модифицированной матрицы шифрования;
- перестановки — простой перестановки, усложненной перестановки по таблице, усложненной перестановки по маршрутам;
- аналитические — подразумевают применение матричной алгебры;
- аддитивные (гаммирование) — предусматривают применение генераторов псевдослучайных чисел;
- комбинированные — подразумевают применение указанных выше методов в различных комбинациях.

В системах шифрования с открытыми ключами выделяют системы RSA, Эль-Гамаля и криптосистемы Мак-Элиса.

Для представления всего диапазона существующих алгоритмов шифрования в этом разделе приведена их классификация (рис. 1.2), а вопросы практической реализации отражены в ходе рассмотрения изложения всей книги в целом.
Было доказано, что в криптографии существуют только два основных типа преобразований — замены и перестановки, все остальные являются лишь комбинацией этих двух типов. Таким образом, есть криптографические алгоритмы, построенные на основе замены, перестановки и объединения этих двух преобразований.

В шифрах перестановки символы открытого текста изменяют свое местоположение. В шифрах замены один символ открытого текста замещается символом зашифрованного текста.

В классической криптографии различают четыре типа шифров замены:

- шифры простой замены. Один символ открытого текста заменяется одним символом зашифрованного текста;
- шифры сложной замены. Один символ открытого текста заменяется одним или несколькими символами зашифрованного текста, например, символ A может быть заменен на C или комбинацию символов P04E;
- шифры блочной замены. Один блок символов открытого текста заменяется блоком закрытого текста, например: последовательность или блок символов ABC могут быть, например, заменены блоком CPT или KAR;
- полифалфавитные шифры замены, в которых к открытому тексту применяются несколько шифров простой замены.

Развитие теории построения алгоритмов шифрования с открытыми ключами, родоначальниками которой стали ученые Уитфилд
Диффи (Whitfield Diffie) и Мартин Хеллман (Martin Hellman), положил начало повсеместному использованию асимметричных алгоритмов шифрования, в которых ключи шифрования и расшифрования различны. В зависимости от применения один из ключей будет открытым, т.е. общедоступным, а другой необходимо хранить в секрете.

Спустя некоторое время симметричные алгоритмы были разделены на два больших класса — блочные и поточные.

1.4.4.1. Общие сведения о потоковых шифрах

Потоковые (поточные) шифры представляют собой разновидность гаммирования и преобразуют открытый текст в шифрованный последовательно по 1 биту. Генератор ключевой последовательности, иногда называемый генератором бегущего ключа, выдает последовательность бит $k_1, k_2, ..., k_i, ...$ Эта ключевая последовательность складывается по модулю 2 с последовательностью бит исходного текста $p_1, p_2, ..., p_i, ...$ для получения шифрованного текста:

$$c_i \oplus p_i \oplus k_i.$$

На приемной стороне шифрованный текст складывается по модулю 2 с идентичной ключевой последовательностью для получения исходного текста:

$$c_i \oplus k_i = p_i \oplus k_i \oplus k_i \oplus p_i.$$

Стойкость системы целиком зависит от внутренней структуры генератора ключевой последовательности. Если генератор выдает последовательность с небольшим периодом, то стойкость системы будет невелика. Напротив, если генератор будет выдавать бесконечную последовательность истинно случайных бит, то получится одноразовый блокнот с идеальной стойкостью.

Реальная стойкость потоковых шифров лежит где-то посередине между стойкостью простой моноалфавитной подстановки и одноразового блокнота. Генератор ключевой последовательности выдает поток битов, который выглядит случайным, но в действительности является детерминированным и может быть в точности воспроизведен на приемной стороне. Чем больше генерируемый поток похож на случайный, тем больше усилий потребуется от криплоаналитика для взлома шифра.

Если каждый раз при включении генератор будет выдавать одну и ту же последовательность, то взлом криптосистемы будет тривиальной задачей. Перехватив два шифрованных текста, злоумыш-
ленник может сложить их по модулю 2 и получить два исходных текста, сложенных также по модулю 2. Такую систему раскрыть очень просто. Если же в руках противника окажется пара исходный текст — шифрованный текст, задача вообще становится тривиальной.

По этой причине все потоковые шифры предусматривают использование ключа. Выход генератора ключевой последовательности зависит от этого ключа. В этом случае простой криptoанализ будет невозможен.

Потоковые шифры наиболее пригодны для шифрования непрерывных потоков данных, например, в сетях передачи данных.

Структуру генератора ключевой последовательности можно представить в виде конечного автомата с памятью, состоящего из трех блоков: блока памяти, хранящего информацию о состоянии генератора; выходной функции, генерирующей бит ключевой последовательности в зависимости от состояния; функции переходов, задающей новое состояние, в которое перейдет генератор на следующем шаге.

1.4.4.2. Метод замены

Шифрование методом замены (подстановки) основано на алгебраической операции, называемой подстановкой. Подстановкой называется взаимно-однозначное отображение некоторого конечного множества M на себя. Число N элементов этого множества называется степенью подстановки. Природа множества M роли не играет, поэтому можно считать, что

\[ M = \{1, 2, \ldots, N\}. \]

Если при данной подстановке S число j переходит в i, то подстановка обозначается символом S:

\[ S = \begin{bmatrix} 1 & 2 & n \\ I_1 & I_2 & I_n \end{bmatrix}. \]

В этой записи числа 1, 2, ..., n можно произвольным образом переставлять, соответственно переставляя числа I_1, I_2, ..., I_n. Результат последовательного выполнения двух подстановок S_1 и S_2 одной и той же степени также является подстановкой, которая называется произведением подстановок S_1 и S_2 и обозначается S_1S_2.

Пусть S — произвольная подстановка степени n. Если для некоторого j число I_j отлично от j, то говорят, что подстановка S действительно перемещает число j; в противном случае подстановка S оставляет число j на месте.
Количество $m$ чисел, действительно перемещаемых подстановкой $S$, называется длиной цикла подстановки.

Подстановка $S$ называется транспозицией, если существует пара $(i_1,i_2)$ различных элементов из $M$, удовлетворяющих условиям:

$$i_1 = j_2; \quad i_2 = j_2; \quad i_j = j \text{ для каждого } j \in \{ M \setminus \{i_1,i_2\} \}.$$ 

Любая подстановка разлагается в произведение транспозиций.

В криптографии рассматриваются четыре типа подстановки (замены): моноалфавитная, гомофоническая, полиалфавитная и полиграмматная. Далее в примерах, где необходимо, будем использовать кодирование букв русского алфавита, приведенное в табл. 1.1. Знак "-" в табл. 1.1 и далее означает пробел.

<table>
<thead>
<tr>
<th>Буква</th>
<th>А</th>
<th>Б</th>
<th>В</th>
<th>Г</th>
<th>Д</th>
<th>Е</th>
<th>Ж</th>
<th>З</th>
<th>И</th>
<th>Й</th>
<th>К</th>
<th>Л</th>
<th>М</th>
<th>Н</th>
<th>О</th>
<th>П</th>
<th>Р</th>
</tr>
</thead>
<tbody>
<tr>
<td>Код</td>
<td>01</td>
<td>02</td>
<td>03</td>
<td>04</td>
<td>05</td>
<td>06</td>
<td>07</td>
<td>08</td>
<td>09</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>Буква</td>
<td>С</td>
<td>Т</td>
<td>У</td>
<td>Ф</td>
<td>Х</td>
<td>Ц</td>
<td>Ч</td>
<td>Ш</td>
<td>Щ</td>
<td>Ъ</td>
<td>Ь</td>
<td>Э</td>
<td>Ю</td>
<td>Я</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Код</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td></td>
</tr>
</tbody>
</table>

При моноалфавитной замене каждой букве алфавита открытого текста ставится в соответствие одна буква шифротекста из этого же алфавита.

Пример 1.1.
Открытый текст: Шифрование_заменой
Шифрование подстановкой представлено в табл. 1.2.

<table>
<thead>
<tr>
<th>Шифр подстановки</th>
</tr>
</thead>
<tbody>
<tr>
<td>ИТ</td>
</tr>
<tr>
<td>ШТ</td>
</tr>
<tr>
<td>ИТ</td>
</tr>
<tr>
<td>ШТ</td>
</tr>
</tbody>
</table>

**ИТ** — алфавит исходного текста;

**ШТ** — алфавит шифротекста.

Шифротекст: ИшМRTЮ_У$pяI$Щ_ФyУТч

Основным недостатком рассмотренного метода является сохранение статистических свойств открытого текста (частота повторения букв) в шифротексте.

Общую формулу моноалфавитной замены можно представить в виде:
\[ Y_i = k_1X_i + k_2 \pmod{N}, \]
где \( Y_i \) — \( i \)-й символ алфавита; \( k_1 \) и \( k_2 \) — константы; \( X_i \) — \( i \)-й символ открытого текста (номер буквы в алфавите); \( N \) — длина используемого алфавита.

Шифр, задаваемый формулой:
\[ Y_i = X_i + k_i \pmod{N}, \]
где \( k_i \) — \( i \)-я буква ключа, в качестве которого используются слово или фраза, называется шифром Виженера.

Пример 1.2.
Открытый текст: ЗАМЕНА
Ключ: КЛЮЧ
Шифрование с использованием алгоритма Виженера представлено в табл. 1.3

| Шифр Виженера | | | | |
|---|---|---|---|
| Открытый текст | Ключ | Преобразование | Шифр |
| З | К | \( y_1 = 8 + 11 \pmod{33} = 19 \) | Т |
| А | Л | \( y_2 = 1 + 12 \pmod{33} = 13 \) | М |
| М | Ю | \( y_3 = 13 + 31 \pmod{33} = 11 \) | К |
| Е | Ч | \( y_4 = 6 + 24 \pmod{33} = 30 \) | Э |
| Н | К | \( y_5 = 14 + 11 \pmod{33} = 25 \) | Ш |
| А | Л | \( y_6 = 1 + 12 \pmod{33} = 13 \) | М |

Шифротекст: ТМКЭШМ

Шифр Беффорта — многоалфавитная криптосистема, аналогичная криптосистеме Виженера. Строками квадрата Беффорта являются строки квадрата Виженера, записанные в обратном порядке. Криптосистема названа в честь адмирала Френсиса Беффорта. Шифр Беффорта использует формулу
\[ y_i = k_i - x_i \pmod{n} \quad \text{и} \quad y_i = x_i - k_i \pmod{n}. \]

Гомофоническая замена одному символу открытого текста ставит в соответствие несколько символов шифротекста. Этот метод применяется для искажения статистических свойств шифротекста.

Пример 1.3.
Открытый текст: ЗАМЕНА.
Шифрование с использованием гомофонической подстановки представлено в табл. 1.4.
<table>
<thead>
<tr>
<th>Гомофоническая подстановка</th>
<th>А</th>
<th>Б</th>
<th>...</th>
<th>Е</th>
<th>Ж</th>
<th>З</th>
<th>...</th>
<th>М</th>
<th>Н</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Алфавит открытого текста</td>
<td>17</td>
<td>23</td>
<td></td>
<td>97</td>
<td>47</td>
<td>76</td>
<td></td>
<td>32</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>Алфавит шифротекста</td>
<td>31</td>
<td>44</td>
<td></td>
<td>51</td>
<td>67</td>
<td>19</td>
<td></td>
<td>28</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>63</td>
<td></td>
<td>15</td>
<td>33</td>
<td>59</td>
<td></td>
<td>61</td>
<td>34</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Шифротекст: 76 17 32 97 55 31
Таким образом, при гомофонической замене каждая буква открытого текста заменяется по очереди цифрами соответствующего столбца.

Полиалфавитная подстановка использует несколько алфавитов шифротекста.
Пусть используется $k$ алфавитов. Тогда открытый текст

$$X = X_1X_2...X_kX_{k+1}...X_{2k}X_{2k+1}...$$

заменяется шифротекстом

$$Y = F_1(X_1) F_2(X_2) ... F_k(X_k) F_{k+1}(X_{k+1}) ... F_{2k}(X_{2k}) F_{2k+1}(X_{2k+1})$$

где $F_i(X_i)$ — символ шифротекста алфавита $i$ для символа открытого текста $X_i$.

Полиграммная замена формируется из одного алфавита с помощью специальных правил.
В качестве примера рассмотрим шифр Плэйфера.
В этом шифре алфавит располагается в матрице. Открытый текст разбивается на пары символов $X_iX_{i+1}$. Каждая пара символов открытого текста заменяется на пару символов из матрицы следующим образом:
• если символы находятся в одной строке, то каждый из символов пары заменяется на стоящий правее его (за последним символом в строке следует первый);
• если символы находятся в одном столбце, то каждый символ пары заменяется на символ, расположенный ниже его в столбце (за последним нижним символом следует верхний);
• если символы пары находятся в разных строках и столбцах, то они считаются противоположными углами прямоугольника. Символ, находящийся в левом углу, заменяется на символ, стоящий в другом левом углу; замена символа, находящегося в правом углу, осуществляется аналогично;
• если в открытом тексте встречаются два одинаковых символа подряд, то перед шифрованием между ними вставляется специальный символ (например, тире).
Пример 1.4.
Открытый текст: ШИФР_ПЛЭЙФЕРА
Матрица алфавита представлена в табл. 1.5.

Таблица 1.5

<table>
<thead>
<tr>
<th>А</th>
<th>Х</th>
<th>Б</th>
<th>М</th>
<th>Ц</th>
<th>В</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ч</td>
<td>Г</td>
<td>Н</td>
<td>Ш</td>
<td>Д</td>
<td>О</td>
</tr>
<tr>
<td>Е</td>
<td>Щ</td>
<td></td>
<td>Ж</td>
<td>У</td>
<td>П</td>
</tr>
<tr>
<td>.</td>
<td>З</td>
<td>Ъ</td>
<td>Р</td>
<td>И</td>
<td>Й</td>
</tr>
<tr>
<td>С</td>
<td>Ъ</td>
<td>К</td>
<td>Э</td>
<td>Т</td>
<td>Л</td>
</tr>
<tr>
<td>Ю</td>
<td>Я</td>
<td></td>
<td></td>
<td>Б</td>
<td>Ф</td>
</tr>
</tbody>
</table>

Шифротекст: ПДИЫ, -СТ- И.ЖЧС
При рассмотрении данных видов шифров становится очевидным, что чем больше длина ключа, тем лучше шифр. Существенного улучшения свойств шифротекста можно достигнуть при использовании шифров с автоключом.

Шифр, в котором сам открытый текст или получающаяся криптограмма используются в качестве ключа, называется шифром с автоключом. Шифрование в этом случае начинается с ключа, называемого первичным, и продолжается с помощью открытого текста или криптограммы, смещенной на длину первичного ключа.

Пример 1.5.
Открытый текст: ШИФРОВАНИЕ ЗАМЕНОЙ
Первичный ключ: КЛЮЧ.
Шифрование с автоключом при использовании открытого текста представлено в табл. 1.6.

Таблица 1.6

| ИТ | Ш | И | Ф | Р | О | В | А | Н | И | Е | З | А | М | Е | Н | О | Й |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Кл | К | Л | Ю | Ч | Ш | И | Ф | Р | О | В | А | Н | И | Е | З | А | М |
| Код | 03 | 21 | 19 | 08 | 07 | 12 | 22 | 31 | 24 | 09 | 01 | 22 | 10 | 19 | 06 | 22 | 16 | 23 |
| ШТ | В | Ф | Т | З | Ж | Л | Ю | Ч | И | А | Х | Й | Т | Е | Х | П | Ц |

**ИТ** — алфавит исходного текста;
**Кл** — ключ;
**ШТ** — алфавит шифротекста.
Шифрование с автоключом при использовании криптограммы представлено в табл. 1.7.
Шифрование с автоключом при использовании криптограммы

<table>
<thead>
<tr>
<th>ИТ</th>
<th>Ш</th>
<th>И</th>
<th>Ф</th>
<th>Р</th>
<th>О</th>
<th>В</th>
<th>А</th>
<th>Н</th>
<th>Е</th>
<th>З</th>
<th>А</th>
<th>М</th>
<th>Е</th>
<th>Н</th>
<th>О</th>
<th>Й</th>
</tr>
</thead>
<tbody>
<tr>
<td>Кл</td>
<td>К</td>
<td>Л</td>
<td>Ю</td>
<td>Ч</td>
<td>В</td>
<td>Ф</td>
<td>Т</td>
<td>З</td>
<td>С</td>
<td>Ч</td>
<td>У</td>
<td>Х</td>
<td>Ь</td>
<td>Э</td>
<td>У</td>
<td>Ь</td>
</tr>
<tr>
<td>Код</td>
<td>03</td>
<td>21</td>
<td>19</td>
<td>08</td>
<td>18</td>
<td>24</td>
<td>20</td>
<td>22</td>
<td>27</td>
<td>30</td>
<td>20</td>
<td>30</td>
<td>28</td>
<td>10</td>
<td>26</td>
<td>11</td>
</tr>
<tr>
<td>ШТ</td>
<td>В</td>
<td>Ф</td>
<td>Т</td>
<td>З</td>
<td>С</td>
<td>Ч</td>
<td>У</td>
<td>Х</td>
<td>Ь</td>
<td>Э</td>
<td>У</td>
<td>Э</td>
<td>Ь</td>
<td>Й</td>
<td>Ш</td>
<td>К</td>
</tr>
</tbody>
</table>

ИТ — алфавит исходного текста;
Кл — ключ;
ШТ — алфавит шифротекста.

1.4.4.3. Гаммирование

Различают две разновидности алгоритмов гаммирования: метод конечной гаммы и метод бесконечной гаммы. В качестве конечной гаммы может использоваться фраза, а в качестве бесконечной — последовательность, вырабатываемая датчиком псевдослучайных чисел.

Принцип зашифрования заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел (ПСЧ) и наложении полученной гаммы на открытые данные обратным образом (например, при использовании логической операции исключающее ИЛИ).

Процесс расшифрования данных сводится к повторной генерации гаммы шифра при известном ключе и наложению такой гаммы на зашифрованные данные. Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, когда гамма шифра не содержит повторяющихся битовых последовательностей. По сути дела, гамма шифра должна изменяться случайным образом для каждого шифруемого слова. Фактически если период гаммы превышает длину всего зашифрованного текста и неизвестна никакая часть исходного текста, то шифр можно раскрыть только прямым перебором (подбором ключа). В этом случае криптостойкость определяется размером ключа.

Чтобы получить линейные последовательности элементов гаммы, длина которых превышает размер шифруемых данных, используются датчики ПСЧ. На основе теории групп было разработано несколько типов таких датчиков. В настоящее время наиболее доступными и эффективными являются конгруэнтные генераторы ПСЧ. Они вырабатывают последовательности псевдослучайных чисел $T(i)$, описываемые соотношением

$$T(i + 1) = (A \cdot T(i) + C) \mod M,$$

где $A$ и $C$ — константы; $T(i)$ — исходная величина, выбранная в качестве порождающего числа.
Для шифрования данных с помощью датчика ПСЧ 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 $H(j)$:

$$G = H(1) \cup H(2) \cup ... \cup H(N),$$

где $H(j)$ — множество соответствующих $j$-му сегменту данных и полученных на основе порождающего числа $X(j)$, определенного как функция от $X(j)$ (например, ПСЧ, полученное на основе $X(j)$).

Разумеется, возможны и другие, более изощрённые варианты выбора порождающих чисел для гаммы шифра. Более того, гамму шифра необязательно рассматривать как объединение непересекающихся множеств. Например, гамма шифра может быть представлена в виде

$$G = H(1) (+) H(2) (+) ... (+) H(N),$$

где символ (+) обозначает операцию исключающее ИЛИ.

Шифрование с помощью датчика ПСЧ является довольно распространённым криптографическим методом, а качество шифра определяется не только и не столько характеристиками датчика, сколько алгоритмом получения гаммы. Хорошие результаты дает метод гаммирования с обратной связью, который заключается в том, что для получения сегмента гаммы используется контрольная сумма определенного участка шифруемых данных.

1.5. Аппаратная и программная реализация алгоритмов шифрования

На практике криптографические алгоритмы в зависимости от области применения имеют несколько типов реализации: программную, аппаратную и программно-аппаратную. Перед тем как перейти непосредственно к рассмотрению достоинств и недостатков перечисленных типов реализации, сформулируем общие требования к реализации криптографических алгоритмов.

Современные алгоритмы шифрования должны удовлетворять следующим условиям:

- должны быть адаптированы к новейшей программно-аппаратной базе (например, алгоритмы блочного шифрования в программной реализации должны быть адаптированы к операциям с 64-разрядными числами);
• объем ключа должен соответствовать современным методам и средст вам дешифрования зашифрованных сообщений (о минимальной длине ключа будет сказано позже);
• операции зашифрования и расшифрования должны по возможности быть простыми, чтобы удовлетворять современным требованиям по скоростным характеристикам;
• не должны допускать появления постоянно увеличивающегося числа ошибок;
• должны сводить к минимуму объем сообщения в ходе выполнения операций шифрования.

1.5.1. Аппаратная реализация криптографических алгоритмов

До недавних пор алгоритмы шифрования реализовывались в виде отдельных устройств, что обусловливалось использованием криптографии для засекречивания различных видов передачи информации (телеграф, телефон, радиосвязь). С развитием средств вычислительной техники и общедоступных сетей передачи данных появились новые возможности применения криптографических алгоритмов. Однако аппаратная реализация до сих пор широко используется не только в военной сфере, но и в коммерческих организациях. Подобная живучесть аппаратных средств криптографической защиты информации объясняется рядом факторов:
• во-первых, аппаратная реализация обладает лучшими скоростными характеристиками, нежели программно-реализуемые алгоритмы шифрования. Использование специальных чипов, адаптированных к реализации на них процедур зашифрования и расшифрования, приводит к тому, что в отличие от процессоров общего назначения они позволяют оптимизировать многие математические операции, применяемые в алгоритмах шифрования;
• во-вторых, аппаратные средства защиты информации обладают несравнимо большей защищенностью как от побочных электромагнитных излучений, возникающих в ходе работы аппаратуры, так и от непосредственного физического воздействия на устройства, где осуществляются операции шифрования и хранение ключевой информации;
• в-третьих, аппаратные средства более удобны в эксплуатации, так как позволяют осуществлять операции зашифрования и расшифрования для пользователя в прозрачном режиме; кроме того, их легко инсталировать;
* в-четвертых, учитывая многообразие вариантов применения средств криптографической защиты информации, аппаратные средства повсеместно используются для защиты телефонных переговоров, отправки факсимильных сообщений и других видов передачи информации, где невозможно использовать программные средства.

1.5.2. Программная реализация криптографических алгоритмов

К достоинствам программной реализации можно отнести ее гибкость и переносимость. Другими словами, программа, написанная под одну операционную систему, может быть модифицирована под любой тип операционной системы (ОС). Кроме того, обновить программное обеспечение можно с меньшими временными и финансовыми затратами. К тому же многие современные достижения в области криптографических протоколов недоступны для реализации в виде аппаратных средств.

К недостаткам программных средств криптографической защиты следует отнести возможность вмешательства в действие алгоритмов шифрования и получения доступа к ключевой информации, хранившейся в общедоступной памяти. Эти операции обычно выполняются при помощи простого набора программных инструментариев (отладчики программ и т.д.). Так, например, во многих операционных системах осуществляется аварийный дамп памяти на жесткий диск; при этом в памяти могут находиться ключи, найти которые не составит труда.

Таким образом, слабая физическая защищенность программных средств является одним из основных недостатков подобных методов реализации алгоритмов шифрования.

К этому можно добавить, что программная реализация средств криптографической защиты не в состоянии обеспечить выполнение некоторых 2—3 характеристик, требуемых для надежного использования алгоритмов шифрования. Например, генерация ключевой информации не должна производиться программными датчиками случайных чисел; для этой цели необходимо использовать специальные аппаратные устройства.

1.5.3. Программно-аппаратная реализация криптографических алгоритмов

Программно-аппаратная реализация позволяет пользователям устранить некоторые недостатки программных средств защиты ин-
формации и при этом сохранить их достоинства (за исключением ценового критерия).

Основными функциями, возлагаемыми на аппаратную часть программно-аппаратного комплекса криптографической защиты информации, обычно являются генерация ключевой информации и хранение ключевой информации в устройствах, защищенных от несанкционированного доступа со стороны злоумышленника. Кроме того, посредством методик такого типа можно осуществлять аутентификацию пользователей с помощью паролей (статических или динамически изменяемых, которые могут храниться на различных носителях ключевой информации — смарт-картах, touch-memory и т.д.) либо на основе уникальных для каждого пользователя биометрических характеристик. Устройства считывания подобных сведений могут входить в состав программно-аппаратной реализации средств защиты информации.

Контрольные вопросы

1. Назовите основные положения Доктрины национальной безопасности РФ.
2. Назовите основные задачи по обеспечению информационной безопасности в свете Доктрины национальной безопасности РФ.
3. Дайте определение понятия «информационная безопасность».
4. Перечислите основные задачи защиты пользовательской информации.
5. Перечислите основные свойства информации с точки зрения информационной безопасности.
6. Перечислите основные угрозы информационной безопасности.
7. В чем суть угрозы конфиденциальности информации?
8. В чем суть угрозы целостности информации?
9. В чем суть угрозы доступности информации?
10. Дайте определение понятия криптографического протокола.
11. Перечислите основные понятия и определения классической криптографии.
12. Дайте определение понятия стойкости криптографического алгоритма.
13. Что такое методы криптографической защиты информации?
14. Что такое методы шифрования (дешифрования) информации?
15. На какие группы по виду воздействия на исходную информацию подразделяют методы криптографической защиты информации?
16. Что такое атака на шифр?
17. Каким требованиям должны отвечать современные методы шифрования?
18. Что такое стойкость шифра?
19. Какая характеристика шифра является основным показателем его эффективности?
20. На какие две основные группы подразделяются все методы шифрования?
21. Что включают в себя методы шифрования с симметричными ключами?
22. Какие два основных типа преобразований существуют в криптографии?
23. В чем суть методов преобразований замены и перестановки?
24. Какие четыре типа шифров замены различают в классической криптографии?
25. На какие основные классы подразделяются симметричные алгоритмы?
26. Представителями какого класса алгоритмов являются блочные и поточные алгоритмы?
27. Что такое потоковые шифры?
28. Что такое метод замены?
29. Какие существуют четыре основных метода замены?
30. Что такое метод гаммирования?
31. Какие различают две разновидности алгоритмов гаммирования?
32. Какой механизм генерации гаммы лежит в основе зашифрования методом гаммирования?
33. В чем суть аппаратной реализации криптографических алгоритмов?
34. В чем суть программной реализации криптографических алгоритмов?
35. В чем суть программно-аппаратной реализации криптографических алгоритмов?
Тема 2

ШИФРЫ ЗАМЕНЫ

Напомним ряд базовых понятий и определений криптографической защиты информации.

Шифр — это метод или способ преобразования (шифрования) информации с целью ее защиты от незаконных пользователей.

Дешифрование — процесс, обратный шифрованию, т.е. преобразование шифрованного сообщения в защищаемую информацию с помощью определенных правил, содержащихся в шифре. Защищаемая информация называется открытым текстом; полученное в результате шифрования сообщение называется шифротекстом или криптограммой.

Опираясь на приведенную в теме 1 классификацию алгоритмов криптографической защиты информации рассмотрим первую группу шифров на примере шифров простой замены. К таким шифрам можно отнести систему шифрования Цезаря, аффинную систему подстановок Цезаря, лозунговый шифр, шифр полибийанский квадрат, шифрующую таблицу Трисемуса.

Шифрами замены называют шифры, в которых преобразование информации осуществляется путем замены каждого символа открытого сообщения на символы шифрообозначения. Формально такая замена выглядит следующим образом. Для каждой буквы α исходного алфавита строится некоторое множество символов Mα, которое называется множеством шифрообозначений для буквы α. При использовании такой замены можно осуществить как зашифрование, так и расшифрование сообщения. При зашифровании каждый символ α открытого сообщения, начиная с первого, заменяется любым символом из множества Mα. Выбирая различные виды замен из множества зашифрования, можно получить различные варианты зашифрованного сообщения для одного и того же открытого сообщения. Проиллюстрируем данный вид шифрования таблицей соответствия символов исходного алфавита и множества шифрообозначений (табл. 2.1).

Шифры простой замены являются примером простейшей реализации алгоритмов замены. В данном случае множество шифрообозначений Mα состоит из одного элемента. Такие шифры принято называть шифрами одноалфавитной подстановки.
Таблица 2.1

<table>
<thead>
<tr>
<th>A</th>
<th>Б</th>
<th>...</th>
<th>Я</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_A$</td>
<td>$M_B$</td>
<td>...</td>
<td>$M_Y$</td>
</tr>
</tbody>
</table>

2.1. Шифры простой замены

2.1.1. Система шифрования Цезаря

В качестве ключа шифра Цезаря используют таблицу, первая строка которой содержит символы исходного алфавита, вторая строка — последовательность символов, записанных в алфавитном порядке с любой буквы, кроме первой, циклично. Данное соответствие отображается табл. 2.2.

Таблица 2.2

<table>
<thead>
<tr>
<th>А</th>
<th>Б</th>
<th>...</th>
<th>Е</th>
<th>Ю</th>
<th>Я</th>
</tr>
</thead>
<tbody>
<tr>
<td>Д</td>
<td>Е</td>
<td>...</td>
<td>Б</td>
<td>В</td>
<td>Г</td>
</tr>
</tbody>
</table>

При зашифровании букву исходного сообщения находят в первой строке таблицы и заменяют ее соответствующей буквой второй строки. Запомнить такой ключ достаточно просто — надо лишь знать первую букву второй строки.

На рис. 2.1 показаны интерфейс и код программы, реализующие пример алгоритма шифра замены:

Рис. 2.1. Окно программы, реализующей алгоритм шифра замены
Код программы:

//Массив abc содержит значения от "а" до "я"
//Массив cba содержит значения символов в алфавитном //порядке, но не с символа "а".
for i:=0 to 34 do
    //заполняется 1-я строка таблицы
    StringGrid1.Cells[i,0]:=abc[i];
    //заполняется массив
    cba[i]:=abc[i];
for i:=k to 34 do
    //заполняется 2-я строка таблицы
    StringGrid1.Cells[i-k,1]:=abc[i];
    //заполняется массив
    cba[i-k]:=abc[i];
end
for i:=34-k to 34 do
begin
    //заполняется 2-я строка таблицы
    StringGrid1.Cells[i-1,1]:=abc[i-34+k];
    //заполняется массив
    cba[i-1]:=abc[i-34+k];
end
for i=1 to Edit1.GetTextLen()+1 do
for j:=0 to 34 do
    //проверяется входжение букв в слово
    if (abc[j]=Edit1.Text[i])
    //зашифровывается исходное слово
    Edit2.Text:=Edit2.Text+cba[j];

2.1.2. Аффинная система подстановок Цезаря

В аффинной системе подстановок Цезаря символы исходного сообщения преобразуются в соответствии с формулой

$$T_1 = (AT + B) \mod m,$$

где $T$ — порядковый номер символа исходной последовательности; $T_1$ — порядковый номер соответствующего символа зашифрованной последовательности; $m$ — размер алфавита; $A$, $B$ — целые числа, причем $A$ и $m$ взаимно простые.

Рассмотрим пример при $A = 13$, $B = 5$, размере алфавита $m = 32$ (используется символы русского алфавита, где вместо символа Й используется І, вместо Є — Е и добавляется пробел как 32-й символ последовательности).
Для исходного сообщения

**КОРАБЛИ ОТПЛЫВАЮТ ВЕЧЕРОМ**

шифротекст будет выглядеть так:

**ЫПИЕУЗОШПВЪЗШ ЕЯВШ ЖЖИПХ.**

На рис. 2.2 показан интерфейс программы, реализующей алгоритм аффинной системы подстановок Цезаря.

Рис. 2.2. Окно программы, реализующей аффинную систему подстановок Цезаря

Код программы:

```c
//Массив abc содержит значения от "а" до "я" и символ
//"пробел", не содержит "ё" и "й"
a:=strtoint(edit4.text);
b:=strtoint(edit5.text);
m:=32;
tocode:=edit1.text;
for i:=1 to length(tocode) do
  begin
    //зашифровывается символ
    n:=(i*a+b)modm;
    //формируется зашифрованное слово
    edit2.Text:=edit2.Text+abc[n];
  end;
```

**Достоинством** аффинной системы является удобное управление ключами — ключи шифрования и расшифрования представляются в компактной форме в виде пары чисел \((a, b)\).

**Недостатками** рассмотренных шифров Цезаря можно считать следующие особенности алгоритмов:
• подстановки, выполняемые в соответствии с системой Цезаря, не маскируют частот появления различных букв исходного открытого текста;
• сохраняется алфавитный порядок в последовательности заменяющихся букв;
• при изменении значения величины сдвига первого символа шифрующей строки изменяются только начальные позиции такой последовательности;
• число возможных ключей шифрования мало;
• шифр Цезаря легко вскрывается на основе анализа частот появления букв в шифротексте.

Анализ криптостойкости системы одноалфавитной замены начинается с подсчета частот появления символов: определяется число появления каждого символа в шифротексте. Затем полученное распределение частот символов в шифротексте сравнивается с распределением частот символов в алфавите исходных сообщений. Символ с наивысшей частотой появления в шифротексте заменяется символом с наивысшей частотой появления в алфавите и т.д. Вероятность успешного вскрытия системы шифрования повышается с увеличением длины шифротекста.

Основные идеи системы шифрования Цезаря плодотворно реализуются в модификациях современных алгоритмов и иллюстрациях основных криптологических положений.

2.1.3. Лозунговый шифр

В данном шифре ключом шифрования выступает так называемое слово-лозунг, в качестве которого используются легко запоминающиеся слово или фраза. В качестве примера реализации шифра рассмотрим слово-лозунг ЗАЯВЛЕНИЕ. Процесс шифрования выполняется по следующему правилу. В первую строку шифрующей таблицы записывается исходный алфавит. Вторая строка формируется следующим образом. Вначале вписывается слово-лозунг, при этом повторяющиеся символы отбрасываются, затем строка дополняется не вошедшими в нее символами исходного алфавита. Шифрующая таблица будет иметь вид табл. 2.3.

Таблица. 2.3

Пример шифрующей таблицы лозунгового шифра

| А | Б | В | Г | Д | Е | Ж | З | И | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Ъ | Ы | Ь | Э | Ю | Я |
| ЗАЯВЛЕНИЕ | Б | Г | Д | Ж | К | М | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Ъ | Ы | Ь | Э | Ю | Я |
На рис. 2.3 показан интерфейс программы, реализующей алгоритм лозунгового шифра.

Рис. 2.3. Окно программы, реализующей алгоритм лозунгового шифра

Код программы:

```plaintext
//talpa содержит строку букв алфавита
//переменной lozung присваивается значение строки
//букв алфавита
lozung:=edit4.Text;
//проходятся все буквы lozung
for i:=1 to length(lozung) do
//проверяется, встречается ли соответствующая
//буква в строке
if pos(lozung[i], tstroka)=0 then
//конкатенируем эту букву со значением строки
tstroka:=tstroka+lozung[i];
for i:=1 to length(talpa) do
  if pos(talpa[i], tstroka)=0 then
    tstroka:=tstroka+talpa[i];
edit6.Text:=tstroka;
//формируется ключ таблицы
for i:=0 to length(talpa)-1 do
  StringGrid1.Cells[i,0]:=talpa[i+1];
for i:=0 to length(tstroka)-1 do
  StringGrid1.Cells[i,1]:=tstroka[i+1];
tocode:=edit1.Text;
//прохождение по исходному тексту
for i:=1 to length(tocode) do
  edit2.Text:=edit2.Text+tstroka[pos(tocode[i], talpa)];
```

43
2.1.4. Полибианский квадрат

В данном шифре процесс шифрования выполняется на базе шифрующей таблицы. Таблица прямоугольной формы заполняется символами исходного алфавита случайным образом. Каждый символ открытого сообщения заменяется символом, расположенном ниже в том же столбце. Если символ находится на последней строке таблицы, то он заменяется верхним символом столбца. В табл. 2.4 приведен пример шифрующей таблицы алгоритма полибианский квадрат.

Таблица 2.4

<table>
<thead>
<tr>
<th>У</th>
<th>К</th>
<th>В</th>
<th>Ъ</th>
<th>М</th>
<th>Ю</th>
<th>Ь</th>
<th>Д</th>
</tr>
</thead>
<tbody>
<tr>
<td>И</td>
<td>Б</td>
<td>Т</td>
<td>Л</td>
<td>Э</td>
<td>Г</td>
<td>Щ</td>
<td>Н</td>
</tr>
<tr>
<td>С</td>
<td>Ф</td>
<td>З</td>
<td>Ы</td>
<td>П</td>
<td>Ц</td>
<td>Е</td>
<td>Я</td>
</tr>
<tr>
<td>А</td>
<td>Р</td>
<td>Х</td>
<td>Ж</td>
<td>Ш</td>
<td>О</td>
<td>Ч</td>
<td></td>
</tr>
</tbody>
</table>

Рассмотрим пример реализации алгоритма полибианский квадрат, зашифровывая исходное слово АЛФАВИТ.

Код программы:

```pascal
procedure TForm1.Button1Click(Sender: TObject);
var
  s, tocode: String;
begin
  tocode := edit1.Text;

  // просматривается каждый символ текста
  for b := 1 to length(edit1.Text) do
    begin
      for c := 0 to 5 do
        for d := 0 to 5 do
          // проверяется равенство очередного символа символу
          // ячейки с номером [c, d]
          if tocode[b] = table1.Cells[c, d] then
            begin
              // проверяется, находится ли символ на последней строке
              if d + 1 = 6 then
                // зашифровывается символом того же столбца 1-й строки
                s := s + table1.Cells[c, 0]
              // зашифровывается символом, находящимся
              // строкой ниже, в том же столбце
              else s := s + table1.Cells[c, d + 1];
            end;
        end;
    end;
end;
```

44
На рис. 2.4 показан интерфейс программы, реализующей алгоритм шифра полибианский квадрат.

Рис. 2.4. Окно программы, реализующей алгоритм шифрования полибианский квадрат

2.1.5. Шифрующая таблица Трисемуса

В данном алгоритме процесс шифрования выполняется на основе шифрующей таблицы и ключевого слова или фразы. В шифрующую таблицу вписывается ключевое слово, при этом повторяющиеся символы отбрасываются. Далее шифрующая таблица дополняется оставшимися символами исходного алфавита.

Пример реализации алгоритма шифрующей таблицы Трисемуса с ключевым словом БАНДЕРОЛЬ представлен в табл. 2.5.

| Пример алгоритма шифрующей таблицы Трисемуса |
|-------|-------|-------|-------|-------|-------|-------|-------|
| \( Б \) | \( А \) | \( Н \) | \( Д \) | \( Е \) | \( Р \) | \( О \) | \( Л \) |
| \( Ь \) | \( В \) | \( Г \) | \( Ж \) | \( З \) | \( И \) | \( К \) | \( М \) |
| \( П \) | \( С \) | \( Т \) | \( У \) | \( Ф \) | \( Х \) | \( Ц \) | \( Ч \) |
| \( Ш \) | \( Щ \) | \( Ъ \) | \( Ы \) | \( Э \) | \( Ю \) | \( Я \) |

Последовательность шагов шифрования на основе шифрующей таблицы Трисемуса аналогична применению алгоритма полибианского квадрата.

На рис. 2.5 показан интерфейс программы, реализующей алгоритм шифрующей таблицы Трисемуса.

Код программы:

tocode:= edit1.Text; // исходное слово
Рис. 2.5. Окно программы, реализующей алгоритм шифрующей таблицы Трисемуса

```pascal
// прохождение всех букв исходного текста
for b:=1 to length(tocode) do
  begin
    for c:=0 to 7 do // перебор всех столбцов
      for d:=0 to 3 do // перебор всех строк
        // проверяется равенство очередной буквы
        // исходного текста букве сетки
        if tocode[b]=table1.Cells[c, d] then begin
          // проверяется, находится ли найденная буква на
          // последней строке
          if d+1=4 then
            // в зашифрованную строку вставляется буква 1-й
            // строки того же столбца
            s:=s+table1.Cells[c, 0]
            // иначе вставляется буква строкой ниже,
            // того же столбца
            else s:=s+table1.Cells[c, d+1];
          end;
        end;
    edit2.Text:=s;
```  

2.1.6. Биграммный шифр Плейфера

Ручная симметричная техника шифрования, в которой впервые использована замена биграмм. Шифр предусматривает шифрование
пар символов (биграмм) вместо одиночных символов, как в шифре подстановки и в более сложных системах шифрования Вигнера. Таким образом, шифр Плейфера более устойчив к взлому по сравнению с шифром простой замены, так как затрудняется частотный анализ. Отличительной особенностью частотного анализа для данного алгоритма является сравнение не частот появления символов, в частности 26 символов латинского алфавита, а частот биграмм. Эта процедура сложнее на порядок, так как исследуется 26 × 26 = 676 возможных биграмм. Конечно, анализ частоты биграмм возможен, но это более трудоемкий процесс, требующий большого объема зашифрованного текста.

Процедура шифрования с использованием биграммного шифра включает в себя следующие шаги.
1. Открытый текст разбивается на пары символов (биграммы). Текст должен иметь четное число символов, в нем не должно быть биграмм, содержащих два одинаковых символа.
2. Последовательность биграмм открытого текста преобразуется с помощью шифрующей таблицы по следующим правилам:
   • если оба символа биграммы открытого сообщения не попадают в одну строку или столбец, тогда для замены находят символы в углах прямоугольника, определяемого данной парой символов, причем каждый символ находит себе замену в своем столбце;
   • если оба символа биграммы открытого сообщения принадлежат одному столбцу таблицы, то их заменяют символами, лежащими под ними. Если символ открытого текста находится в нижней строке, то для шифрования берется символ из верхней строки того же столбца;
   • если оба символа биграммы открытого сообщения принадлежат одной строке таблицы, они заменяются символами, лежащими справа от них. Если символ открытого текста находится в крайнем правом столбце, для шифрования берется символ крайнего левого столбца той же строки.

Рассмотрим реализацию шифра Плейфера на примере зашифровывания слова ЗАМЕНА.
Вариант шифрующей таблицы алгоритма Плейфера приведен в табл. 2.6.

На рис. 2.6 показан интерфейс программы, реализующей алгоритм шифра Плейфера.
Код программы:

```pascal
procedure TForm1.Button1Click(Sender: TObject);
var
```
Пример шифрующей таблицы алгоритма Плейфера

<table>
<thead>
<tr>
<th></th>
<th>А</th>
<th>Ж</th>
<th>Б</th>
<th>М</th>
<th>Ц</th>
<th>В</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ч</td>
<td>Г</td>
<td>Н</td>
<td>Ш</td>
<td>Д</td>
<td>О</td>
<td></td>
</tr>
<tr>
<td>Е</td>
<td>Щ</td>
<td></td>
<td>Х</td>
<td>У</td>
<td>П</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>З</td>
<td>Ъ</td>
<td>Р</td>
<td>И</td>
<td>Й</td>
<td></td>
</tr>
<tr>
<td>С</td>
<td>Ь</td>
<td>К</td>
<td>Э</td>
<td>Т</td>
<td>Л</td>
<td></td>
</tr>
<tr>
<td>Ю</td>
<td>Я</td>
<td>Ь</td>
<td>Ф</td>
<td>:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Рис. 2.6. Окно программы, реализующей алгоритм шифра Плейфера

```java
New: String;
Dлина_str, Dлина_new: Integer;
SIM1, SIM2: char;
begin
  // В открытом тексте вставляется спец. знак «—»
  // между одинаковыми символами
  New:= '';
  Dлина_str:=Length(edit1.Text);
  for i:=1 to Dлина_str do
    if (edit1.Text[i] = edit1.Text[i+1])
      then New:= (New + edit1.Text[I] + '—')
    else New:= (New + edit1.Text[I]);
  // Добавление спец. знака «—» в конец открытого
  // текста в случае нечетного количества символов
```
2.1.7. Система омофонов

Данная шифрующая система характеризуется тем, что символы исходного сообщения имеют несколько замен. Число таких замен символа пропорционально вероятности его появления в открытом тексте. Данные о распределениях вероятностей букв в русском тексте приведены в табл. 2.7.

Шифруя символ исходного сообщения, выбирают случайным образом одну из замен. Система замен, которую принято называть системой омофонов, представляется трехразрядными числами от 000 до 999. Например, в соответствии с таблицей частота символу О присваиваются 90 случайных номеров, символам Б и В — по 14 номеров.
Таблица 2.7

<table>
<thead>
<tr>
<th>Буква</th>
<th>Вероятность</th>
<th>Буква</th>
<th>Вероятность</th>
<th>Буква</th>
<th>Вероятность</th>
<th>Буква</th>
<th>Вероятность</th>
</tr>
</thead>
<tbody>
<tr>
<td>Пробел</td>
<td>0.175</td>
<td>Р</td>
<td>0.040</td>
<td>Я</td>
<td>0.018</td>
<td>Х</td>
<td>0.009</td>
</tr>
<tr>
<td>О</td>
<td>0.090</td>
<td>В</td>
<td>0.038</td>
<td>Ы</td>
<td>0.016</td>
<td>Ж</td>
<td>0.007</td>
</tr>
<tr>
<td>Е</td>
<td>0.072</td>
<td>Л</td>
<td>0.035</td>
<td>З</td>
<td>0.016</td>
<td>Ю</td>
<td>0.006</td>
</tr>
<tr>
<td>А</td>
<td>0.062</td>
<td>К</td>
<td>0.028</td>
<td>Ь</td>
<td>0.014</td>
<td>Ш</td>
<td>0.006</td>
</tr>
<tr>
<td>И</td>
<td>0.062</td>
<td>М</td>
<td>0.026</td>
<td>Б</td>
<td>0.014</td>
<td>Ц</td>
<td>0.004</td>
</tr>
<tr>
<td>Н</td>
<td>0.053</td>
<td>Д</td>
<td>0.025</td>
<td>Г</td>
<td>0.013</td>
<td>Щ</td>
<td>0.003</td>
</tr>
<tr>
<td>Т</td>
<td>0.053</td>
<td>П</td>
<td>0.023</td>
<td>Ч</td>
<td>0.012</td>
<td>Э</td>
<td>0.003</td>
</tr>
<tr>
<td>С</td>
<td>0.045</td>
<td>У</td>
<td>0.021</td>
<td>Й</td>
<td>0.010</td>
<td>Ф</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Если омофоны присваиваются случайным образом при различном появлении одного и того же символа, то наличие каждого омофона в шифротексте равновероятно. Система омофонов обеспечивает простейшую защиту от криптоаналитических атак, основанных на подсчете частот появления букв в шифротексте.

На рис. 2.7 показан интерфейс программы, реализующей алгоритм шифрующей системы омофонов.

Рис. 2.7. Окно программы, реализующей алгоритм шифрующей системы омофонов

Код программы:

tocode:=ansilowercase(memo1.text);
// буквы преобразуются в нижний регистр
for i:=1 to length(tocode) do
if tocode[i] in z then
    edit2.Text:=edit2.Text+tocode[i];
init;
s:='';
for i:=1 to length(edit2.Text) do
    begin
        curr:=pos(edit2.Text[i], talpha);
        // находится ли данная буква в алфавите
        if (nom[curr]-nuc[curr])=cast[curr]
        // определяется количество допустимых замен
        then nom[curr]:=nuc[curr];
        // если не соответствует, то обнуляется
        s:=s+inttostr(nom[curr])+' ';
        // вместо буквы записывается ее омофон
        // увеличивается значение омофона
        inc(nom[curr]);
        end;
    edit3.Text:=s;
    memo2.Text:=edit3.Text;

Контрольные вопросы

1. Какие шифры называют шифрами простой замены?
2. Что называют множеством шифрообозначений?
3. Приведите примеры шифров простой замены. Опишите алгоритм одного из них.
4. Опишите алгоритм системы шифрования Цезаря.
5. Опишите алгоритм аффинной системы подстановок Цезаря.
6. Опишите алгоритм логоногого шифра.
7. Опишите алгоритм Полибианского квадрата.
8. Опишите алгоритм шифрующая таблицы Трисемуса.
9. Опишите алгоритм биграммного шифра Плейфера.
10. Опишите алгоритм системы омофонов.

2.1.9. Тесты по теме
«Алгоритмы простой замены»

Вариант 1

Формула, которая используется для преобразования букв исходного сообщения в аффинной системе подстановок Цезаря, имеет вид:

а) \( T_1 = (A \cdot T + B) \mod m \), где \( T \) — порядковый номер буквы исходной последовательности; \( T_1 \) — порядковый номер соответствующей буквы зашифрованной последовательности;

б) \( T_1 = (A \cdot T + B) \mod m \), где \( T \) — порядковый номер соответствующей буквы зашифрованной последовательности; \( T_1 \) — порядковый номер буквы исходной последовательности;
в) $T_1 = (A \cdot T - B) \mod m$, где $T$ — порядковый номер буквы исходной последовательности; $T_1$ — порядковый номер соответствующей буквы зашифрованной последовательности.

2. В какую часть множества шифрообозначений записывается слово-лозунг для лозунгового шифра:
   а) в середину;
   б) в конец;
   в) в начало?

3. Чем отличается алгоритм шифрования полибианского квадрата от шифрующей таблицы Трисемуса:
   а) наличием ключевого слова во множестве шифрообозначений у алгоритма полибианского квадрата;
   б) наличием ключевого слова в начале множества шифрообозначений таблицы Трисемуса;
   в) наличием ключевого слова в конце множества шифрообозначений таблицы Трисемуса?

4. В чем сущность алгоритма шифра многоязычной замены (системы омофонов):
   а) каждой букве ставится в соответствие несколько эквивалентов, число которых равняется их номеру в алфавите;
   б) каждой букве ставится в соответствие созвучный эквивалент;
   в) каждой букве ставится в соответствие несколько эквивалентов, число которых пропорционально частоте встречаемости буквы в открытом тексте?

5. Для каких шифров характерно наличие ключевого слова во множестве шифрообозначений:
   а) шифр Цезаря и логунговый шифр;
   б) шифр Цезаря и полибианский квадрат;
   в) логунговый шифр и таблица Трисемуса?

Вариант 2

1. Для какого шифра применяется следующая формула: $T_1 = (A \cdot T + B) \mod m$, где $T$ — порядковый номер соответствующей буквы зашифрованной последовательности; $T_1$ — порядковый номер буквы исходной последовательности; $m$ — размер алфавита; $A$, $B$ — целые числа, причем $A$ и $m$ взаимно простые:
   а) аффинная система подстановок Цезаря;
   б) логунговый шифр;
   в) битраммный шифр Плейфера?

2. Вариацией какого шифра является шифр ROT-13:
   а) логунговый шифр;

52
б) шифрующая таблица Трисемуса;
в) шифр Цезаря?
3. В какую часть множества шифрообозначений записывается слово-лозунг для лозунгового шифра:
а) в середину;
б) в конец;
в) в начало?
4. В чем заключается слабость шифров простой замены:
а) наличие открытого и закрытого ключей;
б) отсутствие закрытого ключа;
в) исходное сообщение легко расшифровать по часто появляющемуся символу?
5. Каждая шифrozамена в системе омофонов должна состоять из:
а) двух цифр, и их общее количество равно 1000;
б) трех цифр, и их общее количество равно 1000;
в) двух цифр, и их общее количество равно 100.

Вариант 3
1. Какой из следующих формул задается шифр Виженера:
а) \( Y_i = X_i + k_i \mod N \);
б) \( Y_i = k_1 X_i + k_2 \mod N \);
в) \( Y_i = K_i - X_i \mod n \)?
2. Главный недостаток шифра Цезаря:
а) трудная реализация алгоритмов шифрования;
б) легко взламывается и не имеет практического применения;
в) наличие открытого и закрытого ключей.
3. На какой символ заменяется символ исходного сообщения в полибианском квадрате:
а) расположенный ниже в том же столбце таблицы;
б) расположенный слева в той же строке таблицы;
в) расположенный выше в том же столбце таблицы?
4. Какого размера понадобится таблица, чтобы закодировать слово на латинском языке шифром Плейфера:
а) 5×6;
б) 5×5;
в) 5×4?
5. От какого вида криптоаналитических атак защищен шифр многозначной замены (системы омофонов):
а) лобовое вскрытие;
б) основанных на тестовых фрагментах подобранного открытого текста;
в) основанных на частотном анализе?

Вариант 4
1. Как осуществляется запись ключа для таблицы Трисемуса:
a) ключ записывается в конце множества шифрообозначений, при этом отбрасываются повторяющиеся буквы;
b) ключ записывается в конце множества шифрообозначений, при этом повторяющиеся буквы не отбрасываются;
v) ключ записывается в начале множества шифрообозначений, при этом отбрасываются повторяющиеся буквы?
2. Шифры, которые позволяют шифровать сразу по две буквы, называются:
a) биполярными;
b) биграммными;
v) билитерными.
3. В чем заключается достоинство шифра Плейфера:
a) наличие открытого и закрытого ключей;
b) использование операций побитового исключения (XOR);
v) позволяет кодировать сразу по две буквы?
4. Омофоны могут быть представлены:
a) трехразрядными числами от 000 до 999;
b) двухразрядными числами от 00 до 99;
v) трехразрядными числами от 111 до 999.
5. Какой метод шифрования защищен от криптоаналитических атак, основанных на частотном анализе:
a) шифр Цезаря;
b) полибианский квадрат;
v) система омофонов?

Вариант 5
1. Для каких шифров характерно наличие ключевого слова во множестве шифрообозначений:
a) шифр Цезаря и лозунговый шифр;
b) шифр Цезаря и полибианский квадрат;
v) лозунговый шифр и таблица Трисемуса?
2. Каждая шифрозамена в системе омофонов должна состоять из:
a) двух цифр, и их общее количество равно 1000;
b) трех цифр, и их общее количество равно 1000;
v) двух цифр, и их общее количество равно 100.
3. От какого вида криптоаналитических атак защищен шифр многозначной замены (системы омофонов):
   а) лобовое вскрытие;
   б) основанных на тестовых фрагментах подобранных открытого текста;
   в) основанных на частотном анализе?
4. Какой метод шифрования защищен от криптоаналитических атак, основанных на частотном анализе:
   а) шифр Цезаря;
   б) полибианский квадрат;
   в) система омофонов?
5. Шифры, которые позволяют шифровать сразу по две буквы, называются:
   а) биполярными;
   б) биграммыными;
   в) билинтерными.

Таблица правильных ответов

<table>
<thead>
<tr>
<th>Вопрос</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>в</td>
<td>в</td>
<td>б</td>
<td>в</td>
<td>б</td>
</tr>
<tr>
<td>2</td>
<td>а</td>
<td>в</td>
<td>в</td>
<td>в</td>
<td>б</td>
</tr>
<tr>
<td>3</td>
<td>в</td>
<td>б</td>
<td>а</td>
<td>в</td>
<td>в</td>
</tr>
<tr>
<td>4</td>
<td>в</td>
<td>б</td>
<td>в</td>
<td>а</td>
<td>в</td>
</tr>
<tr>
<td>5</td>
<td>в</td>
<td>в</td>
<td>в</td>
<td>в</td>
<td>в</td>
</tr>
</tbody>
</table>

2.2. Шифры сложной замены

Шифры сложной замены называются многоалфавитными, так как для шифрования каждого символа исходного сообщения применяют свой шифр простой замены. Многоалфавитная подстановка последовательно циклически меняет используемые алфавиты. При \( r \)-алфавитной подстановке символ \( x_0 \) исходного сообщения заменяется символом \( y_0 \) из алфавита \( B_0 \), символ \( x_1 \) — символом \( y_1 \) из алфавита \( B_1 \) и т.д.

Общая схема многоалфавитной подстановки для случая \( r = 4 \) показана в табл. 2.8.

Таблица 2.8

<table>
<thead>
<tr>
<th>Входной символ</th>
<th>X0</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Алфавит подстановки</td>
<td>B0</td>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B0</td>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B0</td>
<td>B1</td>
</tr>
</tbody>
</table>
2.2.1. Шифр Гронсфельда

При реализации шифра Гронсфельда под каждым символом исходного сообщения записывают цифры числового ключа. Если ключ короче исходного сообщения, то запись цифр ключа циклически повторяют. Для замены выбирают тот символ, который смешен по алфавиту на соответствующую цифру ключа. Например, применяя в качестве ключа группу из четырех начальных цифр числа e (осно- вания натуральных логарифмов), а именно 2718, получаем для исходного сообщения ВОСТОЧНЫЙ ЭКСПРЕСС следующий шифротекст.

Сообщение

ВОСТОЧНЫЙ ЭКСПРЕСС

Ключ

27827182 71827182

Шифротекст

ДКТЬЮОГЛ ДЛЩСЧЖЩУ

На рис. 2.8 показан интерфейс программы, реализующей шифр Гронсфельда.

Рис. 2.8. Окно программы, реализующей шифр Гронсфельда

Код программы:

//Функция шифрования
function GronsfeldEncipher(toCode, K: string): string;
//toCode — исходный текст, K-ключ
var i, T, _T: integer;
begin
//определяется номер символа
for i:= 1 to length(toCode) do begin
  T:= Ord(toCode[i])+strtoint(K[(pred(i) mod length(K))+ 1]);
  if T >= 256 then dec(T, 255);
  toCode[i]:= Chr(T);
end;
GronsfeldEncipher:=toCode;

2.2.2. Система шифрования Виженера

Данный алгоритм можно описать таблицей шифрования, называемой квадратом Виженера, который используется для зашифрования и расшифрования. При шифровании исходного сообщения его выписываются в строку, а под ним записывают ключевое слово или фразу. Если ключ оказался короче сообщения, то его циклически повторяют.

Пример таблицы шифрования системы шифрования Виженера представлен в табл. 2.9.

Таблица 2.9

<table>
<thead>
<tr>
<th>Ключ</th>
<th>А</th>
<th>Б</th>
<th>В</th>
<th>Г</th>
<th>Д</th>
<th>...</th>
<th>Э</th>
<th>Ю</th>
<th>Я</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>А</td>
<td>Б</td>
<td>В</td>
<td>Г</td>
<td>Д</td>
<td>...</td>
<td>Э</td>
<td>Ю</td>
<td>Я</td>
</tr>
<tr>
<td>1</td>
<td>Б</td>
<td>В</td>
<td>Г</td>
<td>Д</td>
<td>Е</td>
<td>...</td>
<td>Ю</td>
<td>Я</td>
<td>А</td>
</tr>
<tr>
<td>2</td>
<td>В</td>
<td>Г</td>
<td>Д</td>
<td>Е</td>
<td>Ж</td>
<td>...</td>
<td>Я</td>
<td>А</td>
<td>Б</td>
</tr>
<tr>
<td>3</td>
<td>Г</td>
<td>Д</td>
<td>Е</td>
<td>Ж</td>
<td>З</td>
<td>...</td>
<td>А</td>
<td>Б</td>
<td>В</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>30</td>
<td>Ю</td>
<td>Я</td>
<td>А</td>
<td>Б</td>
<td>В</td>
<td>...</td>
<td>Б</td>
<td>Ь</td>
<td>Э</td>
</tr>
<tr>
<td>31</td>
<td>Я</td>
<td>А</td>
<td>Б</td>
<td>В</td>
<td>Г</td>
<td>...</td>
<td>ъ</td>
<td>є</td>
<td>Ю</td>
</tr>
</tbody>
</table>

В процессе шифрования в верхней строке таблицы находят очередной символ исходного текста, а в левом столбце — очередное значение ключа (номер очередного символа ключевого слова в алфавите, начиная с цифры 0). Символ шифротекста находится на пересечении столбца, определяемого шифруемым символом, и строки, определяемой числовым значением ключа.

На рис. 2.9 показан интерфейс программы, реализующей алгоритм системы шифрования Виженера.

Код программы:

{Функция шифрует текст по формуле:
Vig(m[i]) = m[i]+(i mod d) mod n,
где m-номер буквы,
Рис. 2.9. Окно программы, реализующей алгоритм системы шифрования Вижнера

$k$-номер буквы ключа,
$n$ — мощность алфавита

procedure TForm1.Btn1Click(Sender: TObject);
var
toCode, K: string; // tocode — исходный текст, k — ключ
i: integer;
currK: byte;
s: string;
begin
K := Edit1.text;
toCode := Edit2.text;
s := '';
// проходим все буквы текста
for i := 1 to length(toCode) do
begin
currK := pred(pos(K[pred(i) mod length(K)+1], TViginer));
s := s + TViginer[pred((pos(toCode[i], TViginer)+currK)) mod length(TViginer)+1];
end;
Edit3.text := s;
end;

2.2.3. Шифр Вижнера с автотюском

Дальнейшей модификацией системы Вижнера является система шифров с автотюском (auto-key). Шифрование начинается с помощью «первичного ключа» (который является настоящим ключом в нашем смысле) и продолжается с помощью сообщения или крипограммы, смешенной на длину первичного ключа, затем производится сложение по модулю, равному мощности алфавита.
Например:
Сообщение ПРИВЕТПРИМАТУ
Первичный ключ БГПУ
Автоключ ПРИВЕТПРИ
Шифротекст СУЧХФВЧТНЮПВЫ

Расшифрование не представляет труда: по первичному ключу получается начало сообщения, после чего найденная часть исходного сообщения используется в качестве ключа.

На рис. 2.10 показан интерфейс программы, реализующей алгоритм шифра Виженера с автоключом.

Рис. 2.10. Окно программы, реализующей алгоритм шифра Виженера с автоключом

Код программы:

ТViginer: string = 'абвгдеёжзйклмнопрстуфхцчшщъэюя';
//функция шифрования
procedure TForm1.Btn1Click(Sender: TObject);
  var toCode, K: string;
  //toCode – исходный текст, k-ключ
  i: integer;
  currK: byte;
  s: string;
  begin
    K:= Edit1.text;
    toCode:=Edit2.text;
    s:= '';
    for i:= 1 to length(toCode) do
      begin
        //проходятся все буквы текста
currK:=pred(pos(K[pred(i) mod length(K)+1], T Viginer));
mod length(T Viginer)+1);
end;
    Edit3.text:= s;
end;

2.2.4. Шифр Вижинера с перемешанным алфавитом

Такой шифр представляет собой простую подстановку с последую-
ющим применением шифра Вижинера.

На рис. 2.11 показан интерфейс программы, реализующей алго-
ритм шифра Вижинера с перемешанным алфавитом.

![Шифр Вижинера с перемешанным алфавитом](image)

Рис. 2.11. Окно программы, реализующей алгоритм шифра Вижинера с перемешанным алфавитом

Код программы:

```pascal
//Функция шифрования
procedure TForm1.Btn1Click(Sender: TObject);
    var toCode, K: string;
//toCode-текст, k-ключ
    i: integer;
    currK: byte;
    s: string;
begin
    K:= Edit1.text;
    toCode:=Edit2.text;
    s:= '';
//проходятся все буквы текста
    for i:= 1 to length(toCode) do
    begin
        currK:= pred(pos(K[pred(i) mod length(K)+1], T Viginer));
```

60
2.2.5. Двойной квадрат Уитстона

Процедура шифрования выполняется следующим образом. Перед шифрованием исходное сообщение разбивается на биграммы. Каждая биграмма шифруется отдельно. Первую букву биграммы находят в левой таблице, а вторую букву — в правой таблице. Затем мысленно строят прямоугольник так, чтобы буквы биграммы лежали в его противоположных вершинах. Другие две вершины этого прямоугольника дают буквы биграммы шифротекста.

Если обе буквы биграммы лежат в одной строке, то и буквы шифротекста берут из той же строки. Первую букву биграммы шифротекста берут из левой таблицы в столбце, соответствующем второй букве биграммы сообщения. Вторая же буква биграммы шифротекста берется из правой таблицы в столбце, соответствующем первой букве биграммы сообщения.

Пример: зашифровать сообщение:

ПР ИЛ ЕТ АЮ _Ш ЕС ТО ГО

Шифрующие таблицы биграммного шифра двойной квадрат Уитстона

<table>
<thead>
<tr>
<th>Ж</th>
<th>Ш</th>
<th>Н</th>
<th>Ю</th>
<th>Р</th>
</tr>
</thead>
<tbody>
<tr>
<td>И</td>
<td>Т</td>
<td>Ь</td>
<td>Ц</td>
<td>Б</td>
</tr>
<tr>
<td>Я</td>
<td>М</td>
<td>Е</td>
<td>.</td>
<td>С</td>
</tr>
<tr>
<td>В</td>
<td>Ы</td>
<td>П</td>
<td>Ч</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td>Д</td>
<td>У</td>
<td>О</td>
<td>К</td>
</tr>
<tr>
<td>З</td>
<td>Э</td>
<td>Ф</td>
<td>Г</td>
<td></td>
</tr>
<tr>
<td>Х</td>
<td>А ,</td>
<td>Л</td>
<td>Ь</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>И</th>
<th>Ч</th>
<th>Г</th>
<th>Я</th>
<th>Т</th>
</tr>
</thead>
<tbody>
<tr>
<td>,</td>
<td>Ж</td>
<td>Ь</td>
<td>М</td>
<td>О</td>
</tr>
<tr>
<td>З</td>
<td>Ю</td>
<td>Р</td>
<td>В</td>
<td>Ш</td>
</tr>
<tr>
<td>Ц</td>
<td>:</td>
<td>П</td>
<td>Е</td>
<td>Л</td>
</tr>
<tr>
<td>Ь</td>
<td>А</td>
<td>Н</td>
<td>.</td>
<td>Х</td>
</tr>
<tr>
<td>Э</td>
<td>К</td>
<td>С</td>
<td>Ш</td>
<td>Д</td>
</tr>
<tr>
<td>Б</td>
<td>Ф</td>
<td>У</td>
<td>Ы</td>
<td></td>
</tr>
</tbody>
</table>

С использованием таблиц шифротекст будет иметь следующий вид:

ПЕ ОВ ШН ФМ ЕШ РФ БЖ ДЦ

Контрольные вопросы

1. Какие шифры называют шифрами сложной замены?
2. Приведите примеры шифров сложной замены. Опишите алгоритм одного из них.
3. Поясните понятие «многоалфавитная подстановка».
4. В чем отличие шифров простой и сложной замены?
5. Опишите алгоритм шифра Гронсфельда.
6. Опишите алгоритм системы шифрования Виженера.
7. Опишите алгоритм шифра Виженера с автоключом.
8. Опишите алгоритм шифра Виженера с переменным алфавитом.
9. Опишите принцип использования биграмм в алгоритме двойного квадрата Уитстона.
10. Опишите алгоритм двойного квадрата Уитстона.

2.2.7. Тесты по теме «Алгоритмы сложной замены»

Таблица 2.10

<table>
<thead>
<tr>
<th>Ключ</th>
<th>А</th>
<th>Б</th>
<th>В</th>
<th>Г</th>
<th>Д</th>
<th>…</th>
<th>Э</th>
<th>Ю</th>
<th>Я</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>А</td>
<td>Б</td>
<td>В</td>
<td>Г</td>
<td>Д</td>
<td>…</td>
<td>Э</td>
<td>Ю</td>
<td>Я</td>
</tr>
<tr>
<td>1</td>
<td>Б</td>
<td>В</td>
<td>Г</td>
<td>Д</td>
<td>Е</td>
<td>…</td>
<td>Ю</td>
<td>Я</td>
<td>А</td>
</tr>
<tr>
<td>2</td>
<td>В</td>
<td>Г</td>
<td>Д</td>
<td>Е</td>
<td>Ж</td>
<td>…</td>
<td>Я</td>
<td>А</td>
<td>Б</td>
</tr>
<tr>
<td>3</td>
<td>Г</td>
<td>Д</td>
<td>Е</td>
<td>Ж</td>
<td>З</td>
<td>…</td>
<td>А</td>
<td>Б</td>
<td>В</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>30</td>
<td>Ю</td>
<td>Я</td>
<td>А</td>
<td>Б</td>
<td>В</td>
<td>…</td>
<td>Ы</td>
<td>Ъ</td>
<td>Э</td>
</tr>
<tr>
<td>31</td>
<td>Я</td>
<td>А</td>
<td>Б</td>
<td>В</td>
<td>Г</td>
<td>…</td>
<td>Ъ</td>
<td>Э</td>
<td>Ю</td>
</tr>
</tbody>
</table>

Таблица 2.11

<table>
<thead>
<tr>
<th>_</th>
<th>Ш</th>
<th>Н</th>
<th>Ю</th>
<th>Р</th>
<th>И</th>
<th>Ч</th>
<th>Г</th>
<th>Я</th>
<th>Т</th>
</tr>
</thead>
<tbody>
<tr>
<td>И</td>
<td>Т</td>
<td>Ь</td>
<td>Ц</td>
<td>Б</td>
<td>Ж</td>
<td>В</td>
<td>М</td>
<td>Ш</td>
<td></td>
</tr>
<tr>
<td>Т</td>
<td>Ь</td>
<td>Ц</td>
<td>Б</td>
<td>Ы</td>
<td>Э</td>
<td>В</td>
<td>О</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ы</td>
<td>Ч</td>
<td>П</td>
<td>Е</td>
<td>Л</td>
<td>Э</td>
<td>К</td>
<td>С</td>
<td>Ш</td>
<td></td>
</tr>
<tr>
<td>Я</td>
<td>А</td>
<td>Л</td>
<td>В</td>
<td>Б</td>
<td>Ф</td>
<td>У</td>
<td>Ы</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Вариант 1

1. Какой шифр сложной замены основан на смещении по алфавиту в соответствии с ключом шифра:
   a) шифр Гронсфельда;
   b) шифр Виженера;
   v) шифр Виженера с автоключом?

2. В каком алгоритме используются ключи, состоящие только из цифр:
а) шифр Вижищера;
б) шифр Гронсфельда;
в) двойной квадрат Уитстона?
3. Выберите вариант шифротекста исходного сообщения «мама мыла раму» зашифрованного с использованием шифра Гронсфельда с ключом 9851:
   а) хзсбхгрбщцсф;
   б) дшздужжазшт;
   в) аокгвлитерао.
4. Выберите вариант шифротекста исходного сообщения «информационная безопасность», зашифрованного с использованием шифра Вижищера с ключом тест:
   а) ѣтеавссийтофд тщщаттгтягдб;
   б) ѣтеавссийтофд тщщаттгтягдб;
   в) ѣтеавссийтофд тщщаттгтягдх.
5. Выберите вариант шифротекста исходного сообщения «информация», зашифрованного с использованием алгоритма двойного квадрата Уитстона:
   а) кьербяйб м;
   б) ыделябым и;
   в) ыдкяеббы м.

Вариант 2
1. Выберите вариант шифротекста исходного сообщения «мама мыла раму», зашифрованного с использованием шифра Гронсфельда с ключом 6574:
   а) теудтатдцоуч;
   б) жыъжцъцъён;
   в) ёротентиладк.
2. Выберите вариант шифротекста исходного сообщения «информационная безопасность», зашифрованного с использованием шифра Вижищера с ключом студент:
   а) щяэтхщэтзъбстнс тщщфнгюадцб;
   б) оаэхтэзъбстнс тщщфнгюадцб;
   в) щяэтхщэтзъбстнс тщщфнгюазцб.
3. Выберите вариант шифротекста исходного сообщения «компьютеры», зашифрованного с использованием алгоритма двойного квадрата Уитстона:
   а) хряжымеяь;
   б) яхырежымъя;
   в) хярыжемъяь.
4. В каком из алгоритмов используются ключи, состоящие только из букв:
   а) шифр Вижинера;
   б) двойной квадрат Уитстона;
   в) шифр Гронсфельда с автоключом?
5. В каком шифре используются биграммы:
   а) шифр Гронсфельда с автоключом;
   б) шифр Вижинера;
   в) двойной квадрат Уитстона?

Вариант 3
1. Выберите вариант шифротекста исходного сообщения «мама мыла раму», зашифрованного с использованием шифра Гронсфельда с ключом 1982:
   а) нифэндувисифх;
   б) льчэолтдюпчес;
   в) висоерапстее.
2. Выберите вариант шифротекста исходного сообщения «информационная безопасность», зашифрованного с использованием шифра Вижинера с ключом шифр:
   а) рриау щяннхвщ яиахег;
   б) рриау щяююхвщ яиахег;
   в) рриау щяюотвщ яиатег.
3. Выберите вариант шифротекста исходного сообщения «шифрование», зашифрованного с использованием алгоритма двойного квадрата Уитстона:
   а) цюсе..удмв;
   б) юцес..дувм;
   в) цюе..умдв.
4. В каких алгоритмах используются ключи, состоящие только из цифр:
   а) шифр Вижинера;
   б) шифр Гронсфельда;
   в) двойной квадрат Уитстона?
5. В каком из шифров ключ отсутствует:
   а) шифр Вижинера с перемешанным алфавитом;
   б) шифр Гронсфельда;
   в) двойной квадрат Уитстона?

Вариант 4
1. Выберите вид шифротекста при зашифровывании сообщения «мама мыла раму» с помощью шифра Гронсфельда с ключом 2431:
1. Выберите вариант шифротекста исходного сообщения «шифры сложной замены», зашифрованного с использованием шифра Вижинера с ключом замена:
   a) янахи стоттый оанкьы;
   b) янахи стоттый оанкьы;
   в) янахи стоттый оанкьы.
2. Выберите вариант шифротекста исходного сообщения «отличник», зашифрованного с использованием алгоритма двойного квадрата Уитстона:
   a) хюбю:пэж;
   б) хюбюп:жз;
   в) юхюб:пэж.
4. В каком из шифров используются ключи, состоящие только из букв:
   a) шифр Вижинера;
   б) двойной квадрат Уитстона;
   в) шифр Гронсфельда с автоключом?
5. В каком из шифров ключ отсутствует:
   a) шифр Вижинера с перемешанным алфавитом;
   б) двойной квадрат Уитстона;
   в) шифр Гронсфельда?

Вариант 5
1. Выберите вид шифротекста при зашифровывании сообщения «мама мыла раму» с помощью шифра Гронсфельда с ключом 1432:
   a) лъёолчиюёлъёс;
   б) ндпвняовсдпх;
   в) влаоьтелошъаоз.
2. Выберите вариант шифротекста исходного сообщения «шифры сложной замены», зашифрованного с использованием шифра Вижинера с ключом лифт:
   a) гривж щяасхы лиавшг;
   б) гривр щяасхы лиочшг;
   в) гривж щяасхы лиачшг.
3. Выберите вариант шифротекста исходного сообщения «клавиатура», зашифрованного с использованием алгоритма двойного квадрата Уитстона:
   a) хвыымжкъаяб;
6) вхымжкъвъцья;
в) хвымжкъвъцья.
4. Какой из шифров сложной замены основан на смещении по алфавиту в соответствии с ключом шифра:
а) шифр Гронсфельда;
б) шифр Вижинера;
в) шифр Вижинера с автоключом?
5. В каком из шифров используются биграммы:
а) шифр Вижинера;
б) двойной квадрат Уитстона;
в) шифр Гронсфельда с автоключом?

Таблица правильных ответов

<table>
<thead>
<tr>
<th>Вопрос</th>
<th>Вариант</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>а</td>
<td>а</td>
<td>а</td>
<td>а</td>
<td>а</td>
<td>б</td>
</tr>
<tr>
<td>2</td>
<td>б</td>
<td>а</td>
<td>б</td>
<td>в</td>
<td>в</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>а</td>
<td>в</td>
<td>а</td>
<td>б</td>
<td>в</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>б</td>
<td>в</td>
<td>б</td>
<td>в</td>
<td>а</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>б</td>
<td>в</td>
<td>в</td>
<td>б</td>
<td>а</td>
<td>б</td>
</tr>
</tbody>
</table>

2.2.8. Задание для самостоятельной работы по практической реализации алгоритмов шифрования методами замены

Цель практического занятия: изучение теоретических основ криптографической защиты информации методами шифрования на базе алгоритмов замены и приобретение навыков практической программной реализации данных алгоритмов.

Порядок выполнения задания.

Для нечетных вариантов предлагается реализовать процедуру шифрования файлов, для четных — дешифрования с использованием указанных методов.

Если ключ, используемый при шифровании, не указан, следует задать его самостоятельно.

1—2. Зашифровать исходное сообщение с использованием системы шифрования Цезаря.

3—4. Зашифровать исходное сообщение, используя аффинную систему подстановок Цезаря при $A = 12$, $B = 7$.

5—6. Зашифровать исходное сообщение с использованием полибианского квадрата. Заполнение таблицы размером 8х4 буквами
алфавита реализовать в следующем порядке: сначала нечетные столбцы, затем — четные.

7—8. Зашифровать исходное сообщение с использованием лозунгового шифра. В качестве ключа использовать свое имя или фамилию.

9—10. Зашифровать исходное сообщение с использованием полибианского квадрата. Заполнение таблицы размером 8×4 буквами алфавита реализовать в следующем порядке: по вертикали, начиная с левого верхнего угла сверху вниз и снизу вверх.

11—12. Зашифровать исходное сообщение с использованием шифра Гронсфельда. В качестве ключа использовать группу из пяти цифр.

13—14. Зашифровать исходное сообщение используя аффинную систему подстановок Цезаря при $A = 13$, $B = 5$.

15—16. Зашифровать исходное сообщение с использованием шифрующей таблицы Трисемуса.

17—18. Зашифровать исходное сообщение с использованием биграммного шифра Плейфера.

19—20. Зашифровать исходное сообщение с использованием системы омофонов.

21—22. Зашифровать исходное сообщение с использованием системы шифрования двойного квадрата Уитстона.

23—24. Зашифровать исходное сообщение с использованием системы шифрования Вижинера.

25—26. Зашифровать исходное сообщение с использованием системы шифрования Вижинера с автоключом.
Тема 3

ШИФРЫ ПЕРЕСТАНОВКИ

Шифр перестановки — это шифр преобразования, с помощью которого изменяют только порядок следования символов исходного текста, не изменения их самих. Простейшим шифром перестановки является подстановка степени n. И шифрование, и дешифрование осуществляются с помощью шифрующей таблицы.

3.1. Шифр простейшей перестановки

Шифрование проводится следующим образом: первый символ открытого текста заменяется i₁-м символом из этого же текста; второй — i₂-м и т.д. Дешифрование производится аналогично с помощью этой же таблицы.

Рассмотрим реализацию шифра простейшей перестановки на примере зашифровывания слова ПЕРЕСТАНОВКА.

Вариант шифрующей таблицы шифра простейшей перестановки представлен в табл. 3.1.

<table>
<thead>
<tr>
<th>Пример шифра простейшей перестановки</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>i₁</td>
</tr>
</tbody>
</table>

На рис. 3.1 показан интерфейс программы, реализующей шифр простейшей перестановки.

Рис. 3.1. Окно программы, реализующей шифр простейшей перестановки
Код программы:

```pascal
procedure TForm1.Button1Click(Sender: TObject);
// Шифрование
var
tocode: string;
begin
  // tocode — исходный текст, key-ключ
  tocode:= edit1.text;
  key:=edit4.Text;
a:=0;
len:=length(edit4.Text);
i:=1;
repeat
  b:=a+strtoint(key[(pred(i) mod len+1)]);
  // нахождение буквы, соответствующей цифре ключа
  edit2.Text:=edit2.Text+tocode[b];
  if i mod len=0 then a:=a+len;
  inc(i);
  until i>length(edit1.Text);
end;
```

3.2. Шифр маршрутной перестановки

Согласно данному алгоритму шифруемое исходное сообщение записывается в определенном порядке, определяемом некоторой геометрической фигурой обхода, называемой маршрутом. При этом смысл шифрования в том, что исходное сообщение записывается по направлению одного маршрута, а зашифрованное считывается по ходу другого маршрута.

Рассмотрим пример реализации шифра маршрутной перестановки. Для этого будем вписывать исходное сообщение в прямоугольную таблицу по горизонтали в строку, начиная с левого верхнего угла. Запись выполняется поочередно вначале слева направо, в следующей строке — справа налево.

Зашифрованное сообщение формируется выписыванием символов по вертикали, по столбцам, начиная с верхнего правого угла и двигаясь поочередно сверху вниз и снизу вверх.

Рассмотрим реализацию шифра маршрутной перестановки на примере зашифровывания фразы

**ПРИМЕРМАРШРУТНОЙПЕРЕСТАНОВКИ.**

Используя в качестве шаблона шифрующей таблицы прямоугольник размером 4×7, получаем зашифрованное сообщение
МАСТАЕРЕШРОЕРМИУПВКЙТРПНОЙ

Вариант шифрующей таблицы шифра маршрутной перестановки представлен в табл. 3.2

Таблица 3.2

<table>
<thead>
<tr>
<th>П</th>
<th>Р</th>
<th>И</th>
<th>М</th>
<th>Е</th>
<th>Р</th>
<th>М</th>
</tr>
</thead>
<tbody>
<tr>
<td>Н</td>
<td>Т</td>
<td>У</td>
<td>Р</td>
<td>Ш</td>
<td>Р</td>
<td>А</td>
</tr>
<tr>
<td>О</td>
<td>Й</td>
<td>П</td>
<td>Е</td>
<td>Р</td>
<td>Е</td>
<td>С</td>
</tr>
<tr>
<td>И</td>
<td>К</td>
<td>В</td>
<td>О</td>
<td>Н</td>
<td>А</td>
<td>Т</td>
</tr>
</tbody>
</table>

На рис. 3.2 показан интерфейс программы, реализующей алгоритм маршрутной перестановки.

Рис. 3.2. Окно программы, реализующей алгоритм маршрутной перестановки

Код программы:

```pascal
procedure TForm1.Button1Click(Sender: TObject);
var
  i, j, k, p, u, t, s, ss: integer;
  tocode: string;
begin
  tocode:=edit1.Text;
  j:=1;  // переход по 1-й строке
  k:=8;  // переход по 2-й строке
  p:=15; // переход по 3-й строке
  u:=22; // переход по 4-й строке
  ss:=3; // переход по строкам
  if length(tocode)>28 then
    // 28-макс. количество букв текста
    begin
      showmessage('Длина сообщения не более 28 символов');
      exit;
    end;
  for i:=length(tocode) to 28 do
    tocode:=tocode+'*';
```

70
{исходное сообщение вписывается в прямоугольную таблицу по горизонтали, начиная с левого верхнего угла, поочередно слева направо и справа налево}

```
for i:=0 to table1.ColCount-1 do
   begin
      table1.Cells[i, 0] := tocode[j];
      j:=j+1;
      k:=k+1;
      table1.Cells[i, 2] := tocode[p];
      p:=p+1;
      u:=u+1;
   end;
   s1:='';
   s2:='';
   s3:='';
   s4:='';
   s5:='';
   s6:='';
   s7:='';
```

{сообщение выписывается по вертикали, начиная с верхнего правого угла и двигаясь поочередно сверху вниз и снизу вверх}

```
for s:=0 to 3 do
   begin
      s1:=s1+table1.Cells[6, s];
      s3:=s3+table1.Cells[4, s];
      s5:=s5+table1.Cells[2, s];
      s7:=s7+table1.Cells[0, s];
   end;
   while ss>=0 do
      begin
         s2:=s2+table1.Cells[5, ss];
      end;
      //копируется текст шестого столбца снизу вверх
      s4:=s4+table1.Cells[3, ss];
      s6:=s6+table1.Cells[1, ss];
      ss:=ss-1; //переход на строку выше
   end;
   Edit2.Text:=s1+s2+s3+s4+s5+s6+s7;
end;
```
3.3. Шифр перестановки «Сцитала»

Одними из первых применять этот алгоритм начали древние спартанцы еще в 475 г. до н.э. Для целей зашифровывания сообщения ими применялось специальное приспособление, называемое сциталой. Приспособление сцитала представляло собой полоску ткани, кожи или подобной материи, наматываемой на цилиндр определенного диаметра, причем этот диаметр и являлся секретным ключом. На намотанную на цилиндр полоску по вертикали, вдоль цилиндра, наносилось исходное сообщение. Далее полоска размывалась и доставлялась получателю сообщения, который для расшифровки имел цилиндр такого же диаметра. Без цилиндра текст прочитать невозможно, так как символы будут неупорядочены. На практике метод весьма прост, так как для успешного вскрытия алгоритма всегда можно иметь набор цилиндров различных диаметров и продолжать попытки криптоанализа до тех пор, пока не будет получен корректный текст.

Современная версия алгоритма «Сцитала» состоит в использовании двумерных массивов в качестве шифрующей таблицы, куда в одном направлении помещается исходное сообщение, а в другом направлении происходит считывание шифротекста.

Рассмотрим реализацию шифра перестановки «Сцитала» на примере зашифровывания фразы

**MEET ME AT SUNSET**

В процессе шифрования выбирается двумерный массив определенного размера и инициализируется нулями. Исходное сообщение записывается по строкам данного массива слева направо.

Вариант шифрующей таблицы как аналог двумерного массива для реализации шифра перестановки «Сцитала» представлен в табл. 3.3.

| Пример шифрующей таблицы шифра перестановки «Сцитала» |
|-------------------|---|---|---|---|
| M | e | e | t |   |
| M | e |   | a | t |
|   | s | u | n | s |
| e | t | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

При формировании зашифрованного сообщения следует читать текст по столбцам слева направо и сверху вниз.
Зашифрованное сообщение будет иметь вид

```
MM_EOESTOE_U00TAN00_TS00
```

Для того чтобы декодировать сообщение, необходимо заполнить массив по столбцам, после чего считать текст по строкам слева направо.

На рис. 3.3 показан интерфейс программы, реализующей алгоритм Сцитала.

![Интерфейс программы Сцитала](image)

**Рис. 3.3. Окно программы, реализующей алгоритм «Сцитала»**

Код программы:

```pascal
procedure TForm1.BitBtn1Click(Sender: TObject);
  var
  tocode: string;
  begin
    tocode:=edit1.Text;
    k:=1;
    for i:=length(tocode) to 25 do
      tocode:=tocode+'0';
    // текст добавляется в таблицу
    for i:=0 to 4 do
      begin
        for j:=0 to 4 do
          begin
            table1.Cells[j, i]:=tocode[k];
            k:=k+1;
          end;
      end;
```

```
3.4. Шифр «Поворотная решетка»

Проиллюстрируем построение шифра следующей последовательностью действий. На начальном этапе реализации шифра изготавливается трафарет из прямоугольного листа клетчатой бумаги размером $2m \times 2k$ клеток. В трафарете вырезается $mk$ клеток так, чтобы при наложении его на чистый лист бумаги того же размера четырьмя возможными способами его вырезы полностью покрывали площадь листа. Символы сообщения последовательно вписываются в вырезы трафарета (по строкам, в каждой строке слева направо) при каждом из четырех его возможных положений в заранее установленном порядке.

Поясним процесс шифрования на примере. В качестве ключа используется решетка $6 \times 10$, приведенная на рис. 3.4.

Рис. 3.4. Пример решетки-ключа

Зашифруем с ее помощью текст

ШИФРРЕШЕТКАЯВЛЯЕТСЯ ЧАСТНЫМ СЛУЧАЕМ
ШИФРАМАРШРУТНОЙ ПЕРЕСТАНОВКИ

Наложив решетку на лист бумаги, вписываются первые 15 (по числу вырезов) символов сообщения: ШИФРРЕШЕТКАЯВЛЯ.
Сняв решетку, получим текст, представленный на рис. 3.5. На следующем этапе следует повернуть решетку на 180°. В вырезанных окошках появятся новые, еще не заполненные клетки. В них вписываются следующие 15 символов. В результате получается фрагмент текста, приведенный на рис. 3.6. На очередном этапе шифрования решетка переворачивается на другую сторону и остаток текста зашифровывается аналогичным образом, как показано на рис. 3.7, 3.8.

<table>
<thead>
<tr>
<th>И</th>
<th>Ф</th>
<th>Р</th>
<th>Р</th>
<th>Е</th>
</tr>
</thead>
<tbody>
<tr>
<td>Е</td>
<td>Ш</td>
<td>К</td>
<td></td>
<td></td>
</tr>
<tr>
<td>А</td>
<td>В</td>
<td>Л</td>
<td>Я</td>
<td></td>
</tr>
</tbody>
</table>

Рис. 3.5. Первый этап заполнения решетки

<table>
<thead>
<tr>
<th>Е</th>
<th>И</th>
<th>А</th>
<th>Т</th>
<th>С</th>
<th>Я</th>
</tr>
</thead>
<tbody>
<tr>
<td>Р</td>
<td>Р</td>
<td>Ч</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Е</td>
<td>Ш</td>
<td>С</td>
<td>К</td>
<td>Ы</td>
<td></td>
</tr>
<tr>
<td>М</td>
<td>С</td>
<td>Л</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Я</td>
<td>В</td>
<td>Л</td>
<td>Ч</td>
<td>Я</td>
<td></td>
</tr>
</tbody>
</table>

Рис. 3.6. Второй этап заполнения решетки

<table>
<thead>
<tr>
<th>Е</th>
<th>И</th>
<th>А</th>
<th>Т</th>
<th>С</th>
<th>И</th>
<th>М</th>
<th>Я</th>
<th>Ш</th>
</tr>
</thead>
<tbody>
<tr>
<td>Р</td>
<td>Р</td>
<td>Ч</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Е</td>
<td>А</td>
<td>Ф</td>
<td>Ш</td>
<td>С</td>
<td>Р</td>
<td>Е</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Т</td>
<td>Н</td>
<td>М</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Р</td>
<td>А</td>
<td>М</td>
<td>С</td>
<td>Ш</td>
<td>Л</td>
<td>Р</td>
<td>У</td>
<td></td>
</tr>
<tr>
<td>Т</td>
<td>Я</td>
<td></td>
<td>В</td>
<td>Л</td>
<td>Ч</td>
<td>Я</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Рис. 3.7. Третий этап заполнения решетки

<table>
<thead>
<tr>
<th>Е</th>
<th>И</th>
<th>О</th>
<th>Й</th>
<th>Ф</th>
<th>П</th>
<th>Р</th>
<th>Ч</th>
<th>Е</th>
</tr>
</thead>
<tbody>
<tr>
<td>Р</td>
<td>А</td>
<td>Ф</td>
<td>Е</td>
<td>Ш</td>
<td>С</td>
<td>Р</td>
<td>С</td>
<td>Е</td>
</tr>
<tr>
<td>Т А</td>
<td>Т</td>
<td>Н</td>
<td>М</td>
<td>А</td>
<td>К</td>
<td>Ы</td>
<td>А</td>
<td></td>
</tr>
<tr>
<td>Р</td>
<td>А</td>
<td>М</td>
<td>С</td>
<td>Ш</td>
<td>Л</td>
<td>Р</td>
<td>У</td>
<td>Н</td>
</tr>
<tr>
<td>О</td>
<td>Т</td>
<td>Я</td>
<td>В</td>
<td>К</td>
<td>Б</td>
<td>Л</td>
<td>И</td>
<td>Ч</td>
</tr>
</tbody>
</table>

Рис. 3.8. Четвертый этап заполнения решетки
Получатель сообщения, имеющий точно такую же решетку, без труда прочтет исходный текст, наложив решетку на шифротекст по порядку четырьмя способами.

Можно доказать, что число возможных трафаретов, то есть количество ключей шифра «Поворотная решетка», составляет \( T = 4mk \). Этот шифр предназначен для сообщений длины \( n = 4mk \). Число всех перестановок в тексте такой длины составит \((4mk)!\), что во много раз больше числа \( T \). Однако уже при размере трафарета 8х8 число возможных решеток превосходит 4 миллиарда.

На рис. 3.9 показан интерфейс программы, реализующей алгоритм «Поворотная решетка».

![Изображение интерфейса программы](image.png)

Рис. 3.9. Окно программы, реализующей алгоритм «Поворотная решетка»

Код программы:

```pascal
type
  sType = string[n];
  matrix = array[1 .. n] of sType;
const
  mask: matrix = (  
    'x..x',
    '..x..',
    '..x..',
    '..x..',
    'x..x',
    'x..x',
    'x..x',
    'x..x'  
  );

// Процедура поворота матрицы
```

76
procedure T(var res: matrix);
    var
        i, j: integer;
        mx: matrix;
    begin
        mx := res;
        for i := 1 to n do
            for j := 1 to n do
                res[j, n - i + 1] := mx[i, j];
    end;

procedure EncodeText(const s: string;
    const mask: matrix; var mx: matrix);
    var
        i, j, count: integer;
        masked: matrix;
    begin
        // Заполнение матрицы mx строками по N пробелов
        for i := 1 to n do
            for j := 1 to n do
                mx[i] := mx[i] + #32;
        masked := mask;
        count := 1;
        while count <= length(s) do begin
            for i := 1 to n do
                for j := 1 to n do
                    if masked[i, j] = 'x' then begin
                        mx[i][j] := s[count];
                        inc(count);
                    end;
            T(masked);
        end;

procedure TForm1.Button1Click(Sender: TObject);
    begin
        EncodeText(edit1.text, mask, encoded);
        for i := 1 to n do
            for j := 1 to n do
                grid2.Cells[j-1, i-1] := encoded[i, j];
        for i := 1 to n do
            edit2.text := edit2.text + encoded[i];
    end;
3.5. Шифр вертикальной перестановки

Профилактикуем построение шифра следующей последовательностью действий. В качестве основы шифрующей таблицы используем прямоугольник, куда исходное сообщение вписывается по строкам слева направо. Для получения зашифрованного сообщения символы выписываются по вертикали. При этом порядок следования столбцов не подряд, один за другим, а определяется некоторым ключом.

Пусть, например, этот ключ таков: (5, 1, 4, 7, 2, 6, 3) — и с его помощью надо зашифровать сообщение

**ВОТПРИМЕРШИФРОВАТЬВЕРТИКАЛЬНОЙПЕРЕСТАНОВКИ**

Выбирая столбцы в порядке, заданном ключом, и выписыва последовательно буквы каждого из них сверху вниз, получаем

**ОРЕБЭКРФЙА-МААЕО-ТШРНСИВЕВЛРВИРКПН-ПИТОТ-**.

**Таблица шифрования шифром вертикальной перестановки**

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>В</td>
<td>О</td>
<td>Т</td>
<td>П</td>
<td>Р</td>
<td>И</td>
<td>М</td>
</tr>
<tr>
<td>Е</td>
<td>Р</td>
<td>Ш</td>
<td>И</td>
<td>Ф</td>
<td>Р</td>
<td>А</td>
</tr>
<tr>
<td>Б</td>
<td>Е</td>
<td>Р</td>
<td>Т</td>
<td>И</td>
<td>К</td>
<td>А</td>
</tr>
<tr>
<td>Л</td>
<td>Н</td>
<td>О</td>
<td>Й</td>
<td>П</td>
<td>Е</td>
<td></td>
</tr>
<tr>
<td>Р</td>
<td>Е</td>
<td>С</td>
<td>Т</td>
<td>А</td>
<td>Н</td>
<td>О</td>
</tr>
<tr>
<td>В</td>
<td>К</td>
<td>И</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ключ шифра вертикальной перестановки можно извлекать из какого-либо легко запоминающегося слова или предложения. Например, можно приписывать буквам числа в соответствии с обычным алфавитным порядком букв. Допустим, в качестве ключевого выбрано слово ПЕРЕСТАНОВКА. Присутствующая в нем буква А получает номер 1. Если какая-либо буква входит в ключевое слово несколько раз, то ее появления нумеруются последовательно слева направо. Поэтому второе вхождение буквы А получает номер 2. Поскольку буквы Б в этом слове нет, то буква В получает номер 3 и т.д. Процесс продолжается до тех пор, пока всем буквам не будут присвоены номера. Таким образом получается следующий ключ.

**Таблица шифрования с использованием ключа ПЕРЕСТАНОВКА**

<table>
<thead>
<tr>
<th>П</th>
<th>Е</th>
<th>Р</th>
<th>Е</th>
<th>С</th>
<th>Т</th>
<th>А</th>
<th>Н</th>
<th>О</th>
<th>В</th>
<th>К</th>
<th>А</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>
Для обеспечения дополнительной криптостойкости алгоритма можно повторно зашифровать сообщение, которое уже прошло шифрование. Такой метод шифрования называется двойной перестановкой. В этом случае перестановки определяются отдельно для столбцов и отдельно для строк. Сначала в таблицу записывается текст сообщения, затем поочередно переставляются столбцы, затем строки. При расшифровании порядок перестановок должен быть обратным.

Следует отметить, что даже двойная перестановка не отличается высокой стойкостью и сравнительно просто взламывается при любом размере таблицы шифрования.

На рис. 3.10 показан интерфейс программы, реализующей алгоритм шифра вертикальной перестановки.

**Рис. 3.10. Окно программы, реализующей алгоритм шифра вертикальной перестановки**

Код программы:

```pascal
procedure TForm1.BitBtn1Click(Sender: TObject);
    var   n, b, kl, t: integer;
         tocode, s: string;
         c: array[1..7] of integer;
    begin
         tocode:=edit1.Text;
         for i:=length(tocode) to 42 do
             tocode:=tocode+’*’;
         // ключ вписывается в таблицу
         for i:=0 to 6 do
```
begin
    table1.Cells[i, 0]:=edit3.Text[i+1];
    end;
k:=1; n:=1;

// текст записывается в таблицу
for i:=1 to 6 do
    begin
        for j:=0 to 6 do
            begin
                table1.Cells[j, i]:=tocode[k];
                k:=k+1;
            end;
    end;
for k:=1 to 7 do
    begin
        s:=inttostr(k);    //номер столбца
        for i:=0 to 6 do    //проход по всем столбцам
            if table1.Cells[i, 0]=s then
                for j:=1 to 6 do
        //текст столбца копируется
        // в зашифрованную строку
    end;
end;

3.6. Шифр на основе магических квадратов

Магическими квадратами называют квадратные таблицы со вписаными в их клетки последовательными натуральными числами, начиная с единицы, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число.

Шифруемый текст вписывается в магический квадрат в соответствии с нумерацией его клеток. Если затем выписать содержимое такой таблицы по строкам, то получится шифротекст, сформированный благодаря перестановке букв исходного сообщения.

Пример таблицы магического квадрата и его заполнение имеют вид

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>3</td>
<td>2</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>11</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>7</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>14</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>О</td>
<td>И</td>
<td>Р</td>
<td>М</td>
<td></td>
</tr>
<tr>
<td>Е</td>
<td>О</td>
<td>С</td>
<td>Ю</td>
<td></td>
</tr>
<tr>
<td>В</td>
<td>Т</td>
<td>А</td>
<td>Б</td>
<td></td>
</tr>
<tr>
<td>Л</td>
<td>Г</td>
<td>О</td>
<td>П</td>
<td></td>
</tr>
</tbody>
</table>

80
На рис. 3.11 показан интерфейс программы, реализующей шифр на основе магических квадратов.

Рис. 3.11. Окно программы, реализующей шифр на основе магических квадратов

Код программы:

```pascal
procedure TForm1.BitBtn1Click(Sender: TObject);
var
tocode: string;
begin
tocode:=edit1.Text;
// в магический квадрат буквы записываются согласно их порядковому номеру в открытом тексте
for a:=1 to length(tocode) do begin
  for b:=0 to 3 do begin
    for c:=0 to 3 do
      if table1.Cells[b, c]=inttostr(a) then
table2.Cells[b, c]:=edit1.Text[a];
  end;
end;
// считывается текст магического квадрата по строкам
for a:=0 to 3 do // проход по всем строкам begin
  for b:=0 to 3 do // проход по всем столбцам
    edit2.Text:=edit2.Text+table2.Cells[b, a];
end;
end;
```

3.7. Контрольные вопросы

1. Видом какого простейшего шифра является подстановка степени n?
2. В чем заключается метод шифрования перестановкой?
3. Что такое шифр маршрутной перестановки?
4. Какой маршрут можно использовать для реализации шифра Сцитала?
5. Что называется поворотной решеткой?
6. В чем суть шифра вертикальной перестановки?
7. Какой метод шифрования называется двойной перестановкой?
8. Оцените количество ключей шифра вертикальной перестановки. Во сколько раз это количество ключей возрастает при использовании двойной перестановки?
9. Что такое шифрующие таблицы магического квадрата и как они заполняются?
10. Приведите пример использования магического квадрата для шифрования сообщения ЯУЕЗАЮВНОВГОРОД.

3.8. Тесты по теме
«Шифры перестановки»

Вариант 1

1. В каком из алгоритмов метод преобразования состоит в том, что отрезок открытого текста записывается в фигуру по некоторой одной траектории, а выписывается по другой траектории:
   а) шифр Цезаря;
   б) битграммный шифр Плейфера;
   в) шифр маршрутной перестановки?

2. Формула, которая используется для вычисления второго параметра — количества столбцов ХХ — при известном первом — количестве строк ХХ и длине сообщения ХХ — в шифре перестановки «Сцитала»:
   а) \( n = \left[ \frac{m-1}{n} \right] + 1 \), где \([\cdot]\) — целая часть числа;
   б) \( n = [(k - 1) \cdot m] - 1 \), где \([\cdot]\) — целая часть числа;
   в) \( n = \left[ \frac{k-1}{m} \right] + 1 \), где \([\cdot]\) — целая часть числа.

3. Какой шифр считают первым транспозиционным (геометрическим) шифром:
   а) шифр маршрутной перестановки;
   б) шифр вертикальной перестановки;
   в) шифр поворотной решетки?

4. Каким образом происходит шифрование методом магического квадрата:
   а) буквы открытого текста необходимо вписать в магический квадрат в соответствии с нумерацией его клеток;
   б) буквы открытого текста необходимо вписать по горизонтали в магический квадрат, строки при этом берутся в порядке, определяемом ключом;
в) буквы открытого текста необходимо вписать по вертикали в магический квадрат, столбцы при этом берутся в порядке, определяемом ключом?

5. Ключом к шифру двойной перестановки служит:
а) последовательность номеров строк исходной таблицы, размеры таблицы, маршруты вписывания и выписывания;
б) последовательность номеров столбцов и номеров строк исходной таблицы, размеры таблицы, маршруты вписывания и выписывания;
в) последовательность номеров столбцов исходной таблицы, размеры таблицы, маршруты вписывания и выписывания.

Вариант 2
1. Шифр перестановки — это:
а) шифр преобразования, с помощью которого изменяют и порядок следования символов исходного текста, и сами символы;
б) шифр преобразования, с помощью которого изменяют только порядок следования символов исходного текста, не изменяя их самих;
в) шифр преобразования, с помощью которого не изменяют порядок следования символов исходного текста, а изменяют сами символы.

2. Какой шифр носит второе название «шифр Древней Спарты»:
а) шифр перестановки «Сцитала»;
б) шифр «Поворотная решетка»;
в) шифр маршрутной перестановки?

3. Для использования шифра, называемого «Поворотная решетка», изготавливается трафарет из прямоугольного листа клетчатой бумаги размера X2 клеток. В трафарете вырезано XX клеток, число возможных трафаретов, т.е. количество ключей шифра «Поворотная решетка», составляет:
а) \( T = 4 \cdot m \cdot k \);
б) \( T = 4^{(m \cdot k)} \);
в) \( T = \frac{4}{m \cdot k} \).

4. Число ключей в шифре вертикальной перестановки не более:
а) \( k! \), где \( k \) — число строк таблицы;
б) \( m \cdot k \), где \( k \) — число строк таблицы; \( m \) — число столбцов таблицы;
в) \( m! \), где \( m \) — число столбцов таблицы.

5. Каким образом происходит расшифровка в методе вертикальной перестановки:
а) шифрограмма выписывается из таблицы любой формы по вертикалям;
б) шифрограмма выписывается из прямоугольной таблицы по вертикалям, при этом столбцы выбираются в порядке, определяемом ключом;

в) шифрограмма выписывается из таблицы любой формы по вертикалям, при этом столбцы выбираются в порядке, определяемом ключом?

Вариант 3

1. Формула, которая используется для вычисления второго параметра — количества столбцов XX — при известном первом — количестве строк XX и длине сообщения XX — в шифре перестановки «Сцитала»:
   
a) \( n = \left\lfloor \frac{m-1}{m} \right\rfloor + 1 \), где \([\cdot]\) — целая часть числа;

   b) \( n = \lfloor (k-1) \cdot m \rfloor - 1 \), где \([\cdot]\) — целая часть числа;

   в) \( n = \left\lfloor \frac{k-1}{m} \right\rfloor + 1 \), где \([\cdot]\) — целая часть числа.

2. Сколькими способами наложения на таблицу трафарета с вырезанными ячейками происходит шифрование/декифрование в шифре «Поворотная решетка»:
   
a) 3;

   б) 4;

   в) 5;

   г) 6?

3. Квадратная таблица \( n \cdot n \), заполненная \( n^2 \) числами таким образом, что сумма чисел в каждой строке, каждом столбце и на обеих диагоналях одинакова, называется:
   
a) магический квадрат;

   б) полумагический квадрат;

   в) нормальный квадрат.

4. Каким образом происходит шифрование методом магического квадрата?
   
a) буквы открытого текста необходимо вписать в магический квадрат в соответствии с нумерацией его клеток;

   б) буквы открытого текста необходимо вписать по горизонтали в магический квадрат, строки при этом берутся в порядке, определяемом ключом;

   в) буквы открытого текста необходимо вписать по вертикали в магический квадрат, столбцы при этом берутся в порядке, определяемом ключом.

5. Ключом к шифру двойной перестановки служит:
а) последовательность номеров строк исходной таблицы, размеры таблицы, маршруты вписывания и выписывания;
б) последовательность номеров столбцов и номеров строк исходной таблицы, размеры таблицы, маршруты вписывания и выписывания;
в) последовательность номеров столбцов исходной таблицы, размеры таблицы, маршруты вписывания и выписывания.

Вариант 4
1. Число различных преобразований шифра перестановки, предназначенно для шифрования сообщений, где \( n \) — длина сообщения, \( m \) — количество столбцов таблицы, меньше либо равно:
а) \( P_n = n!; \)
б) \( A_n^m = \frac{n!}{(n-m)!}; \)
в) \( C_n^m = \frac{(n-m)!}{m!}. \)

2. Какой шифр представляет собой полоску материи, наматывающую на цилиндр с фиксированными длиной и диаметром, причем текст пишется поперек — слева направо:
а) шифр Цезаря;
б) диск Энея;
в) шифр перестановки «Сцифала»?

3. Ключом для дешифрования шифра поворотной решетки являются:
а) трафарет, порядок поворотов, длина (количество знаков) сообщения;
б) трафарет, маршрут вписывания и порядок поворотов;
в) трафарет, маршрут вписывания и порядок поворотов, длина (количество знаков) сообщения.

4. Число ключей в шифре вертикальной перестановки не более:
а) \( m! \), где \( m \) — число столбцов таблицы;
б) \( k! \), где \( k \) — число строк таблицы;
в) \( m \cdot k \), где \( k \) — число строк таблицы; \( m \) — число столбцов таблицы.

5. Каким образом происходит расшифровка в методе вертикальной перестановки:
а) шифрограмма выписывается из таблицы любой формы по вертикалям;
б) шифрограмма выписывается из прямоугольной таблицы по вертикалям, при этом столбцы выбираются в порядке, определяемом ключом;
в) шифрограмма выписывается из таблицы любой формы по вертикалям, при этом столбцы выбираются в порядке, определяемом ключом?

Вариант 5
1. Шифр перестановки — это:
   а) шифр преобразования, с помощью которого изменяют только порядок следования символов исходного текста, не изменяя их самих;
   б) шифр преобразования, с помощью которого изменяют и порядок следования символов исходного текста, а сами символы;
   в) шифр преобразования, с помощью которого не изменяют порядок следования символов исходного текста, а изменяют сами символы.

2. Для какого шифра применяется следующая формула:

   \[ n = \left[ \frac{k - 1}{m} \right] + 1, \]

где \( n \) — количество столбцов; \( \left[ \right] \) — целая часть числа; \( k \) — длина сообщения; \( m \) — количество строк:
   а) шифр маршрутной перестановки;
   б) шифр перестановки «Сизале»;
   в) шифр вертикальной перестановки?

3. Для шифрования и дешифрования с помощью шифра поворотной решетки изготавливается трафарет с вырезанными ячейками, при наложении которого на таблицу того же размера четырьмя возможными способами его вырезы должны покрывать:
   а) полностью все клетки таблицы ровно по одному разу;
   б) часть клеток таблицы, а остальные заполняются любым другим более или менее осмысленным текстом (или «небрежными» буквами);
   в) часть клеток таблицы, а остальные остаются пустыми.

4. Равная сумма в каждой строке, столбцу и диагонали магического квадрата вычисляется по формуле:

   а) \( n \frac{n^2 - n}{2} \);
   б) \( n \frac{n^2 + 1}{2} \);
   в) \( n \frac{n^2 + n}{2} \).

5. Ключом к шифру двойной перестановки служит:
   а) последовательность номеров строк исходной таблицы, размеры таблицы, маршруты вписывания и выписывания;
б) последовательность номеров столбцов исходной таблицы, размеры таблицы, маршруты вписывания и выписывания;
в) последовательность номеров столбцов и номеров строк исходной таблицы, размеры таблицы, маршруты вписывания и выписывания.

**Таблица правильных ответов**

<table>
<thead>
<tr>
<th>Вопрос</th>
<th>Вариант</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>в</td>
<td>б</td>
<td>в</td>
<td>а</td>
<td>а</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>в</td>
<td>а</td>
<td>б</td>
<td>в</td>
<td>б</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>в</td>
<td>а</td>
<td>а</td>
<td>б</td>
<td>а</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>а</td>
<td>в</td>
<td>а</td>
<td>а</td>
<td>б</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>б</td>
<td>б</td>
<td>б</td>
<td>а</td>
<td>в</td>
</tr>
</tbody>
</table>

3.9. Задание для самостоятельной работы по практической реализации алгоритмов шифрования методами перестановки

**Цель практического занятия:** изучение теоретических основ криптографической защиты информации методами шифрования на базе алгоритмов перестановки и приобретение навыков практической программной реализации данных алгоритмов.

**Порядок выполнения задания.**

Для нечетных вариантов предлагается реализовать процедуру шифрования файлов, для четных — дешифрования с использованием указанных методов.

Если ключ, используемый при шифровании, не указан, следует задать его самостоятельно.

**Варианты заданий**

1—2. Исходную последовательность разбейте на группы по четыре символа. В каждой группе символы переставьте с использованием подстановки, выбираемой самостоятельно.

3—4. Исходную последовательность разбейте на группы по четыре символа. Реализуйте двойную перестановку каждой последовательности символов.

5—6. Исходную последовательность разбейте на группы по восемь символов. Реализуйте шифрование методом перестановки по заданному ключу, при этом четные группы символов шифровать в исходном направлении, нечетные — в обратном.
7—8. Исходную последовательность разбейте на группы по восемь символов. В каждой группе символы переставьте с использованием подстановки, выбираемой самостоятельно.

9—10. Исходную последовательность разбейте на группы по восемь символов. Реализуйте двойную перестановку каждой группы символов, начиная с последнего.

11—12. Исходную последовательность разбейте на группы по восемь символов. Реализуйте шифрование методом перестановки по заданному ключу, при этом нечетные группы шифровать в исходном направлении, четные — в обратном.


15—16. Реализуйте маршрутную перестановку с использованием шифрующей таблицы 6×4. Маршрут: по вертикали, начиная с левого верхнего угла, поочередно сверху вниз и снизу вверх.

17—18. Зашифруйте исходное сообщение поворотной решеткой размером 6×5. Выполните поворот по часовой стрелке. Решетку выберите самостоятельно.


21—22. Зашифруйте исходное сообщение поворотной решеткой размером 5×8. Выполните поворот против часовой стрелки. Решетку выберите самостоятельно.

23—24. Реализуйте шифрование вертикальной перестановкой. Основа ключа — ваше собственное имя.

ШИФРОВАНИЕ МЕТОДОМ ГАММИРОВАНИЯ

Под гаммированием понимают процесс наложения по определенному закону гаммы шифра на открытые данные. Гамма шифра — это псевдослучайная последовательность, выработанная по заданному алгоритму, для шифровки открытых данных и дешифровки зашифрованных данных. Процесс шифровки заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы на исходный открытый текст обратимым образом, например с использованием операции сложения по модулю 2. Следует отметить, что перед шифровкой открытые данные разбивают на блоки $T_0^{(i)}$ одинаковой длины, обычно по 64 бита. Гамма шифра вырабатывается в виде последовательности блоков $\Gamma_w^{(i)}$ аналогичной длины.

Уравнение шифровки представим в виде

$$T_w^{(i)} = \Gamma_w^{(i)} \oplus T_0^{(i)}, \ i = 1, \ldots, M,$$

где $T_w^{(i)}$ — $i$-й блок шифротекста; $\Gamma_w^{(i)}$ — $i$-й блок гаммы шифра; $T_0^{(i)}$ — $i$-й блок открытого текста; $M$ — количество блоков открытого текста.

Процесс дешифровки сводится к повторной генерации гаммы шифра и наложению этой гаммы на зашифрованные данные. Уравнение дешифровки имеет вид

$$T_0^{(i)} = \Gamma_w^{(i)} \oplus T_w^{(i)}, \ i = 1, \ldots, M.$$

4.1. Методы генерации псевдослучайных последовательных чисел

При шифровании методом гаммирования в качестве ключа используется случайная строка битов, которая объединяется с открытым текстом, также представленным в двоичном виде, с помощью побитового сложения по модулю 2. В результате такого преобразования получается шифрованный текст. Для генерирования непредсказуемых двоичных последовательностей большой длины широко используются генераторы двоичных псевдослучайных последовательностей.
Генерируемые псевдослучайные ряды чисел часто называют гаммой шифра или просто гаммой. Обычно для генерации последовательности псевдослучайных чисел применяют компьютерные программы, которые, хотя и называются генераторами случайных чисел, на самом деле вырабатывают детерминированные числовые последовательности, по своим свойствам очень похожие на случайные. К криптографически стойкому генератору псевдослучайной последовательности чисел (гаммы шифра) предъявляются три основных требования:

- **период гаммы** должен быть достаточно большим для шифрования сообщений различной длины;
- **гамма** должна быть практически непредсказуемой, что означает невозможность предсказать следующий бит гаммы, даже если известны тип генератора и предшествующий фрагмент гаммы;
- **генерирование гаммы** не должно вызывать больших технических сложностей.

Длина периода гаммы является самой важной характеристикой генератора псевдослучайной последовательности чисел. По окончании периода числа начнут повторяться и их можно будет предсказать.

Наиболее часто применяется так называемый линейный конгруэнтный генератор псевдослучайных чисел (ПСЧ). Этот генератор вырабатывает последовательность ПСЧ \( Y_1, Y_2, \ldots, Y_{i-1}, Y_i, \ldots \) с использованием соотношения \( Y_i = (aY_{i-1} + b) \mod m \), где \( Y_i = i - e \) (текущее) число последовательности; \( Y_{i-1} \) — предыдущее число последовательности; \( a \) — множитель; \( b \) — приращение; \( Y_i \) — порождающее число (исходное значение). Текущее псевдослучайное число \( Y_i \) получают из предыдущего числа \( Y_{i-1} \) умножением его на коэффициент \( a \), сложением с приращением \( b \) и вычислением остатка от деления на \( m \). Данное уравнение генерирует ПСЧ с периодом повторения, зависящим от выбранных значений \( a \) и \( b \), и может достигать значения \( m \). Значение \( m \) обычно устанавливается равным \( 2^n \), где \( n \) — длина машинного слова в битах, либо равным простому числу, например \( m = 231 - 1 \). Как показано Д. Кнутом, линейный конгруэнтный генератор ПСЧ имеет максимальный период тогда и только тогда, когда \( b = a \mod 4 = 1 \).

Мультипликативный генератор вырабатывает последовательности чисел с помощью рекуррентного соотношения

\[ Y_i = (aY_{i-1}) \mod m. \]

Требования к значениям констант \( a \) и \( m \) такие же, как и для линейного конгруэнтного генератора.
Текущее случайное число $Y_i$ аддитивного датчика получается из суммы чисел $Y_{i-1}$ и $Y_{i-2}$ вычислением модуля от деления этой суммы на $m$:

$$Y_i = (Y_{i-1} + Y_{i-2}) \mod m.$$ 

Смешанный датчик вырабатывает последовательности чисел с помощью рекуррентного соотношения:

$$Y_i = (aY_{i-1} + \mu) \mod m.$$ 

4.1.1. Аддитивный генератор

На рис. 4.1 показан интерфейс программы, реализующей работу аддитивного генератора.

![Интерфейс программы](image)

Рис. 4.1. Окно программы, реализующей работу аддитивного генератора

Код программы:

```pascal
procedure TForm1.Btn1Click(Sender: TObject);
var
tocode, str: string;
Buff, TextSh: array [1..8] of integer;
i, j: integer;
ch: char;
begin
  m:=strtoint(edit4.text);
y0:=strtoint(edit5.text);
y1:=strtoint(edit6.text);
tocode:=edit1.Text;
j:=0;
repeat
  Rnd(Gamma); // формируется гамма
  for i:=1 to 8 do begin
    ...
  end;
end;
```

91
4.1.2. Линейный конгруэнтный генератор

На рис. 4.2 показан интерфейс программы, реализующей работу линейного конгруэнтного генератора.

Рис. 4.2. Окно программы, реализующей работу линейного конгруэнтного генератора

Код программы:

```pascal
procedure TForm1.BitBtn1Click(Sender: TObject);
var
tocode, str: string;
Buff, TextSh: array [1..8] of integer;
i, j: integer;
ch: char;
begin
  m:=strtoln(edit4.text);
y0:=strtoln(edit5.text);
```
a:=strtoint(edit6.text);
b:=strtoint(edit7.text);
tocode:=edit1.Text;
j:=0;
repeat
    Rnd(Gamma); //формируется гамма
    for i:=1 to 8 do begin
        ch:=tocode[i+j]; //считывается символ из исходника
        Buff[i]:=ord(ch); //запоминается его код
    end;
    for i:=1 to 8 do begin
        //блок текста складывается по модулю с гаммой шифра
        TextSh[i]:=Buff[i] xor Gamma[i];
        //записывается шифрованный символ
        str:=str+chr(TextSh[i]);
    end;
    inc (j,8); //переход к следующему блоку текста
until j=length(tocode);
edit2.Text:=str;
end;

4.1.3. Мультипликативный генератор

На рис. 4.3 показан интерфейс программы, реализующей работу мультипликативного генератора.

![Шифр гаммирования (мультипликативный)](image)

Рис. 4.3. Окно программы, реализующей работу мультипликативного генератора

Код программы:

```pascal
procedure TForm1.Button1Click(Sender: TObject);
var
```
to code, str: string;
Buff, TextSh: array [1..8] of integer;
i, j: integer;
ch: char;
begin
m:=strtoint(edit4.text);
y0:=strtoint(edit5.text);
a:=strtoint(edit6.text);
tocode:=edit1.Text;
j:=0;
repeat
  Rnd(Gamma); // формируется гамма
for i:=1 to 8 do begin
  ch:=tocode[i+j]; // считывается символ исходного текста
  Buff[i]:=ord(ch); // запоминается его код
end;
  for i:=1 to 8 do begin
    TextSh[i]:=Buff[i] xor Gamma[i];
    // блок текста складывается по модулю с гаммой шифра
    str:=str+chr(TextSh[i]);
    // записывается зашифрованный символ
  end;
  inc (j,8); // переход к следующему блоку текста
  // пока исходный файл не закончится
  until j=length(tocode);
  edit2.Text:=str;
end;

4.1.4. Смешанный генератор

На рис. 4.4 показан интерфейс программы, реализующей работу смешанного генератора.
Код программы:

procedure TForm1.BitBtn1Click(Sender: TObject);
var
tocode, str: string;
Buff, TextSh: array [1..8] of integer;
i, j: integer;
ch: char;
begin
  m:=strtoint(edit4.text);
  n:=strtoint(edit7.text);
  y0:=strtoint(edit5.text);
  a:=strtoint(edit6.text);
  tocode:=edit1.Text;
  j:=0;
4.2. Описание алгоритмов шифрования и дешифрования методом гаммирования

4.2.1. Алгоритм шифрования

1. Проинициализировать датчик случайных чисел выбранного типа.
2. Выделить блок открытого текста.
3. Сгенерировать гамму шифра.
4. Получить блок зашифрованного текста, сложив по модулю 2 блок открытого текста с гаммой шифра.
5. Если текст не закончился, перейти к п. 2, иначе — к п. 6.
6. Конец алгоритма шифрования.

4.2.2. Алгоритм дешифрования

1. Проинициализировать датчик случайных чисел выбранного типа.
2. Выделить блок зашифрованного текста.
3. Сгенерировать гамму шифра.
4. Получить блок открытого текста, сложив по модулю 2 блок зашифрованного текста с гаммой шифра.
5. Если зашифрованный текст не закончился, перейти к п. 2, иначе — к п. 6.
6. Конец алгоритма дешифрования.

4.3. Контрольные вопросы

1. Что такое гаммирование? Что понимают под гаммой шифра?
2. Какие операции можно применять при наложении гаммы? В чем заключается процесс шифрования и дешифрования?
3. Какие преобразования выполняются перед шифровкой над открытыми данными?
4. Какая характеристика является самой важной для генератора псевдослучайной последовательности чисел?
5. Каким образом формируются последовательности псевдослучайных чисел? Являются ли такие последовательности истинной совокупностью случайных чисел?
6. Какие требования предъявляются к криптографически стойкому генератору ПСЧ? Почему наиболее важна длина периода гаммы?
7. Опишите способ формирования последовательности ПСЧ с использованием линейного контргуэнтного генератора.
8. Опишите способ формирования последовательности ПСЧ с использованием аддитивного генератора.
9. Опишите способ формирования последовательности ПСЧ с использованием мультипликативного генератора.
10. Опишите алгоритмы шифрования и дешифрования открытого текста методом гаммирования.

4.4. Тесты по теме

«Шифрование методом гаммирования»

Вариант 1

1. На блоки какой длины разбивают открытые данные перед шифрованием методом гаммирования:
1. a) 8 бит; б) 16 бит; в) 64 бита?
2. Что используется в качестве ключа при шифровании методом гаммирования:
   a) кодовое слово; б) дополнительный алфавит; в) случайная строка битов?
3. Выберите вид формулы на которой основаны конгруэнтные методы:
   a) $n_{i+1} = (ln_i + m) \mod m$; б) $n_{i+1} = (ln_i + m) \mod 2$; в) $n_{i+1} = (ln_i) \mod m$.
4. Укажите основной недостаток линейного конгруэнтного генератора псевдослучайных чисел:
   a) сложная реализация алгоритмов; б) возможность восстановить всю последовательность псевдослучайных чисел по нескольким значениям; в) все числа являются абсолютно случайными.
5. Какое соотношение используется для линейного конгруэнтного генератора ПСЧ:
   a) $Y_i = (aY_{i-1} + b) \mod m$; б) $Y_i = (aY_{i-1}) \mod m$; в) $Y_i = (Y_{i-1} + Y_{i-2}) \mod m$?

Вариант 2
1. Основным достоинством шифрованием методом гаммирования является:
   a) невозможность предсказать следующий бит псевдослучайной последовательности; б) небольшая длина периода гаммы; в) можно предсказать следующий бит псевдослучайной последовательности.
2. Какой генератор является криптостойким:
   a) мультипликативный генератор; б) аддитивный генератор; в) ни один из вышеперечисленных генераторов?
3. Основное достоинство аддитивного генератора псевдослучайных чисел:
   a) наличие открытого ключа; б) большая длина периода ПСЧ;
в) криптостойкость.

4. Какой из генераторов можно назвать генератором Фибоначчи:
   а) аддитивный;
   б) мультипликативный;
   в) смешанный?

5. Какое рекуррентное соотношение используется для мультипликативного генератора ПСЧ:
   а) $Y_i = (Y_{i-1} + Y_{i-2}) \mod m$;
   б) $Y_i = (aY_{i-1} + \mu) \mod m$;
   в) $Y_i = (aY_{i-1}) \mod m$?

Вариант 3
1. Какой генератор псевдослучайных чисел чаще применяется для создания гаммы:
   а) аддитивный;
   б) мультипликативный;
   в) линейный конгренцный?

2. Какой генератор является криптостойким:
   а) мультипликативный;
   б) аддитивный;
   в) ни один из перечисленных генераторов?

3. Какой генератор ПСЧ позволяет увеличить длину периода:
   а) аддитивный;
   б) мультипликативный;
   в) смешанный?

4. На какой формуле основаны все конгренцтные методы:
   а) $n_{i+1} = (ln_i + m) \mod m$;
   б) $n_{i+1} = (ln_i + m) \mod 2$;
   в) $n_{i+1} = (ln_i) \mod m$?

5. Какое соотношение используется для аддитивного генератора ПСЧ:
   а) $Y_i = (aY_{i-1}) \mod m$;
   б) $Y_i = (Y_{i-1} + Y_{i-2}) \mod m$;
   в) $Y_i = (aY_{i-1} + \mu) \mod m$?

Вариант 4
1. Какой длины должны быть блоки гамма-шифра:
   а) равной длине блоков открытого текста;
   б) больше длины блоков открытого текста;
   в) меньше длины блоков открытого текста?

2. Какой генератор является криптостойким:
   а) мультипликативный;
б) аддитивный;
в) ни один из вышеперечисленных?
3. Какими достоинствами обладает линейный конгруэнтный генератор псевдослучайных чисел:
   а) все числа являются абсолютно случайными;
   б) простота и высокая скорость получения псевдослучайных значений;
   в) высокая криптоустойчивость?
4. Какой генератор ПСЧ позволяет увеличить длину периода:
   а) аддитивный;
   б) мультипликативный;
   в) смешанный?
5. Какое соотношение используется для смешанного генератора ПСЧ:
   а) \( Y_i = (aY_{i-1}) \bmod m \);
   б) \( Y_i = (Y_{i-1} + Y_{i-2}) \bmod m \);
   в) \( Y_i = (aY_{i-1} + \mu) \bmod m \)

Вариант 5
1. Какая операция применяется для объединения открытого текста с гамма-шифром:
   а) импликация;
   б) сложение по модулю 2;
   в) конъюнкция?
2. Какой из генераторов можно назвать генератором Фибоначчи:
   а) аддитивный;
   б) мультипликативный;
   в) смешанный?
3. В каком случае линейный конгруэнтный датчик псевдослучайных чисел имеет максимальный период \( Y_i = (aY_{i-1} - 1) \bmod m \), где \( Y_i \) — текущее число последовательности; \( Y_{i-1} \) — предыдущее число последовательности:
   а) \( b \) — нечетное, где \( b \) — приращение; \( a \) — множитель;
   б) \( b \) — четное, где \( b \) — приращение; \( a \) — множитель;
   в) не зависит от четности или нечетности величины \( b \)?
4. Какой генератор псевдослучайных чисел чаще применяется для создания гаммы:
   а) аддитивный;
   б) мультипликативный;
   в) линейный конгруэнтный?
5. Для какого генератора ПСЧ используется следующее соотношение: \( Y_i = (aY_{i-1} + b) \bmod m \), где \( Y_i \) — \( i \)-е (текущее) число последова-
тельности; $Y_{t-1}$ — предыдущее число последовательности; $a$ — множитель; $b$ — приращение; $Y_i$ — порождающее число (исходное значение):

а) линейный конгруэнтный датчик;

б) аддитивный генератор;

в) смешанный генератор?

<table>
<thead>
<tr>
<th>Вопрос</th>
<th>Вариант</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>в</td>
<td>в</td>
<td>а</td>
<td>б</td>
<td>а</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>а</td>
<td>в</td>
<td>б</td>
<td>а</td>
<td>в</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>в</td>
<td>в</td>
<td>а</td>
<td>а</td>
<td>б</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>а</td>
<td>в</td>
<td>б</td>
<td>а</td>
<td>в</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>б</td>
<td>а</td>
<td>б</td>
<td>в</td>
<td>а</td>
</tr>
</tbody>
</table>

4.5. Задание для самостоятельной работы по практической реализации алгоритмов шифрования методами гаммирования

Цель практического занятия: изучение теоретических основ криптографической защиты информации методами шифрования на базе алгоритмов гаммирования и приобретение навыков практической программной реализации данных алгоритмов.

Порядок выполнения задания.
Для нечетных вариантов предлагается реализовать процедуру шифрования файлов, для четных — дешифрования с использованием указанных методов.

Для генерации гаммы использовать датчик, соответствующий конкретному номеру варианта.

<table>
<thead>
<tr>
<th>№ вар.</th>
<th>Тип датчика</th>
<th>Исходные данные датчика</th>
</tr>
</thead>
<tbody>
<tr>
<td>1—2</td>
<td>Аддитивный</td>
<td>$M = 4096; Y_i = 4003, Y_1 = 59$</td>
</tr>
<tr>
<td>3—4</td>
<td>Линейный конгруэнтный</td>
<td>$a = 5, b = 7, m = 4096, Y_i = 4003$</td>
</tr>
<tr>
<td>5—6</td>
<td>Мультипликативный</td>
<td>$a = 7, m = 4096, Y_i = 502$</td>
</tr>
<tr>
<td>7—8</td>
<td>Аддитивный</td>
<td>$M = 4096 \cdot 4, Y_i = m - 5, Y_1 = 4091$</td>
</tr>
<tr>
<td>9—10</td>
<td>Мультипликативный</td>
<td>$A = 5, m = 4096, Y_i = m - 5$</td>
</tr>
<tr>
<td>11—12</td>
<td>Мультипликативный</td>
<td>$a = 5, m = 4096, Y_i = 3091$</td>
</tr>
<tr>
<td>13—14</td>
<td>Смешанный</td>
<td>$a = 2045, m = 4096, \mu = 1162, Y_i = 4001$</td>
</tr>
<tr>
<td>№ вар.</td>
<td>Тип датчика</td>
<td>Исходные данные датчика</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>15—16</td>
<td>Линейный конгруэнтный</td>
<td>$A = 9, b = 5, m = 4096, Y_i = 502$</td>
</tr>
<tr>
<td>17—18</td>
<td>Смешанный</td>
<td>$a = 165, m = 4096 \cdot 4, \mu = 3463, Y_i = 3887$</td>
</tr>
<tr>
<td>19—20</td>
<td>Аддитивный</td>
<td>$m = 4096 \cdot 4, Y_i = 3971, Y_1 = 1013$</td>
</tr>
<tr>
<td>21—22</td>
<td>Мультипликативный</td>
<td>$a = 9, m = 4096, Y_i — любое$</td>
</tr>
<tr>
<td>23—24</td>
<td>Линейный конгруэнтный</td>
<td>$a = 5, b = 5, m = 4096 \cdot 4, Y_i = 3215$</td>
</tr>
<tr>
<td>25—26</td>
<td>Аддитивный</td>
<td>$m = 4096, Y_i и Y_1 — любые$</td>
</tr>
</tbody>
</table>
СИСТЕМА СИММЕТРИЧНОГО ШИФРОВАНИЯ

В качестве системы симметричного шифрования рассмотрим шифрование методом Вернама. Шифр является разновидностью криптосистемы одноразовых блокнотов и служит примером системы с абсолютной криптографической стойкостью.

При этом система Вернама является частным случаем системы подстановок Виженера при \( m = 2 \). Конкретная версия этого шифра предложена в 1926 г. Пильбертом Вернамом. Старинный телетайп фирмы «AT&T» со считающим устройством Вернама и оборудованием для шифрования использовался для обеспечения связи в армии. Каждая буква исходного текста в алфавите, расширенном некоторыми дополнительными знаками, сначала переводилась с использованием телеграфного кода Бодо в пятибитовый блок \((b_0, b_1, ..., b_4)\). К исходному тексту Бодо добавлялся (по модулю 2) ключ \( k = (k_0, k_1, ..., k_{k-1})\), записанный на бумажной ленте.

В качестве информации, подлежащей шифрованию и дешифрованию, будут рассматриваться тексты, построенные на некотором алфавите.

Алфавит — конечное множество используемых для кодирования информации знаков. Текст — упорядоченный набор из элементов алфавита.

В качестве примеров алфавитов, используемых в современных ИС, можно привести следующие:
- алфавит \( Z_{33} \) — 32 буквы русского алфавита и пробел;
- алфавит \( Z_{256} \) — символы, входящие в стандартные коды ASCII и KOI-8;
- бинарный алфавит — \( Z_2 = \{0, 1\} \);
- восьмеричный алфавит или шестнадцатеричный алфавит.

Шифрование — преобразовательный процесс: исходный текст, который носит также название открытого текста, заменяется шифрованным текстом. Дешифрование — обратный шифрованию процесс. На основе ключа шифрованный текст преобразуется в исходный.
Ключ — информация, необходимая для беспрепятственного шифрования и дешифрования текстов.

Для абсолютной криптографической стойкости ключ должен обладать тремя критически важными свойствами:
• иметь случайное равномерное распределение;
• совпадать по размеру с заданным открытым текстом;
• применяться только один раз.

5.1. Моделирование кодера и декодера

Шифрование исходного текста, предварительно преобразованного в последовательность двоичных символов х, осуществляется путем сложения по модулю 2 символов х с последовательностью двоичных ключей k. Получим символы шифртекста: y = x ⊕ k.

Расшифровка состоит в сложении по модулю 2 символов шифртекста с той же последовательностью ключей k:

\[ y ⊕ k = x ⊕ k ⊕ k = x. \]

При этом последовательности ключей, использованные при шифровании и расшифровании, компенсируют друг друга (при сложении по модулю 2), и в результате восстанавливаются символы x исходного текста.

При моделировании преобразование исходного текста в последовательность двоичных символов можно осуществить следующим образом.

1. Создать новый проект Lab. На форме Form1 разместить следующие элементы:
   • текстовое поле textBox1 — для ввода исходного текста;
   • текстовое поле textBox2 — для ввода двоичного ключа;
   • кнопку Button — для инициализации процедуры шифрования;
   • текстовое поле textBox3 — для вывода зашифрованного текста.

На рис. 5.1 показан интерфейс программы, иллюстрирующей эти действия.

2. Перейти в Form1.cs.

Задать следующую константу:

```csharp
const int constant = sizeof(char) * 8;
```

где 8 — количество бит в байте.

3. Создать функцию перевода числа из десятичной системы счисления в двоичную:

```csharp
// функция перевода числа из десятичной системы счисления
// в двоичную:
```
private string IntToBin(int value)
{
    string sDv = string.Empty;
    int iMod;
    int iDch;

    iDch = value;
    while (iDch>=2)
    {
        iMod = iDch % 2; // остаток от деления
        iDch = iDch / 2; // целочисленное деление
        sDv = Convert.ToString(iMod) + sDv;
    }
    sDv = Convert.ToString(iDch) + sDv;
    return sDv;
}

4. Далее создать процедуру Shifr, осуществляющую моделирование и шифрование исходной строки:

/*Процедура моделирования исходной строки посимвольно в двоичное число и ее шифрование*/
private void Shifr()
{
    int amount= textBox1.Text.Length * constant;
    int[] m = new int[amount];
    int d = 0;
    int c = 0;

    for (int i = 1; i<=textBox1.Text.Length; i++)
    {
        string st = string.Empty;
        st = IntToBin((int)(textBox1.Text[i - 1]));
        int pc = 0;

        // Дальше выполняется процесс шифрования...
    }
}
for (int j = d; j <= (c + st.Length-1);)
{
    m[d] = (int) st[pc];
    pc++;
    d++;
}
c += st.Length;

int dm = c;
int dpk = textBox2.Text.Length;
it I = 0;
it J = 0;
string s = string.Empty;

while (I <= dm-1)
{
    s += Convert.ToString(m[I] ^ Convert.ToInt32 (textBox2.Text[J]));
    ToInt32(textBox2.Text[J]));
    if (J < dpk-1)
        J++;
    else
        J = 0;
    I++;
}

где m — массив двоичных символов; dm — длина массива, dpk — длина последовательности ключей; pc, d — вспомогательные переменные.

5. Запустить программу на выполнение.

На рис. 5.2 показан интерфейс программы, реализующей схему шифрования методом Вернама.

Рис. 5.2. Окно программы, реализующей схему шифрования методом Вернама
Декодер полностью совпадает с кодером. Метод Вернама не зависит от длины последовательности ключей и позволяет использовать случайную последовательность ключей. Однако при реализации метода возникают серьезные проблемы, связанные с необходимостью доставки получателю такой же последовательности ключей, как и у отправителя, либо с необходимостью безопасного хранения идентичных последовательностей ключей у отправителя и получателя. Последовательность ключей можно реализовать как последовательность псевдослучайных чисел. В этом случае необходимо синхронизировать только параметры псевдослучайного генератора.

Примером такого датчика может быть аддитивный:

\[ y_{n+1} = (y_n + y_{n-1}) \% m, \]

где \( \% \) — остаток от деления.

Параметры датчика можно задать следующими:

\[ m = 4096 \cdot 4; y_1 = 4091; y_2 = m - 5. \]

В настоящее время система шифрования Вернама используется достаточно редко. В большой степени из-за существенного размера ключа, длина которого должна совпадать с длиной сообщения. То есть использование таких шифров требует огромных затрат на производство, хранение, уничтожение ключевых материалов. Тем не менее совершенно стойкие шифры типа Вернама все же нашли практическое применение для защиты особо важных линий связи с относительно небольшим объемом информации. Так, например, англичане и американцы использовали шифры типа Вернама во время Второй мировой войны. Шифр Вернама по модулю 2 использовался на правительственной «горячей линии» между Вашингтоном и Москвой, где ключевые материалы представляли собой бумажные ленты, на которые знаки ключевой последовательности наносились с помощью перфорации.

На практике можно один раз физически передать носитель информации с длинным истинно случайным ключом, а потом по мере необходимости пересылать сообщения. На этом основана идея шифроблокнотов: шифровальщик по дипломатической почте или при личной встрече снабжается блокнотом, каждая страница которого содержит ключи. Такой же блокнот есть и у принимающей стороны. Использованные страницы уничтожаются.

Наряду с отмеченным свойством криптостойкости, система шифрования Вернама при использовании на практике обладает рядом существенных недостатков. К ним можно отнести следующие.
• Для работы шифра Вернама необходима истинно случайная последовательность (ключ). По определению, последовательность, полученная с использованием любого алгоритма, является не истинно случайной, а псевдослучайной. Для получения распределения, предельно близкого к равномерному, случайная последовательность обычно пропускается через хэш-функцию.

• Проблемой использования шифра Вернама является отсутствие подтверждения подлинности и целостности сообщения. Получатель не может удостовериться в отсутствии повреждений или в подлинности отправителя. Решением проблемы является применение хэш-функции. От открытого текста вычисляется хэш-функция, и ее значение шифруется вместе с сообщением. При изменении сообщения значение хэш-функции изменится, поэтому, даже если злоумышленник заполучил шифроблокнот, не зная алгоритм вычисления хэш-функции, он не сможет использовать его для передачи информации.

• Необходимо наличие достаточного количества ключей, которые могут понадобиться для шифрования больших объемов открытого текста. Реальный же объем текста зачастую трудно оценить заранее, в особенности это касается дипломатической и военной сфер, где ситуация способна меняться быстро и непредсказуемо. Это может приводить к нехватке ключей, что может заставить шифровальщика либо использовать ключ повторно, либо полностью прервать шифрованную связь.

• Проблемой является защищенная передача последовательности и сохранение ее в тайне. Если существует надежно защищенный от перехвата канал передачи сообщений, шифры вообще не нужны: секретные сообщения можно передавать по этому каналу. Если же передавать ключ системы Вернама с помощью другого шифра, то полученный шифр окажется защищенным ровно настолько, насколько защищен другой шифр. При этом, поскольку длина ключа та же, что и длина сообщения, передать его не просто, чем сообщение. Шифроблокнот на физическом носителе можно украсть или скопировать.

• Шифр Вернама чувствителен к любому нарушению процедуры шифрования. В случае когда одна и та же страница шифроблокнота по различным причинам применяется дважды, пользователь сталкивается с надежностью своей шифропереписки.
5.2. Контрольные вопросы

1. В чем заключается шифрование методом Вернама?
2. Каким образом можно реализовать бесконечную последовательность ключей?
3. В каком виде должен быть текст перед непосредственным шифрованием?
4. Какой математический аппарат использован при формировании шифротекста?
5. Перечислите критически важные свойства ключа для обеспечения абсолютной криптографической стойкости шифрования.
6. В чем суть процедуры моделирования кодера?
7. В чем суть процедуры моделирования декодера?
8. В чем суть шифрования с использованием одноразовых блокнотов?
9. В чем проблема использования псевдослучайной последовательности при формировании шифротекста?
10. Перечислите проблемы при использовании системы шифрования Вернама.

5.3. Тесты по теме
«Шифрование методом гаммирования»

Вариант 1

1. Система Вернама — частный случай системы:
   а) подстановок Виженера;
   б) подстановки Эйлера;
   в) подстановки Виета.
2. Символы, входящие в стандартные коды ASCII и КОИ-8:
   а) $Z_2 = \{0, 1\}$;
   б) $Z_33$;
   в) $Z_{256}$.
3. Что рассматривается в качестве информации, подлежащей шифрованию и дешифрованию:
   а) числа;
   б) тексты;
   в) графика?
4. Шифрование исходного текста осуществляется:
   а) сложением по модулю 2;
   б) сложением;
   в) делением.
5. Последовательность ключей можно реализовать как:
   а) упорядоченный набор элементов алфавита;
   б) последовательность псевдослучайных чисел;
   в) последовательность случайных чисел.
Вариант 2
1. Как осуществляется сложение по модулю 2:
   a) \( y = x \oplus k \);
   b) \( y = x + k \);
   в) \( y = |x + k| \)?
2. Расшифровка сложения по модулю 2:
   a) \( |x + k + k| = |x| \);
   b) \( x + k + k = x \);
   в) \( x \oplus k \oplus k = x \).
3. Остаток от деления обозначается как:
   a) \( \text{div} \);
   b) \( \text{mod} \);
   в) \( m \).
4. Конечное множество используемых для кодирования информации знаков — это:
   a) код;
   b) алфавит;
   в) текст.
5. Исходный текст также называют:
   a) открытым текстом;
   b) шифрованным текстом;
   в) закрытым текстом.

Вариант 3
1. Информация, необходимая для беспрепятственного шифрования и дешифрования текстов, — это:
   a) алфавит;
   b) код;
   в) ключ.
2. Что рассматривается в качестве информации, подлежащей шифрованию и дешифрованию:
   a) числа;
   b) тексты;
   в) графика?
3. Открытый текст — это:
   a) начальный текст;
   б) исходный текст;
   в) шифрованный текст.
4. Алфавит — это:
   а) упорядоченный набор элементов;
   б) информация, необходимая для беспрепятственного шифрования и дешифрования текстов;
конечное множество используемых для кодирования информации знаков.
5. Сложение по модулю 2 используется для:
а) шифрования текста;
б) дешифрования текста;
в) моделирования текста.

Вариант 4
1. Исходный текст также называют:
а) открытым текстом;
б) шифрованным текстом;
в) закрытым текстом.
2. Шифрование исходного текста осуществляется:
а) сложением по модулю 2;
б) сложением;
в) делением.
3. Что рассматривается в качестве информации, подлежащей шифрованию и дешифрованию:
а) числа;
б) тексты;
в) графика?
4. Конечное множество используемых для кодирования информации знаков — это:
а) код;
б) алфавит;
в) текст.
5. Как осуществляется сложение по модулю 2:
а) \( y = x \oplus k \);
б) \( y = x + k \);
в) \( y = |x + k| \)?

Вариант 5
1. Последовательность ключей можно реализовать как:
а) упорядоченный набор элементов алфавита;
б) последовательность псевдослучайных чисел;
в) последовательность случайных чисел.
2. Расшифровка сложения по модулю 2:
а) \( |x + k + k| = |x| \);
б) \( x + k + k = x \);
в) \( x \oplus k \oplus k = x \).
3. Алфавит — это:
а) упорядоченный набор элементов;
б) информация, необходимая для беспрепятственного шифрования и дешифрования текстов;
в) конечное множество используемых для кодирования информации знаков.
4. Информация, необходимая для беспрепятственного шифрования и дешифрования текстов:
а) алфавит;
б) код;
в) ключ.
5. Система Вернама — частный случай системы:
а) подстановок Вижинера;
б) подстановки Эйлера;
в) подстановки Виета.

Таблица правильных ответов

<table>
<thead>
<tr>
<th>Вопрос</th>
<th>Вариант</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>а</td>
<td>а</td>
<td>в</td>
<td>а</td>
<td>б</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>в</td>
<td>в</td>
<td>б</td>
<td>а</td>
<td>в</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>б</td>
<td>б</td>
<td>б</td>
<td>б</td>
<td>в</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>а</td>
<td>б</td>
<td>в</td>
<td>б</td>
<td>в</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>б</td>
<td>а</td>
<td>а</td>
<td>а</td>
<td>а</td>
</tr>
</tbody>
</table>

5.4. Задание для самостоятельной работы по практической реализации алгоритмов шифрования методом Вернама

Цель практического занятия: изучение теоретических основ криптографической защиты информации методами шифрования на базе алгоритма системы Вернама и приобретение навыков практической программной реализации данных алгоритмов.

Порядок выполнения задания.
Для нечетных вариантов предлагается реализовать процедуру шифрования файлов, для четных — дешифрования с использованием указанного метода.

Варианты заданий

1—3. Смоделировать кодер с последовательностью ключей из пяти элементов.
2—4. Смоделировать декодер с последовательностью ключей из пяти элементов.
5—7. Смоделировать кодер с последовательностью ключей из 17 элементов.
6—8. Смоделировать декодер с последовательностью ключей из 17 элементов.
9—11. Смоделировать кодер с бесконечной последовательностью ключей, формируемых с помощью аддитивного псевдослучайного датчика случайных чисел.
10—12. Смоделировать декодер с бесконечной последовательностью ключей, формируемых с помощью аддитивного псевдослучайного датчика случайных чисел.
13—15. Смоделировать кодер с последовательностью ключей из семи элементов, формируемых с помощью аддитивного псевдослучайного датчика случайных чисел.
14—16. Смоделировать декодер с последовательностью ключей из семи элементов, формируемых с помощью аддитивного псевдослучайного датчика случайных чисел.
17—19. Смоделировать кодер с бесконечной последовательностью ключей, формируемых с помощью стандартного датчика случайных чисел.
18—20. Смоделировать декодер с бесконечной последовательностью ключей, формируемых с помощью стандартного датчика случайных чисел.
21—22. Смоделировать кодер с последовательностью ключей из 13 элементов, формируемых с помощью стандартного датчика случайных чисел.
23—25. Смоделировать декодер с последовательностью ключей из 13 элементов, формируемых с помощью стандартного датчика случайных чисел.
24—26. Смоделировать кодер с бесконечной последовательностью ключей, формируемых с помощью аддитивного псевдослучайного датчика случайных чисел. Параметры датчика нужно взять следующими:

\[ m = 4096 \cdot 2; \ y_1 = 4091; \ y_2 = m - 3. \]
Тема 6

РЕЗУЛЬТИРУЮЩИЕ ТЕСТЫ
ПО ВСЕМ РАЗДЕЛАМ ДИСЦИПЛИНЫ

Вариант 1
1. Чем отличается полибианский квадрат от шифрующей таблицы Трисемуса:
   а) наличием ключевого слова во множестве шифрообозначений у алгоритма полибианского квадрата;
   б) наличием ключевого слова в начале множества шифрообозначений таблицы Трисемуса;
   в) наличием ключевого слова в конце множества шифрообозначений таблицы Трисемуса?
2. Какой из генераторов можно назвать генератором Фибоначчи:
   а) аддитивный;
   б) мультипликативный;
   в) смешанный?
3. Открытый текст — это:
   а) начальный текст;
   б) исходный текст;
   в) шифрованный тест.
4. Какой шифр носит второе название «шифр Древней Спарты»:
   а) шифр перестановки «Сцитала»;
   б) шифр «Поворотная решетка»;
   в) шифр маршрутной перестановки?
5. Из скольких элементов состоит множество шифрообозначений в шифре простой замены:
   а) одного;
   б) двух;
   в) трех?
6. Сколько возможными способами вырезы трафарета полностью покрывают площадь листа при использовании шифра «Поворотная решетка»:
   а) тремя;
   б) четырьмя;
   в) шесть?
7. Кодом называется:
а) последовательность знаков;
б) система условных обозначений или сигналов;
в) двоичное число фиксированной длины.
8. При символьном кодировании:
а) кодируется каждый символ защищаемого сообщения;
б) символы защищаемого сообщения меняются местами в соответствии с днем недели;
в) закодированное сообщение имеет вполне определенный смысл (слова, предложения, группы предложений).
9. Какое число в сумме по каждому столбцу, каждой строке и каждой диагонали дают последовательные натуральные числа начиная с единицы, вписанные в клетки квадратных таблиц с использованием шрифта на основе магических квадратов:
а) одно и то же;
б) четное;
в) нечетное?
10. Какой шифр сложной замены основан на смещении по алфавиту в соответствии с ключом шифра:
а) шифр Гронсфельда;
б) шифр Виженера;
в) шифр Виженера с автоключом?

Вариант 2
1. В какую часть множества шифрообозначений записывается «лозунг» для лозунгового шифра:
а) в середину;
б) в конец;
в) в начало?
2. Какой длины должны быть блоки гамма-шифра:
а) равной длине блоков открытого текста;
б) больше длины блоков открытого текста;
в) меньше длины блоков открытого текста?
3. Алфавит — это:
а) упорядоченный набор элементов;
б) информация, необходимая для беспредметственного шифрования и дешифрования текстов;
в) конечное множество используемых для кодирования информации знаков.
4. Сколькими способами наложения на таблицу трафарета с вырезанными ячейками происходит шифрование/дешифрование в шифре «Поворотная решетка»:

114
5. Как заполняется прямоугольная таблица буквами алфавита в полибийском квадрате:
   а) в случайном порядке;
   б) по алгоритму системы подстановок;
   в) с помощью замен?
6. Что делают, если ключ оказался короче сообщения в системе шифрования Виженера:
   а) выписывают в строку;
   б) циклические повторяют;
   в) пропускают?
7. Конечное множество используемых для кодирования информации знаков — это:
   а) алфавит;
   б) текст;
   в) шифр.
8. Шифр, у которого буквы открытого текста не замещаются на другие, а меняется порядок их следования, называется:
   а) перестановка;
   б) подстановка;
   в) гаммирование.
9. Где используются ключи, состоящие только из цифр:
   а) шифр Виженера;
   б) шифр Гронсфельда;
   в) двойной квадрат Уитстона?
10. Где используются ключи, состоящие только из букв:
    а) шифр Виженера;
    б) двойной квадрат Уитстона;
    в) шифр Гронсфельда с автоключом?

Вариант 3
1. Омофоны могут быть представлены:
   а) трехразрядными числами от 000 до 999;
   б) двухразрядными числами от 00 до 99;
   в) трехразрядными числами от 111 до 999.
2. Что используется в качестве ключа при шифровании методом гаммирования:
   а) кодовое слово;
   б) дополнительный алфавит;
   в) случайная строка битов?
3. Сложение по модулю 2 используется для:
   а) шифрования текста;
   б) дешифрования текста;
   в) моделирования текста.
4. Шифр перестановки — это:
   а) шифр преобразования, с помощью которого изменяют и порядок следования символов исходного текста, и сами символы;
   б) шифр преобразования, с помощью которого изменяют только порядок следования символов исходного текста, не изменяя их самих;
   в) шифр преобразования, с помощью которого не изменяют порядок следования символов исходного текста, а изменяют сами символы.
5. Как многоалфавитная подстановка последовательно меняет используемые алфавиты:
   а) линейно;
   б) циклически;
   в) разветвляющее?
6. На что разбивается исходное сообщение перед шифрованием в двойном квадрате Уитстона:
   а) анаграммы;
   б) монограммы;
   в) биграммы?
7. Кодом называется:
   а) последовательность знаков;
   б) система условных обозначений или сигналов;
   в) двоичное число фиксированной длины.
8. Какой из генераторов можно назвать генератором Фибоначчи:
   а) аддитивный;
   б) мультипликативный;
   в) смешанный?
9. Какое соотношение используется для смешанного генератора ПСЧ:
   а) \( Y_i = (aY_{i-1}) \mod m \);
   б) \( Y_i = (Y_{i-1} + Y_{i-2}) \mod m \);
   в) \( Y_i = (aY_{i-1} + \mu) \mod m \)?
10. Какой шифр сложной замены основан на смещении по алфавиту в соответствии с ключом шифра:
    а) шифр Гронсфельда;
    б) шифр Вишинера;
    в) шифр Вишинера с автоключом?
Вариант 4
1. На какой символ заменяется символ исходного сообщения в полибианском квадрате:
   a) расположенный ниже в том же столбце таблицы;
   b) расположенный слева в той же строке таблицы;
   v) расположенный выше в том же столбце таблицы?
2. Какую операцию выполняют в начале шифрования методом гаммирования:
   a) разбиение открытого текста на блоки равной длины;
   b) сложение по модулю 2;
   в) генерирование гамма-шифра?
3. Остаток от деления:
   a) div;
   b) mod;
   v) m.
4. В каком из шифров преобразования состоят в том, что отрезок открытого текста записывается в фигуру по некоторой одной траекtorии, а выписывается по другой траектории:
   a) шифр Цезаря;
   b) биграммный шифр Плейфера;
   в) шифр маршрутной перестановки?
5. Какое соотношение используется для смешанного генератора ПСЧ:
   a) \( Y_i = (aY_{i-1}) \mod m \);
   b) \( Y_i = (Y_{i-1} + Y_{i-2}) \mod m \);
   в) \( Y_i = (aY_{i-1} + \mu) \mod m \)?
6. Какие шифры называют шифрами замены:
   a) которые осуществляются путем замены каждого символа открытого сообщения на другие символы — шифрообозначения;
   b) которые осуществляются путем замены всего алфавита сообщения;
   в) которые осуществляются путем замены конечного сообщения?
7. Расшифровка сложения по модулю 2:
   a) \( |x + k + k| = |x| \);
   b) \( x + k + k = x \);
   v) \( x \oplus k \oplus k = x \).
8. Сколько возможными способами вырезы трафарета полностью покрывают площадь листа при использовании шифра «Поворотная решетка»:
   a) тремя;
   b) четыремя;
в) шести?
9. Достоинством шифрованием методом кодирования является:
   а) невозможность предсказать следующий бит псевдослучайной последовательности;
   б) небольшая длина периода гаммы;
   в) можно предсказать следующий бит псевдослучайной последовательностью.
10. В каком шифре используются биграммы:
   а) шифр Виженера;
   б) двойной квадрат Уитстона;
   в) шифр Гронсфельда с автоключом?

Вариант 5
1. Какое количество букв должен иметь текст при процедуре шифрования методом биграммного шифра Плейфера:
   а) четное число;
   б) нечетное число;
   в) не имеет значения?
2. На какой формуле основаны конгруэнтные методы формирования датчиков:
   а) \( n_{i+1} = ln_i + m \mod m \);
   б) \( n_{i+1} = ln_i + m \mod 2 \);
   в) \( n_{i+1} = ln_i \mod m \)?
3. Каким термином обозначается конечное множество используемых для кодирования информации знаков:
   а) код;
   б) алфавит;
   в) текст?
4. Какое число в сумме по каждому столбцу, каждой строке и каждой диагонали дают последовательные натуральные числа, начиная с единицы, вписанные в клетки квадратных таблиц с использованием шрифта на основе магических квадратов:
   а) одно и то же;
   б) четное;
   в) нечетное?
5. Достоинством шифрованием методом кодирования является:
   а) невозможность предсказать следующий бит псевдослучайной последовательности;
   б) небольшая длина периода гаммы;
   в) можно предсказать следующий бит псевдослучайной последовательностью.
6. Шифр, у которого буквы открытого текста не замещаются на другие, а меняется порядок их следования, называется:
   а) перестановка;
   б) подстановка;
   в) гаммирование.
7. Символы, входящие в стандартные коды ASCII и КОИ-8:
   а) \( Z_2 = \{0, 1\} \);
   б) \( Z_{33} \);
   в) \( Z_{256} \).
8. Сколькими способами наложения на таблицу трафарета с вырезанными ячейками происходит шифрование/дешифрование в шифре «Поворотная решетка»:
   а) 3;
   б) 4;
   в) 5?
9. Какие действия выполняются в том случае, если ключ оказался короче сообщения в системе шифрования Вижинера:
   а) выписывают в строку;
   б) циклические повторяют;
   в) пропускают?
10. В каких шифрах используются ключи, состоящие только из букв:
    а) шифр Вижинера;
    б) двойной квадрат Уитстона;
    в) шифр Гронсфельда с автоключом?

<table>
<thead>
<tr>
<th>Вариант</th>
<th>Вопрос</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>б</td>
<td>а</td>
<td>б</td>
<td>а</td>
<td>а</td>
<td>б</td>
<td>в</td>
<td>а</td>
<td>а</td>
<td>а</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>в</td>
<td>а</td>
<td>в</td>
<td>б</td>
<td>а</td>
<td>а</td>
<td>б</td>
<td>а</td>
<td>б</td>
<td>в</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>а</td>
<td>в</td>
<td>а</td>
<td>б</td>
<td>б</td>
<td>в</td>
<td>в</td>
<td>а</td>
<td>в</td>
<td>б</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>а</td>
<td>а</td>
<td>б</td>
<td>в</td>
<td>в</td>
<td>а</td>
<td>в</td>
<td>б</td>
<td>а</td>
<td>а</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>а</td>
<td>а</td>
<td>б</td>
<td>а</td>
<td>а</td>
<td>а</td>
<td>в</td>
<td>б</td>
<td>б</td>
<td>в</td>
</tr>
</tbody>
</table>
СЛОВАРЬ КРИПТОГРАФИЧЕСКИХ ТЕРМИНОВ

А

Автомат шифрующий [ciphering automaton] — автомат, реализующий шифрование. Точнее, автомат, зависящий от параметра, реализующего зашифрование или расшифрование. Параметр А.ш. (ключ) принимает конечное множество значений. От ключа могут зависеть начальное состояние, функции переходов и выходов. При каждом фиксированном значении ключа А.ш. становится обратимым инициальным автоматом.


Алгоритм зашифрования [encryption algorithm] — алгоритм криптографический, реализующий функцию зашифрования. В случае шифрования блочных получается использованием алгоритма зашифрования блочного базового в конкретном режиме шифрования.

Алгоритм зашифрования блочный базовый [basic block encryption algorithm] — алгоритм зашифрования, реализующий при каждом фиксированном значении ключа одно обратимое отображение множества блоков текста открытого, имеющих фиксированную длину. Представляет собой алгоритм простой замены блоков текста фиксированной длины.

Алгоритм зашифрования поточный [stream encryption algorithm, stream cipher] — алгоритм зашифрования, реализующий при каждом фиксированном значении ключа последовательность обратимых отображений (вообще говоря, различных), действующую на последовательность блоков текста открытого.

Алгоритм кодирования имитозащищающего [integrity protection algorithm] — алгоритм криптографический преобразования информации, обеспечивающий контроль ее целостности (как правило, за счет внесения избыточности). В отличие от алгоритма формирования подписи цифровой использует криптосистемы симметричные. Примерами А.к.и. являются код аутентификации, некоторые автоматные преобразования и алгоритмы шифрования.

Алгоритм криптографический [cryptographic algorithm] — алгоритм, реализующий вычисление одной из функций криптографических.
Алгоритм проверки подписи цифровой [signature verification algorithm] — составная часть схемы подписи цифровой. Алгоритм, на вход которого подаются подпись цифровая, ключ открытый и другие открытые параметры схемы подписи цифровой. В схемах подписи цифровой с восстановлением сообщения результатом работы алгоритма является заключение о корректности подписи и, если она корректна, — само сообщение, извлеченное из подписи. В остальных случаях сообщение является частью входных данных и алгоритм проверки выдает лишь заключение о корректности подписи. В некоторых разновидностях схемы подписи цифровой при проверке подписи используется протокол интерактивный.

Алгоритм расшифрования [decryption algorithm, deciphering] — алгоритм криптографический, обратный к алгоритму зашифрования и реализующий функцию расшифрования.

Алгоритм формирования подписи цифровой [signature generation algorithm, син. алгоритм генерации подписи цифровой] — составная часть схемы подписи цифровой. Алгоритм (вообще говоря, рандомизированный), на вход которого подаются подписываемое сообщение, ключ секретный, а также открытые параметры схемы подписи цифровой. Результатом работы алгоритма является подпись цифровая. В некоторых разновидностях схемы подписи цифровой при формировании подписи используется протокол.

Алгоритм шифрования [encryption algorithm] — под ним в зависимости от контекста понимается алгоритм зашифрования или алгоритм расшифрования.

Алгоритм шифрования инволютивный [involutive encryption algorithm] — алгоритм шифрования, для которого алгоритмы зашифрования и расшифрования совпадают. Другими словами, если к тексту открытому дважды применить алгоритм зашифрования, то получится тот же самый открытый текст. Исторически для таких алгоритмов употребляется название «обратимый», но правильно называть их именно «инволютивными» в соответствии с общим пониманием инволюции в математике.

Алгоритм шифрования итеративный [iterative encryption algorithm] — алгоритм шифрования, для которого соответствующие алгоритм зашифрования и алгоритм расшифрования состоят из последовательных однотипных циклов шифрования. Подобные алгоритмы относительно просто реализуются и позволяют обеспечивать, в частности, свойство перемешивания, свойство рассеивания, свойство усложнения.

Анализ криптографический [cryptanalysis, син. криптоанализ] — исследование системы криптографической с целью получения обосно-
ванных оценок ее стойкости криптографической. Результаты А.к. могут использоваться разработчиком и пользователем законным криптосистемы для оценки эффективности системы защиты информации от потенциального противника и/или нарушителя, а потенциальным противником и/или нарушителем — для подготовки и реализации атаки на криптосистему. А. к. проводится путем исследования криптосистемы, а также моделирования (выполнения) различных атак на криптосистему.

Аналіз трафіка [traffic analysis] — анализ совокупности сообщений шифрованных, передаваемых по системе связи, не приводяший к дешифрованию, но позволяющий противнику и/или нарушителю получить косвенную информацию о передаваемых сообщениях открытых и в целом о функционировании наблюдаемой системы связи. А.т. использует особенности оформления сообщений шифрованных, их длину, время передачи, данные об отправителе и получателе и т.п.

Анонимность [anonymity] — понятие, родственное неотслеживаемости. Выражает предоставляемую участникам (протокола) возможность выполнять какое-либо действие анонимно, т.е. не идентифицируя себя. При этом, однако, участник обязан доказать свое право на выполнение этого действия. Анонимность бывает абсолютной и отзываемой. В последнем случае в системе есть выделенный участник, арбитр, который при определенных условиях может нарушать анонимность и идентифицировать участника, выполнившего данное конкретное действие.

Апроксимация афнимная [affine approximation] — см. Приближение аффинное.

Апроксимация аффинная наилучшая [the best affine approximation] — см. Приближение аффинное наилучшее.

Арбитр [arbiter] — участник протокола с арбитром, выполняющий процедуру арбитража.

Арбитраж [arbitration] — формализованная процедура разрешения споров о трактовке результатов выполнения протокола криптографического. Такая процедура необходима для многих протоколов криптографических прикладных, в том числе схем подписи цифровой, протоколов подписания контракта, систем платежей электронных и т.п., и должна рассматриваться как неотъемлемая часть этих протоколов. Для самой процедуры арбитража требуется либо алгоритм, выполняемый арбитром на входных данных, предоставленных ему заявителем (заявителями), либо специальный протокол с участием всех заинтресованных сторон.
Атака адаптивная [adaptive attack] — атака на криптосистему, при которой характер воздействия противника и/или нарушителя может изменяться во времени в зависимости от действий пользователей законных криптосистемы или от других условий. Например, противник может подбирать различные исходные данные для воздействия на криптосистему.

Атака адаптивная на основе подобранного сообщения открытого [adaptive chosen-plaintext attack, син. атака адаптивная по сообщению выбранному] — атака адаптивная, при которой противник и/или нарушитель имеет возможность вынудить пользователя законного криптосистемы к обработке (расшифрованию) некоторого сообщения открытого, выбранного противником и/или нарушителем, и наблюдать соответствующее сообщение шифрованное.

Атака адаптивная на основе подобранного сообщения шифрованного [adaptive chosen-ciphertext attack] — атака адаптивная, при которой противник и/или нарушитель имеет возможность вынудить пользователя законного криптосистемы к обработке (расшифрованию) некоторого сообщения шифрованного, выбранного противником и/или нарушителем, и наблюдать соответствующее сообщение открытого.

Атака адаптивная по сообщению выбранному [adaptive chosen-plaintext attack] — см. Атака адаптивная на основе подобранного сообщения открытого.

Атака активная [active attack] — атака на криптосистему или на протокол криптографический, при которой противник и/или нарушитель может влиять на действия пользователя законного, например подменять или удалять сообщения пользователя законного, создавать и передавать сообщения от его имени и т.п.

Атака «встреча посередине» [meet-in-the-middle attack] — атака на криптосистему, основанная на методе «встреча посередине».

Атака дифференциальная [differential attack] — см. Атака разностная.


Атака корреляционная [correlation attack] — атака на криптосистему, основанная на методе корреляционном.

Атака линейная [linear attack] — атака на криптосистему, основанная на методе линейном.

Атака лобовая [brute-force attack] — атака на криптосистему, основанная на методе полного (тотального) опробования ключей.

Атака на криптосистему [attack on the cryptosystem] — попытка противника и/или нарушителя понизить уровень безопасности кон-
кретной системы криптографической на основе определенных мето-
dов криптоанализа и при некоторых предположениях криптоанализа. Разработчик, пользователь законный и противник при проведении анализа криптографического моделируют атаки на криптосистему. Совокупность различных атак постоянно расширяется за счет развития теоретических методов и возможностей техники.

Атака на криптосистему на основе известного текста открытого [known plaintext attack] — атака на криптосистему, при которой противнику и/или нарушителю известен текст открытый.

Атака на криптосистему на основе только текста шифрованного [ciphertext-only attack] — атака на криптосистему, при которой противнику и/или нарушителю известен текст шифрованный и не известен текст открытый.

Атака на основе ключей эквивалентных [equivalent keys attack] — атака на криптосистему, основанная на методе ключей эквивалентных.

Атака на протокол криптографический [attack on the protocol] — попытка проведения анализа сообщений протокола и/или выполнения не предусмотренных протоколом действий с целью нарушения работы протокола и/или получения информации, составляющей секрет его участников.

Атака на протокол с передачей повторной [replay attack] — атака на протокол криптографический, при которой противник и/или нарушитель записывает все передаваемые сообщения и впоследствии повторно передает их от имени пользователя законного.

Атака опробованием последовательным [sequential key search] — атака на криптосистему, основанная на методе последовательного опробования ключа.

Атака опробованием с использованием памяти [memory using attack, memory-used search attack] — атака на криптосистему, основанная на методе, существенно использующем память.

Атака пассивная [passive attack] — атака на криптосистему или протокол криптографический, при которой противник и/или нарушитель наблюдает и использует передаваемые сообщения шифрованные, но не влияет на действия пользователей законных.

Атака «противник в середине» [man-in-the-middle attack] — атака на протокол криптографический, в которой противник С выполняет этот протокол как с участником А, так и с участником В. Противник С выполняет сеанс с участником А от имени В, а с участником В — от имени А. В процессе выполнения противник пересылает сообщения от А к В и обратно, возможно подменяя их (отсюда название
атаки). В частности, в случае протокола аутентификации абонента успешное осуществление А.«п.в с.» позволяет противнику аутентифицировать себя для В под именем А. Для осуществления А.«п.в с.» необходимо обеспечивать синхронизацию двух сеансов протокола.

Атака протяжкой слова вероятного [moving probable word attack] — атака на криптосистему, основанная на методе протяжки слова вероятного.

Атака разностная [differential attack, син. атака дифференциальная] — атака на криптосистему, основанная на методе разностном.

Атака разностная на основе искажений [differential fault attack] — атака на криптосистему, основанная на методе искажений разностном.


Атака со словарем [dictionary attack] — атака на криптосистему, использующая словарь элементов текста открытого.

Атака со словарем паролей [password attack] — атака на криптосистему, основанная на переборе значений пароля.

Б

Биграмма [digram] — пара букв (символов), блоков текста.

Блок текста [text block] — мультигамма текста (текста открытого, текста шифрованного или промежуточного), составленная из подряд идущих знаков. Обычно текст разбивается на блоки одинаковой длины.

Блокнот одноразовый [one-time pad] — записанный на некотором материальном носителе (например, в специальных бумажных блокнотах) набор данных, используемых для получения последовательностей управляющих для однократного шифрования. Этот набор данных, обладающий определенными свойствами, должен обеспечивать стойкость (шифросистемы) совершенную при однократном применении.

Бумажник электронный [e-wallet, wallet] — специальное электронное устройство, предназначенное для решения проблемы повторной траты денег электронных. Б.э., выдаваемый клиенту банка, состоит из компьютера, которому доверяет клиент, и защищенного модуля, называемого наблюдателем, которому доверяет банк. Наблюдатель имеет возможность общаться с «внешним миром» только через компьютер клиента. Клиент не может потратить деньги цифровые без санкции наблюдателя; тем самым предотвращается повторная траата денег
электронных. Протоколы криптографические системы платежей электронных автономной с Б.э. обеспечивают неотслеживаемость действий клиентов, даже если и банк, и наблюдатель нечестные, и информация, накопленная банком, сопоставляется с содержимым наблюдателя.

В

Вектор инициализации [initialization vector] — вектор, который передается по каналу управления и используется для инициализации алгоритма шифрования. См. также Синхропосылка.

Вес функции булевой [weight of Boolean function] — число двоичных наборов в табличном задании булевой функции, на которых функция принимает значение «1».

Время жизни ключа [key life period, key life time] — временной интервал цикла жизненного ключа от генерации до уничтожения.

Г

Гамма (шифра) [keystream, ciphering sequence, key sequence] — последовательность управляющая знаков (блоков) алфавита, используемая в шифросистемах поточечных, реализующих гаммирование. Для обеспечения стойкости криптографической Г.(ш.) должна удовлетворять ряду требований, в частности быть близкой по своим свойствам к реализации последовательности случайной идеальной.

Гаммирование [running key ciphering, one-time padding] — шифрование, в котором функция зашифрования \( f(y, x) \) обратима по каждой переменной \( y \) обозначает знак (блок) гаммы шифра, \( x \) — знак (блок) текста открытого, значение функции \( f(y, x) \) — знак (блок) текста шифрованного). Важным частным случаем является так называемое модульное \( \Gamma \), когда \( f(y, x) = x + y(\text{mod} \, N) \), где \( N \) — размер числового алфавита \( \{0, 1, ..., N - 1\} \), из которого выбираются \( y \) и \( x \).

Генератор битов псевдослучайных криптографически сильный [cryptographically strong pseudorandom bit generator] — см. Генератор последовательностей псевдослучайных криптографически сильный.

Генератор ключей [key generator] — техническое устройство или программа, предназначенные для выработки массивов чисел или других данных, используемых в качестве ключей (криптосистемы), последовательности ключевой, векторов инициализации и т.п.

Генератор линейный конгруэнтный [linear congruent generator] — генератор, порождающий последовательность линейную конгруэнтную \( v(1), v(2), ... \)
Генератор подстановок псевдослучайных [pseudorandom permutation generator] — генератор функций псевдослучайных из семейства, в котором каждая функция является подстановкой.

Генератор последовательностей псевдослучайных [pseudorandom generator] — техническое устройство или программа для выработки последовательностей псевдослучайных.

Генератор последовательностей псевдослучайных квантовый [quantum pseudorandom generator] — генератор последовательностей псевдослучайных, основанный на использовании квантовых эффектов.

Генератор последовательностей псевдослучайных криптографически сильный [cryptographically strong pseudorandom bit generator, син. генератор битов псевдослучайных криптографически сильный] — математическая модель генератора последовательностей псевдослучайных, выходом которого являются последовательности псевдослучайные, неотличимые эффективно (с полиномиальной сложностью) статистическими тестами от последовательностей случайных идеальных.

Генератор с неравномерным движением [irregularly clocked generator, clock-controlled generator] — генератор, построенный на основе регистров сдвига, при этом выходные последовательности одних регистров используются для управления движением других регистров. Такой способ позволяет строить на основе линейных отображений, реализуемых регистрами сдвига, нелинейные преобразования множеств состояний генератора.

Генератор фильтрующий [filter generator] — генератор последовательности управляющей \( v(1), v(2), \ldots \), образованной с использованием последовательности линейной рекуррентной \( u(1), u(2), \ldots \) над кольцом \( R \) и функции упрощения \( f: R^n \rightarrow R \), называемой функцией фильтрующей, по следующему правилу: \( v(i) = f(u(i), \ldots, u(i + n - 1)) \), \( i = 1, 2 \).

Генератор функций псевдослучайных [pseudorandom function generator] — алгоритм, который псевдослучайным образом выбирает функцию из заданного семейства функций псевдослучайных.

Генератор функций с секретом [trapdoor function generator] — см. Функция с секретом.

Д

Деньги электронные [electronic money, e-money] — банковские платежные средства, представляющие собой записи (в так называемом электронном кошельке) о наличии реальных денежных средств, которыми обладает некоторое лицо. Реализованы в системах смарт-карт. Защита хранимого значения основывается на невозможности
создания фальшивой карты или осуществления операций с использо-
ванием чужой карты. Для защиты платежей применяются системы
криптографические, обеспечивающие конфиденциальность, целост-
ность, аутентификацию и невозможность отказа. В широком смысле
Д.э. — форма организации денежного обращения с использованием
компьютерных сетей.

Депонирование ключей [key escrow] — хранение копии ключа крип-
tosистемы у доверенного лица (организации, участника протокола
и т.п.) с целью восстановления работоспособности криптосистемы,
например, в случае утери ключа.

Дешифрование [decryption, breaking of cryptosystem] — процесс
аналитического раскрытия противником и/или нарушителем сообще-
ния открытого без предварительного полного знания всех элементов
системы криптографической. Если этот процесс поддается математи-
ческой формализации, говорят об алгоритме дешифрования.

Длина (размер) ключа [key length] — длина слова в определенном
алфавите, представляющего ключ. Длина ключа бинарного измеряется
в битах.

Длина покрытия группы [group cover length] — наименьшее $k$, для
которого конечная группа $G$ с системой образующих $M$ представима
в виде $G = M \cup M^2 \cup ... \cup M^k$.

Доказательство знания [proof of knowledge, син. протокол доказа-
tельства знания] — доказательство интерактивное, при котором
dоказывающий убеждает проверяющего в том, что он владеет секрете-
ной информацией, не раскрывая ее. Д.з. характеризуется двумя свой-
ствами: полнотой (протокола) и корректностью (протокола). К кате-
gorии Д.з. относятся протоколы идентификации.

Доказательство интерактивное [interactive proof] — понятие теории
сложности вычислений, составляющее основу понятия доказатель-
ства с разглашением нулевым. Д.и. — доказательство путем выпол-
нения протокола с двумя участниками, доказывающим и проверяю-
щим, в процессе работы которых участники обмениваются сообще-
шениями (запросы и ответы), обычно зависящими от случайных
чисел, которые могут содержаться в секрете. Цель доказывающего —
убедить проверяющего в истинности некоторого утверждения.
Проверяющий либо принимает, либо отвергает доказательство.
В отличие от обычного математического понятия доказательства
в данном случае доказательство носит не абсолютный, а вероят-
ностный характер и характеризуется двумя вероятностями. Если
dоказываемое утверждение верно, то доказательство должно быть
верным с вероятностью, стремящейся к единице при увеличении
числа циклов протокола. Если же доказываемое утверждение ложно, то при увеличении числа циклов протокола вероятность правильности доказательства должна стремиться к нулю. Криптографическое качество протокола Д.и. характеризуется свойствами полноты, корректности и разглашения нулевого.

Доказательство неинтерактивное с разглашением нулевым [noninteractive zero-knowledge proof] — доказательство с разглашением нулевым, выполняемое за один цикл (протокола): доказывающий посылает сообщение проверяющему, который на основе анализа этого сообщения либо принимает, либо отвергает доказательство.

Доказательство с разглашением минимальным [minimum-knowledge proof] — вид доказательства интерактивного, решающего задачу распознавания языка и удовлетворяющего требованиям к стойкости криптографической, которые аналогичны требованиям к стойкости доказательств с разглашением нулевым. В Д. с р.м. для данного фиксированного языка L общим входом доказывающего и проверяющего может быть произвольная строка x. Доказывающий должен определить, принадлежит ли эта строка языку L, и сообщить результат (x ∈ L или x ∉ L) проверяющему. При этом проверяющий, даже нечестный, не получает по завершении доказательства никакой дополнительной информации, за исключением значения предиката x ∈ L. Для противника, перехватывающего сообщения в процессе доказательства, случаи x ∈ L и x ∉ L могут быть неразличимы.

Доказательство с разглашением нулевым [zero-knowledge proof] — доказательство знания, которое обладает свойством разглашения нулевого.

Доказательство с разглашением нулевым совершенное [perfect zero-knowledge proof] — предельный случай доказательства с разглашением нулевым, в котором количество дополнительной информации, которую может получить проверяющий в результате выполнения протокола, равно нулю.

Доля секрета [share, secret share] — ключевая информация, получаемая отдельным участником схемы разделения секрета, позволяющая ему вместе с другими участниками правомочной коалиции восстановить значение секрета. См. также Структура доступа.

3

Задача логарифмирования дискретного [discrete logarithm problem] — задача нахождения логарифма дискретного в группе конечной. В последние десятилетия интерес к 3.л.д. существенно усилился в связи
с синтезом шифросистем асимметричных. Разработан ряд алгоритмов логарифмирования в мультипликативных группах конечных полей и других конечных группах.

Задача факторизации чисел целых [integer factoring problem] — задача разложения целого положительного числа в произведение простых чисел. З.ф.ч.ц. является классической математической задачей. В последние десятилетия интерес к ней существенно усилился в связи с синтезом шифросистем асимметричных. Разработан ряд алгоритмов факторизации целых чисел.

Зашифрование [encryption, enciphering] — процесс преобразования $f_{k}: x \rightarrow y$ сообщения открытого $x$ в сообщение шифрованное $y$ с помощью инъективной функции $D$, зависящей от ключа $k$ из множества ключевого (криптосистемы). При неизвестном ключе секретном для каждого открытого сообщения $x$ задача определения этого сообщения, или хотя бы близкого к нему (в некотором смысле), исходя из заданного множества $\{x = f_{k}^{-1}(y) | s \in K\}$, должна с заданной надежностью характеризоваться высокой ск-сложностью (теоретико-информационной, алгоритмической и вычислительной). См. также Алгоритм зашифрования, Стохастичность криптографическая.

И

Идентификатор ключа [key identifier] — указатель на ключ, представляющий собой системное имя ключа в программной реализации алгоритма криптографического и имеющий установленный в системе формат. Используется в качестве переменной при записи различных операций криптографических в тексте программы.

Идентификация [identification] — процедура установления присвоенного данной стороне уникального системного имени — идентификатора, которое позволяет отличать ее от других сторон. Обычно процедура И. заключается в предъявлении этого имени и предшествует процедуре аутентификации, т.е. подтверждению правильности идентификации. Термин И. часто для краткости используют для обозначения общей процедуры идентификации/аутентификации сторон (см., например, Протокол идентификации).

Идентификация пользователя (в системе информационной) [user identification in an information system] — присвоение пользователям идентификаторов и проверка вхождения предъявляемых идентификаторов в список присвоенных. Идентификации. Обязательно должна дополняться аутентификацией — проверкой принадлежности пользователю предъявленного им идентификатора.
Идентификация с разглашением нулевым [zero-knowledge identification] — вид доказательства с разглашением нулевым, целью которого является аутентификация сторон, т.е. доказательство одним из участников своей идентичности.

Имитация [imitation] — атака активная на протокол криптографический, целью которой является навязывание противником и/или нарушителем одной из сторон сообщения от имени другой стороны, которое не будет отвергнуто при приеме.

Имитовставка [message authentication code] — см. Код аутентичности сообщения.

Имитозащита [integrity protection, protection from imitation] — защита сообщений в системе связи от навязывания ложных данных.

Имитостойкость [imitation resistance] — свойство системы криптографической (протокола криптографического), характеризующее способность противостоять атакам активным со стороны противника и/или нарушителя, целью которых является навязывание ложного сообщения, подмена передаваемого сообщения или изменение хранимых данных.

К

Квадрат латинский [Latin square] — матрица размером $n \times n$, у которой каждая строка и каждый столбец являются перестановкой элементов некоторого конечного алфавита из $n$ элементов. В криптографии К.л. используются, например, для задания функции зашифрования при гаммировании.

Квадраты латинские ортогональные [orthogonal Latin squares] — два квадрата латинских размера $n \times n$, составленные из элементов множества $\{1, 2, ..., n\}$, при совмещении которых друг с другом получается таблица, содержащая в своих клетках все $n^2$ возможных упорядоченных пар чисел. В криптографии К.л.о. используются, например, для обеспечения имитостойкости.

Ключ «отбеливания» [whitening key] — бинарный подключ алгоритма зашифрования блочного базового, размер которого совпадает с размером блока текста и который используется в первом и последнем цикле шифрования. Обычно используется пара К.о., один из них поразрядно складывается по модулю 2 с блоком текста открытого, после чего эта сумма преобразуется в блок текста, сумма которого по модулю 2 с другим К.о. образует блок текста шифрованного.

Ключ (криптосистемы) [key (of a cryptosystem)] — изменяемый элемент (параметр), каждому значению которого однозначно соот-
всегда одно из отображений, реализуемыми криптосистемой. Все возможные значения ключа составляют множество ключевое криптосистемы. Ключи могут быть составными, т.е. содержать несколько частей, обеспечивающих различные функции криптосистемы. Например, при реализации алгоритма шифрования электронной схемой в качестве ключей могут использоваться начальные состояния элементов памяти схемы, функциональные узлы и др.

Ключ бинарный [binary key] — ключ, заданный вектором с двоичными координатами.

Ключ главный [master key] — элемент ключа составного, который используется для шифрования ключей шифрования ключей, предназначенных для шифрования ключей разовых или для генерации других видов ключей посредством шифрования определенных данных.

Ключ долговременный [long-term key] — элемент ключей составных, действующий в неизменном виде длительное время.

Ключ зашифрования [enciphering key] — ключ, используемый при зашифровании.

Ключ коммутиаторный [commutation key] — ключ, являющийся подстановкой степени \( n \) или бесповоротной выборкой размера \( m \) из \( n \) элементов, \( m < n \). Например, в шифре замены простой ключ представляет собой подстановку на множестве блоков текста.

Ключ открытый [public key] — неsekретный ключ шифросистемы асимметричной.

Ключ разовый [once-only key] — ключ, однократно используемый для шифрования в цикле жизненного ключей. Обычно не подлежит хранению и является элементом ключа составного.

Ключ расшифрования [decryption key] — ключ, используемый при расшифровании.

Ключ сеансовый [session key] — ключ, специально сгенерированный для одного сеанса связи между двумя участниками (протокола).

Ключ секретный [secret key] — ключ, сохраняемый в секрете от лиц, не имеющих допуска к ключам данной шифросистемы симметричной или к использованию некоторых функций данной шифросистемы асимметричной.

Ключ секретный квантовый [quantum secret key] — ключ секретный, полученный в ходе реализации распределения ключей квантового. Выделен как самостоятельный термин ввиду принципиальной важности задачи безопасного распределения секретных ключей среди участников защищенной сети; эта задача может быть удовлетворительно решена только при помощи канала связи, гарантированно защищенного от перехвата, например канала связи квантового.
Ключ скомпрометированный [compromised key] — ключ секретный, ставший доступным лицам, не имеющим допуска к ключам данной шифросистемы симметричной или к использованию некоторых функций данной шифросистемы асимметричной.

Ключ слабый [weak key] — ключ криптосистемы, при котором заметно ухудшаются характеристики стойкости криптографической криптосистемы по сравнению со средними значениями тех же характеристик при ключе, случайно равновероятно выбранном из множества ключевого криптосистемы.

Ключ составной [composite key] — см. Ключ (криптосистемы).

Ключ цикловой (раундовый) [round key] — набор, вычисляемый по ключу секретному алгоритма шифрования итеративного в процессе разветвления ключа. Используется для преобразования блока информации на одном из циклов (раундов) шифрования.

Ключ шифрования данных [data encryption key] — элемент ключа составного, предназначенный для шифрования данных.

Ключ шифрования ключей [key enciphering key (KEK)] — элемент ключа составного, используемый для шифрования ключей разовых.

Ключи эквивалентные [equivalent keys] — ключи, при которых криптосистема реализует одинаковые отображения.

Компрометация абонента [compromise of a party] — факт ознакомления противника и/или нарушителя с ключами секретными абонента защищенной сети связи (пользователя законного, участника (протокола)). Может иметь явный или тайный характер.

Конфиденциальность (информации) [privacy, confidentiality] — означает, что информация предназначена только определенному кругу лиц и должна храниться в тайне от всех остальных.

Конфиденциальность трафика [traffic (flaw) confidentiality] — свойство, характеризующее защищенность системы связи от получения противником и/или нарушителем информации о передаваемых в системе данных и/или функционировании системы в целом путем анализа трафика. Защита от анализа трафика обеспечивается путем сокрытия идентификаторов и адресов отправителя и получателя, длин пакетов, интенсивности передач и т.п.

Корректность (протокола) [soundness property] — способность протокола криптографического противостоять угрозам со стороны противника и/или нарушителя, не располагающего необходимой секретной информацией, но пытающегося выполнить протокол за участника, который по определению должен такой информацией владеть.

Корреляция функций [correlation of functions] — см. Коэффициент корреляции функций.
Криптоанализ [cryptanalysis] — см. Анализ криптографический.
Криптоанализ квантовый [quantum cryptanalysis] — анализ криптографический, основанный на применении алгоритмов квантовых вычислений.
Криптоаналитик [cryptanalyst] — специалист, занимающийся анализом криптографическим криптосистем.
Криптограмма [cryptogram] — сообщение шифрованное, оформленное по действующим правилам пользования системой шифрования. Содержит кроме текста шифрованного адрес, грифы срочности и другую служебную информацию.
Криптография [cryptography] — область научных, прикладных, инженерно-технических исследований и практической деятельности, которая связана с разработкой средств криптографических защит информации от угроз со стороны противника и/или нарушителя, а также анализом и обоснованием их стойкости криптографической. В настоящее время основными задачами К. являются обеспечение конфиденциальности, целостности, аутентификации, невозможности отказа, неотслеживаемости. В отличие от организационных и других способов защиты информации под криптографическими понимаются такие, которые используют математические методы преобразования защищаемой информации. К., с некоторой долей условности, делится на криптосинтез и криптоанализ; К. включает криптологию.
Криптография квантовая [quantum cryptography] — раздел криптографии, посвященный применению методов квантовой физики для синтеза и анализа систем криптографических.
Криптография компьютерная [computer cryptography] — общее название области криптографических исследований, связанной с применением криптографии для обеспечения компьютерной безопасности. Изучает особенности реализации систем криптографических в операционных системах, компьютерных сетях, системах управления базами данных и т.п.
Криптология (математическая криптография) [cryptology (mathematical cryptography)] — отрасль криптографии, математики и математической кибернетики, изучающая математические модели систем криптографических. Так же как и криптография, условно делится на две части: криптосинтез и криптоанализ.
Криптопротокол [cryptoprotocol] — см. Протокол криптографический.
Криптосистема [cryptosystem] — см. Система криптографическая.
Логарифм дискретный в группе конечной [discrete logarithm in a finite group] — минимальное целое положительное решение \( x \) уравнения \( a^x = b \), где \( a, b \) — элементы конечной группы \( G \). Наиболее часто в криптографии при анализе шифротекстов асимметричных рассматриваются логарифм дискретный в поле конечном и логарифм дискретный в группе точек кривой эллиптической.

Логарифм дискретный в поле конечном [discrete logarithm in a finite field] — логарифм дискретный в группе конечной в случае, когда группа является мультипликативной группой конечного поля.

Метод анализа криптографического (криптоанализа) [crypt-analytic method, method of cryptanalysis] — совокупность приемов и способов, направленных на исследование стойкости криптографической криптосистемы, объединенных одной или несколькими идеями (математическими, техническими или другими). Можно предположить, что и разработчик криптосистем, и противник и/или нарушитель используют одну и ту же совокупность М.а.к.(к.). В качестве наиболее важных характеристик М.а.к.(к.) обычно рассматривают трудоемкость М.а.к.(к.) и надежность М.а.к.(к.).

Механизм (средство) заполнения трафика [traffic padding] — средство заполнения пауз между передаваемыми сообщениями или их частями для сокрытия передаваемой информации в общем потоке передаваемых данных. Наиболее эффективным способом заполнения трафика является шифрование всего трафика, включая заполняющую паузы информацию.

Механизм криптографический [cryptographic mechanism] — термин, принятый в стандарте ISO 7498.2 для обозначения механизмов безопасности, использующих алгоритмы криптографические и протоколы криптографические. Устаревший синоним термина операція криптографическая, принятого в стандарте ISO/IEC 15408-99.

Механизм (средство) разграничения доступа [access control mechanism] — средство реализации идентификации, проверки полномочий пользователя и разрешения или отказа в доступе к объекту. Для этого могут использоваться различные средства: списки полномочий, системы идентификации, специальные режимы и особенности работы, метки, временные ограничения и выделенные маршруты. Наиболее надежно эти средства реализуются на основе системы управления ключами, дающими право доступа к соответствующей информации.

135
Многочлен периода максимального [maximal period polynomial] — см. Многочлен примитивный.

Многочлен примитивный [primitive polynomial, син. многочлен периода максимального] — неприводимый многочлен степени $m$ над полем $GF(q^m)$, порядок корней которого в поле разложения $GF(q^m)$ максимален и равен $q^m - 1$.

Многочлен разреженный [sparse polynomial] — многочлен, имеющий мало ненулевых коэффициентов по сравнению со своей степенью. Свойство разреженности многочлена находит применение в некоторых методах анализа криптографического.

Множество ключевое (криптосистемы) [key set (of a cryptosystem)] — множество всех возможных значений ключа криптосистемы.

Модель текста открытого [plain text model] — математическая модель, описывающая свойства реальных текстов открытых, вырабатываемых определенными источниками, либо естественными (осмысленный текст на каком-то языке), либо искусственными (межмашинный обмен, телеметрия и др.). Простейшими М.т.о. являются последовательность независимых испытаний и цепь Маркова. М.т.о. лежат в основе различных подходов к определению стойкости криптографической, а также методов анализа криптографического.

Модуль управляющий шифросистемы поточной [stream cipher control module] — часть шифросистемы поточной, генерирующая в зависимости от ключа криптосистемы последовательность ключевую (управляющую), определяющую в каждом такте выбор отображения для шифрования очередного блока текста.

Модуль шифрующий шифросистемы поточной [stream cipher ciphering module] — часть шифросистемы поточной, реализующая под воздействием последовательности ключевой (управляющей) шифрование очередного блока текста.

Монета электронная [e-coin] — название электронных платежных средств, используемых в системах платежей электронных автономных. Такая трактовка термина не является общепринятой. Многие авторы называют М.э. любое электронное платежное средство. См. также Деньги электронные, Деньги цифровые.

Мультигамма (m-грамм) [m-tuple] — набор из $m$ знаков алфавита. Обычно рассматривается случай $m \geq 2$.

Наблюдатель [observer] — термин, применяемый в системах платежей электронных и обозначающий защищенный модуль бумажника электронного, которому доверяет банк.
Набор ключей конфиденциальный [validator] — комплект, состоящий из ключа секретного схемы подписи цифровой, соответствующего ключа открытого и его сертификата ключа, используемый в системах платежей электронных автономных с бумажниками электронными. Н.к.к. выдается специальным органом (центром выдачи Н.к.к.), создаваемым для этих целей, вслепую так, что впоследствии центр выдачи Н.к.к. не сможет идентифицировать клиента, которому был выдан данный Н.к.к. Тем самым обеспечивается неотслеживаемость клиентов.

Набор тестов статистических [battery of tests] — в криптографии — совокупность статистических критериев (тестов), предназначенная для проверки соответствия анализируемой последовательности гипотезе о независимости и равномерности ее элементов. Каждый тест состоит в вычислении по анализируемой последовательности некоторой статистики, имеющей известное распределение для последовательности случайной идеальной, и использовании критерия согласия. Стандартными Н.т.с. являются набор тестов Д. Кнута, пакет DIEHARD (Дж. Марсаль), набор тестов NIST (Института стандартов США), пакет TestU01 (Л’Экуйера). В эти наборы входят тест автокорреляции, тест бита следующего, тест профиля сложности линейной, тест серий, тест универсальный Мауэрера, тест частотный и др.

Нарушитель [adversary, син. — участник нечестный, нарушитель внутренний] — участник протокола, нарушающий предписанные протоколом действия.

Нарушитель активный [active adversary] — нарушитель, который недопустимым образом влияет на ход выполнения протокола криптографического. Как правило, полный анализ всех результатов однократного выполнения криптографического протокола позволяет обнаружить присутствие Н.а.

Нарушитель внешний [outside adversary] — см. Противник. Рекомендуется использовать термин противник.

Нарушитель внутренний [inside adversary] — см. Нарушитель. Рекомендуется использовать термин нарушитель.

Нарушитель пассивный [passive adversary, eavesdropper] — нарушитель, который ограничивается сбором и анализом информации о ходе выполнения протокола криптографического, но не вмешивается в него. Полный анализ результатов неоднократного выполнения криптографического протокола не позволяет обнаружить присутствие Н.п.

Невозможность отказа [pop repudiation] — свойство протокола криптографического, состоящее в том, что его участники (все или
некоторые) не могут отказаться от факта совершения определенных действий. Обеспечивается системой подписи цифровой.

Нелинейность функции булевой [nonlinearity of Boolean function] — расстояние от функции булевой до класса функций аффинных.


Несвязываемость [unlinkability] — свойство, родственное неотслеживаемости и означающее, что противник и/или нарушитель не только не может установить, кто именно выполнил данное конкретное действие, но даже выяснить, были ли разные действия выполнены одним и тем же участником.

О

Обновление ключей [key updating] — способ смены ключей, при котором новый ключ генерируется с помощью вычисления значения функции (обычно функции одноразправленной) от аргумента, определяемого предыдущим ключом и, возможно, другими данными.

Операция криптографическая [cryptographic operation] — термин, принятый в криптографии компьютерной и введенный в стандарте ISO/IEC15408-99 для обозначения алгоритмов криптографических и протоколов криптографических. Под О.к. понимаются: зашифрование и расшифрование данных или ключей, алгоритм формирования подписи цифровой, алгоритм проверки подписи цифровой, вычисление кода аутентичности сообщения, вычисление значения хэш-функции, протокол выработки ключей и др. В более ранних стандартах использовался термин механизм криптографический.

II

Парадокс дней рождения [birthday paradox] — свойство выборки из $t$ независимых случайных знаков $n$-элементного алфавита, состоящее в том, что вероятность появления в выборке двух одинаковых знаков не меньше $1 - e^{-a}$, если $C_2 > a \cdot n > 0$. П.д.р. используется в криптографии для оценки различных характеристик криптосистем, например для оценки длины текста шифрованного, позволяющего составить уравнения относительно неизвестного ключа.
Пароль [password] — последовательность символов, задающая ключ или служащая для получения доступа к средствам криптографическим, вычислительным средствам и пр. Часто П. обладают лингвистическими особенностями, способствующими их запоминанию.

Перекрытие гаммы [repeated use of a key sequence] — полное или частичное повторное использование гаммы (последовательности управляющей) при зашифровании двух или более различных текстов открытых.

Подмена [substitution] — атака на криптосистему, состоящая в перехвате противником и/или нарушителем сообщения и замене его другим сообщением. При этом выбор последнего может зависеть от перехваченного сообщения.

Подпись цифровая (сообщения или документа) [digital signature] — представляет собой строку в некотором алфавите (например, цифровую), зависящую от сообщения или документа и от некоторого ключа секретного, известного только подписывающему субъекту. Предполагается, что П.ц. должна быть легко проверяемой без получения доступа к ключу секретному. При возникновении спорной ситуации, связанной с отказом подписывающего от факта п.ц. некоторого сообщения либо с попыткой подделки подписи, третьей сторона должна иметь возможность разрешить спор. Реализуется системой подписи цифровой. П.ц. позволяет решить следующие три задачи: осуществить аутентификацию источника данных, установить целостность сообщения или электронного документа, обеспечить невозможность отказа от факта подписи конкретного сообщения или документа.

Подпись электронная [electronic signature] — термин, применяемый в международных правовых актах для электронной идентификационной информации, добавляемой в качестве атрибута к электронному документу. Позволяет идентифицировать физических лиц путем сличения их подписей. Охватывает различные технологии, в том числе использующие биометрические характеристики, временные и физические характеристики процесса собственноручной подписи, подписи цифровые, ключи электронные, пластиковые карты и др. Основное назначение: идентификация пользователя (в системе информационной), подтверждение целостности подписываемого документа, а также обеспечение невозможности отказа от факта подписи. В настоящее время всем трем целям удовлетворяет только технология подписи цифровой.

Подпись электронная цифровая (ЭЦП) [electronic digital signature] — юридический термин, относящийся к технологии подписи цифровой
применительно к электронным документам. Является частным случаем подписи электронной. Основной проблемой является определение условий, при которых П.э.ц. в электронном документе юридически равнозначна собственноручной подписи в документе на бумажном носителе. Например, в соответствии с Федеральным законом Российской Федерации от 10 января 2002 г. № 1-ФЗ «Об электронной цифровой подписи» «электронная цифровая подпись в электронном документе равнозначна собственноручной подписи в документе на бумажном носителе при одновременном соблюдении следующих условий: сертификат ключа подписи, относящийся к этой электронной цифровой подписи, не утратил силу (действует) на момент проверки или на момент подписания электронного документа при наличии доказательств, определяющих момент подписания; подтверждена подлинность электронной цифровой подписи в электронном документе; электронная цифровая подпись используется в соответствии со сведениями, указанными в сертификате ключа подписи».

Подстановка односторонняя [one-way permutation] — взаимно-однозначная функция односторонняя.

Полнота (протокола) [completeness property] — свойство протокола криптографического, означающее, что при выполнении участниками честными протокол решает ту задачу, для которой он создан.

Пользователь законный [legal user] — официально зарегистрированный в системе пользователь.

Помехоустойчивость шифра [noise stability of a cipher] — способность шифра противостоять действию случайных помех (в отличие от целенаправленных действий противника), возникающих при передаче сообщения шифрованного по каналу связи.

Последовательность истинно случайная [true random sequence] — последовательность, порожденная недетерминированным физическим устройством или процессом. Такая последовательность (в отличие от последовательности псевдослучайной) непредсказуема и не воспроизводима. Статистические свойства П.и.с. могут отличаться от статистических свойств последовательности случайной идеальной.

Последовательность ключевая [key stream] — в шифросистемах поточных — последовательность управляющая, однозначно определяющая в каждом тактом выбор функции зашифрования для зашифрования очередного знака текста открытого. Иногда термин П.к. используется в качестве синонима гами в шифре гаммирования.

Последовательность псевдослучайная [pseudo-random sequence] — последовательность, порожденная детерминированным устройством или программой. Важной задачей криптографии является построение
П. п., обладающих свойствами, близкими к свойствам типичных реализаций последовательности случайной идеальной. См. также Генератор последовательностей псевдослучайных криптографически сильный.

Последовательность псевдослучайная криптографически сильная [cryptographically strong pseudorandom sequence] — последовательность псевдослучайная, вырабатываемая генератором последовательностей псевдослучайных криптографически сильным.

Последовательность рекуррентная [recurrent sequence] — последовательность, в которой каждый элемент однозначно определяется некоторым фиксированным числом ее предыдущих элементов с помощью функции, именуемой законом рекурсии.

Последовательность сбалансированная [balanced sequence] — последовательность знаков конечного алфавита $X$, в которой все элементы из $X$ встречаются одинаковое число раз.

Последовательность случайная идеальная [ideal random sequence] — последовательность, являющаяся реализацией последовательности независимых случайных величин, имеющих равномерное распределение на заданном конечном алфавите.

Последовательность управляющая [control sequence] — последовательность псевдослучайная или последовательность истинно случайная, используемая при реализации алгоритма криптографического. Частными случаями являются гамма шифра и последовательность ключевая.

Постулаты Голомба [Golomb postulates] — сформулированные С. Голомбом постулаты для последовательностей псевдослучайных двоичных, используемых в криптографических приложениях. Согласно им последовательность должна удовлетворять определенным ограничениям на встречаемость знаков, мультиграмм и функцию автокорреляционную последовательности. Последовательности, удовлетворяющие П.Г., иногда называют псевдослучайными.

Правило Керкгоффса [Kerckhoffs assumption] — общепринятое в криптографии предположение проведения криптоанализа, впервые сформулированное голландским криптографом Н. Керкгоффсом («компрометация системы не должна причинять неудобств корреспондентам»). В современном понимании это правило означает, что описание криптосистемы (криптотротокола) может быть полностью известно противнику и/или нарушителю, а стойкость криптографическая основана только на том, что неизвестен ключ (секретный).

Предположение криптографическое (криптологическое) [cryptographic assumption] — предположение о вычислительной сложности
какой-либо математической задачи, на основе которого доказывается стойкость теоретико-сложностная криптосистем и протоколов криптографических. Примерами П.к.(к.) являются предположения о существовании функций односторонних, функций с секретом, о вычислительной сложности задачи логарифмирования дискретного, задачи факторизации чисел целых.

Предположение о ящике черном [black box assumption] — предположение криптоанализа, означающее, что алгоритм шифрования неизвестен и возможно лишь наблюдение выхода алгоритма при любом заданном входе. См. также Ящик черный.

Предположения криптоанализа [cryptanalytic assumptions] — совокупность условий и допущений, при которых проводится анализ системы криптографической. Фактически предположения разработчика и пользователя законного описывают модель противника и/или нарушителя, т.е. его цели, возможности и имеющиеся исходные данные. Предположения противника и/или нарушителя обычно описывают свойства криптосистемы и особенноности ее реализации и применения.

Примитив криптографический [cryptographic primitive] — функция (семейство функций), которая используется как составной элемент при построении криптосистем (протоколов криптографических) и обладает определенным криптографическим свойством. Примеры П.к.: функция односторонняя, хэш-функция, генератор последовательностей псевдослучайных, семейство функций псевдослучайных. Иногда П.к. называют такие объекты, как подпись цифровая, деньги электронные, сертификат ключа и т.п., если они используются при построении протокола криптографического.

Противник [adversary, син. — нарушитель внешний] — внешний по отношению к участникам криптопротокола (системы криптографической) субъект (или коалиция субъектов), наблюдающий за передаваемыми сообщениями и, возможно, вмешивающийся в работу участников путем перехвата, искажения (модификации), вставки (создания новых), повтора и перенаправления сообщений, блокирования передачи и т.п. с целью нарушения одной или нескольких функций-сервисов безопасности. Может образовывать коалицию с нарушителем.

Противник активный [active adversary] — противник, который вмешивается в ход выполнения протокола криптографического или работу системы криптографической. Как правило, полный анализ всех результатов однократного выполнения криптографического протокола позволяет обнаружить присутствие П.а.
Противник пассивный [passive adversary, eavesdropper] — противник, который может получать некоторую информацию о выполнении протокола криптографического или работы системы криптографической, но не вмешивается в их работу. В случае протоколов полный анализ результатов неоднократного их выполнения не позволяет обнаружить присутствие П.п.

Протокол [protocol] — описание распределенного алгоритма, в процессе выполнения которого два участника (или более) последовательно выполняют определенные действия и обмениваются сообщениями. Последовательность шагов протокола группируется в циклы (роуны).

Протокол аутентификации абонентов [user authentication protocol] — см. Аутентификация абонента (пользователя), протокол идентификации.

Протокол аутентификации сообщений [message authentication protocol] — протокол криптографический, предназначенный для обеспечения целостности сообщений, под которой понимается гарантируемая получателю возможность удостовериться, что сообщение поступило от заявленного отправителя и в неискаженном виде. В случае когда участники протокола доверяют друг другу и защищаются противника, П.а.с. строятся на основе кодов аутентичности сообщений. Если же участники друг другу не доверяют, то для аутентификации сообщений используется схема подписи цифровой.

Протокол голосования [election scheme, voting scheme, voting protocol] — протокол криптографический прикладной, позволяющий проводить процедуру голосования, в которой избирательные бюллетени существуют только в электронной форме. Является протоколом криптографическим, так как обеспечивает тайный характер голосования. Основное свойство П.г. — универсальная проверяемость, т.е. предоставление возможности всякому желающему, включая сторонних наблюдателей, в любой момент времени проверить правильность подсчета голосов.

Протокол групповой [group-oriented protocol] — протокол криптографический, в котором какой-либо алгоритм, требующий знания ключа секретного, является распределенным. Например, в протоколе подписи групповой подписывающий заменяется группой участников таким образом, что корректная подпись может быть сформирована только при участии всех членов группы.

Протокол двусторонний [two-party protocol] — протокол с двумя участниками.

Протокол идентификации [identification protocol, син. — схема идентификации] — протокол аутентификации сторон, участвующих во взаимодействии и не доверяющих друг другу. Различают П.и.
с аутентификацией односторонней или аутентификацией взаимной. П.и., как правило, основаны на известной обеим сторонам информации (пароли, личные идентификационные номера, ключи секретные или ключи открытые) и реализуются с использованием техники «запрос—ответ» или доказательства знания. В дополнение к П.и. могут использоваться некоторые физические приборы, с помощью которых проводится идентификация (магнитная или интеллектуальная пластиковая карта, прибор, генерирующий меняющиеся со временем пароли), а также биометрические параметры.

Протокол идентификации взаимной [mutual identification protocol] — см. Аутентификация взаимная, протокол идентификации.

Протокол идентификации односторонней [one-way authentication protocol] — см. Аутентификация односторонняя, протокол идентификации.

Протокол интерактивный [interactive protocol] — протокол, выполняемый за два цикла (раунда) или более.

Протокол криптографический [cryptographic protocol, син. — криптопротокол] — протокол, предназначенный для выполнения функций системы криптографической, в процессе выполнения которого участники используют алгоритмы криптографические.

Протокол криптографический квантовый [quantum cryptographic protocol] — протокол криптографический, использующий канал связи квантовый.

Протокол криптографический прикладной [application cryptographic protocol] — протокол криптографический, предназначенный для решения практических задач обеспечения функций-сервисов безопасности с помощью систем криптографических. Примеры: протокол конфиденциальной передачи сообщений, схема подписи цифровой, система платежей электронных, протокол голосования, протокол подписания контракта и др.

Протокол криптографический примитивный [primitive cryptographic protocol] — протокол криптографический, который не имеет самостоятельного прикладного значения, но используется как базовый компонент при построении протоколов криптографических прикладных. Как правило, П.к.п. решает какую-либо одну абстрактную задачу. Примеры: протокол обмена секретами, протокол привязки к биту, протокол подбрасывания монеты (по телефону).

Протокол (алгоритм) обмена ключами Диффи—Хеллмана [Diffie—Hellman algorithm] — один из первых протоколов распределения ключей открытого. Предназначен для формирования «общего секрета» (ключа, идентификационного номера и др.) сторонами, обменива-
ющими данными по открытому каналу с использованием ключей открытых (общедоступных) и ключей секретных (индивидуальных). Вместе с алгоритмом шифрования RSA положил начало развитию шифрсистем асимметричных.

Протокол обмена секретами [secret exchange protocol] — протокол криптографический примитивный с двумя участниками. Входные слова участников называются секретами. В протоколе обмен секретами организован таким образом, чтобы в случае его прерывания (по любой причине) знания участников о секретах друг друга были приблизительно одинаковыми.

Протокол подбрасывания монеты (по телефону) [coin flipping (by telephone) protocol] — протокол криптографический примитивный, позволяющий двум не доверяющим друг другу участникам сгенерировать общий случайный равномерно распределенный бит. Главное свойство таких протоколов состоит в том, что если хотя бы один из участников является участником честным, то сгенерированный бит будет случайным независимо от действий другого участника. Имеются обобщения на случай конечных битовых строк, а также на случай произвольного количества участников.

Протокол подписания контракта [contract signing protocol] — протокол криптографический прикладной, как правило с двумя участниками, которые, обмениваясь сообщениями по каналам связи, должны подписать контракт, существующий только в электронной форме. Основное требование к стойкости криптографической П.п.к. таково: при любом прерывании выполнения протокола шансы каждого из участников получить контракт, подписанный другим, и при этом не подписаться самому ничтожно малы. Поэтому П.п.к. должен включать в себя протокол обмена секретами. Имеются и другие требования к стойкости протокола, в частности так называемая защита от злоупотреблений (abuse). Последняя означает, что если выполнение протокола было прервано и контракт остался неподписанным, то ни один из участников не сможет доказать третьим лицам (арбитрам), что другой участвовал в выполнении протокола (а следовательно, имел намерение подписать данный контракт).

Протокол подписи групповой [group signature protocol] — описание алгоритма формирования подписи цифровой, предполагающего одновременное участие заранее определенной группы участников. В случае отсутствия хотя бы одного участника из группы формирование подписи невозможно.

Протокол (схема) привязки к биту [bit commitment protocol (scheme)] — протокол криптографический примитивный с двумя
участниками (отправителем и получателем), посредством которого отправитель передает получателю бит информации (битовое обязательство) таким образом, что выполняются следующие два условия: 1) после передачи бита получателю (так называемого этапа привязки) отправитель уже не может изменить его значение; 2) получатель не может самостоятельно определить значение бита и узнает его только после выполнения отправителем так называемого этапа раскрытия.

Протокол разделения секрета [secret sharing protocol] — протокол криптографический, реализующий схему разделения секрета в модели, где участники являются абонентами сети связи. В этой модели имеется дополнительный участник (дилер), которому известно значение секрета. Дилер генерирует долю секрета и рассылает их остальным участникам. Всякая правомочная коалиция участников может восстановить секрет, выполнив протокол восстановления секрета. П.р.с. могут найти применение в организации хранения конфиденциальной информации, например ключей криптосистемы, а также как протоколы криптографические примитивные. См. также Структура доступа.

Протокол разделения секрета проверяемого [verifiable secret sharing protocol] — протокол разделения секрета, предназначенный для случая, когда участники не доверяют друг другу, в том числе и дилеру. Для защиты от нечестного дилера П.р.с.п. предоставляет каждому из остальных участников возможность проверить, что от дилера получена корректная доля секрета.

Протокол распределения ключей [key distribution protocol] — протокол получения пользователями ключей, необходимых для функционирования системы криптографической. Различают следующие типы П.р.к.: протоколы передачи (уже сгенерированных) ключей; протоколы совместной выработки общего ключа (распределение ключей открытого); схемы распределения ключей предварительного.

Протокол с арбитром [arbitrated protocol, син. — протокол с посредником] — протокол криптографический, в котором для разрешения споров между участниками требуется арбитраж. П.с. а. делятся на два класса. В пессимистических протоколах арбитр должен участвовать в каждом сеансе выполнения протокола. В оптимистических протоколах участие арбитра требуется только в случае возникновения конфликтов между участниками.

Протокол с посредником [arbitrated protocol] — см. Протокол с арбитром.
Развертывание ключа [key scheduling] — в шифросистемах поточ-
ных — выработка, последовательности ключевой по короткому ключу. 
В шифросистемах блочных — алгоритм вычисления ключей цикловых 
(раундовых) по ключу разовому.

Разглашение нулевое [zero-knowledge property] — свойство прото-
кола доказательства знания, обеспечивающее такое его выполнение, 
что никакая информация о доказываемом утверждении, кроме факта 
его истинности, не может быть получена честным проверяющим 
из переданных сообщений за время, полиномиально зависящее 
от суммарной длины этих сообщений.

Разглашение нулевое относительно проверяющего честного [honest-
verifier zero-knowledge] — ослабленный вариант разглашения нулевого, 
при котором требуется, чтобы протокол доказательства интерактив-
ного не давал никакой дополнительной информации о доказываемом 
утверждении лишь честному проверяющему, т.е. выполняющему 
действия, предписанные протоколом. С криптографической точки 
зрения данное свойство защищает доказывающего не от честного 
проверяющего, а от противника, который подслушивает сеанс вы-
полнения протокола.

Разграничение доступа [access control] — см. Функция-сервис раз-
граничения доступа, система разграничения доступа.

Разделение секрета [secret sharing] — см. Схема разделения секрета, 
протокол разделения секрета.

Разделение секрета проверяемое [verifiable secret sharing] — см. Про-
токол разделения секрета проверяемого.

Распределение ключей квантовое [quantum key distribution] — про-
цедура распределения ключей секретных, реализуемая с помощью 
протоколов криптографических квантовых и каналов связи квантовых.

Распределение ключей открытые [public key distribution] — протокол 
совместной выработки пользователями (общего) ключа секретного 
путем обмена сообщениями по открытому каналу связи. Протокол 
должен исключать возможность получения информации о ключе 
посторонними, а также любым участником до завершения им дей-
ствий, предусмотренных протоколом.

Режим выработки имитовставки [message authentication code 
mode] — см. Режим выработки кода аутентичности сообщения.

Режим выработки кода аутентичности сообщения [message authentica-
tion code mode, син. — режим выработки имитовставки] — режим шиф-
рования, применяемый для выработки кода аутентичности сообщения.
Режим шифрования [encryption mode] — способ получения алгоритма зашифрования, исходя из алгоритма зашифрования блочного базового. Основными Р.ш. являются: простая замена или электронная кодовая книга (ECB), сцепление блоков шифротекста (CBC), обратная связь по шифротексту (CFB), обратная связь по выходу (OFB). Выбор Р.ш. имеет целью обеспечение определенных свойств алгоритма шифрования (ограничение распространения искажений, простота синхронизации и др.).

С

Свойство перемешивания [mixing property] — строго неформализуемое свойство функции зашифрования, выражающееся, в современном понимании, в существенном усложнении взаимосвязи статистических и аналитических характеристик элементов текста шифрованного по сравнению с подобными взаимосвязями элементов текста открытого. Термин «перемешивание» перенесен в криптографию К. Шенноном из теории вероятностей.

Свойство рассеивания [diffusion property] — строго неформализуемое свойство функции зашифрования, состоящее в том, что каждый знак текста открытого влияет на большое число знаков текста шифрованного. Термин введен К. Шенноном.

Свойство усложнения [confusion property] — строго неформализуемое свойство функции зашифрования, означающее сложную зависимость между ключом и текстом шифрованным. Термин введен К. Шенноном.

Семейство подстановок псевдослучайных [pseudorandom permutation family] — семейство функций псевдослучайных, в котором каждая функция является подстановкой.

Семейство подстановок с секретом [trapdoor permutation family] — семейство функций с секретом, в котором каждая функция является подстановкой.

Семейство функций псевдослучайных [pseudorandom function family] — семейство функций, обладающее следующим свойством. Функция, выбранная случайно равномерно от семейства, алгоритмически неотличима от случайной функции. При этом каждая из функций, и выбранная из С.ф.п., и случайная функция, рассматривается как ящик черный, т.е. алгоритм может только получать значения функции на выбираемых им значениях аргумента.

Сертификат ключа [key certificate] — структура данных заранее определенного формата, включающая ключ открытый, идентифи-
кационную информацию владельца ключа, а также другую служебную информацию (время действия и предназначение ключа, тип используемых алгоритмов криптографических и др.), заверенная подписью цифровой уполномоченного лица доверенного центра сертификации (центра удостоверяющего).

Синтез криптографический [cryptosynthesis, син. — криптосинтез] — условно выделяемая часть криптографии (криптологии), связанная с разработкой систем криптографических (протоколов криптографических).

Синхропосылка [synchronsignal] — комбинация знаков, передаваемая по каналу связи и предназначенная для вхождения в связь аппаратуры шифрования или для синхронизации аппаратуры. См. также Вектор инициализации.

Система идентификации [identification system] — система криптографическая, выполняющая аутентификацию сторон в процессе информационного взаимодействия. Математическая модель С.и. включает протокол идентификации и систему ключевую.

Система имитозащиты [integrity system] — система криптографическая, выполняющая аутентификацию сообщений и предназначенная для защиты от несанкционированного изменения информации или навязывания ложной информации. В частности, С.и. обеспечивает целостность информации. Математическая модель С.и. включает алгоритм кодирования имитозащищающего (это может быть алгоритм шифрования, код аутентификации, либо другое преобразование) и алгоритм принятия решения об истинности полученной информации, а также систему ключевую.

Система ключевая [key system] — состоит из множества ключевого (криптосистемы) и двух подсистем: системы установки ключей и системы управления ключами.

Система ключевая шифросистемы асимметричной [key system of a public key cryptosystem] — система ключевая, основанная на использовании каждым участником пары, состоящей из ключа открытого и ключа секретного. Такие системы облегчают реализацию многих ключевых протоколов по сравнению с шифросистемами симметричными. Вместе с тем для обеспечения взаимного доверия между пользователями требуется дополнительная система, называемая инфраструктурой ключей открытых.

Система ключевая шифросистемы симметричной [key system of a secret key cryptosystem] — система ключевая, основанная на использовании только ключей секретных, что исключает необходимость механизмов доверия, свойственных шифросистемам асимметричным,
но, вместе с тем, затрудняет реализацию ряда протоколов цикла жизненного ключей (например, распределения ключей).

Система криптографическая [cryptographic system (cryptosystem), син. — криптосистема] — система обеспечения безопасности информации криптографическими методами в части конфиденциальности, целостности, аутентификации, невозможности отказа и неотслеживаемости. В качестве подсистем может включать системы шифрования, системы идентификации, системы имитозащиты, системы подписи цифровой и др., а также систему ключевую, обеспечивающую работу остальных систем. В основе выбора и построения С. к. лежит условие обеспечения стойкости криптографической.

Система криптографическая квантовая [quantum cryptographic system] — система криптографическая, использующая каналы связи квантовые.

Система криптографическая пороговая [threshold cryptographic scheme] — система криптографическая, в которой ключ секретный распределен между n участниками так, что для функционирования системы необходима и достаточна совместная работа любых t участников, где t < n — заданное число.

Система обмена данными электронная [electronic data interchange (EDI)] — автоматизированная технология осуществления сбора, классификации, хранения, поиска, обработки и передачи информации, основанная на едином представлении и структурировании данных. В настоящее время интегрируется с технологиями автоматизации делопроизводства и документооборота. Для защиты С.о.д.э. применяются системы криптографические, обеспечивающие конфиденциальность, целостность, аутентификацию и невозможность отказа.

Система подписи цифровой [digital signature cryptosystem] — система криптографическая, выполняющая аутентификацию источника данных или аутентификацию сообщения. Предназначена для защиты от отказа субъектов от некоторых из ранее совершенных ими действий. Например, отправитель может отказаться от факта передачи сообщения, утверждая, что его создал сам получатель, а получатель может модифицировать, подменить или создать новое сообщение, а затем утверждать, что оно получено от отправителя. Математическая модель С.п.ц. включает схему подписи цифровой и систему ключевую, в качестве которой обычно выступает инфраструктура ключей открытых. Для разрешения споров необходима процедура арбитража, с помощью которой третья сторона — арбитр — разрешает споры о подлинности подписи цифровой.

150
Система разграничения доступа [access control mechanism] — обозначение или программное обеспечение, процедуры автоматизированной системы, процедуры администратора и их различные комбинации, которые обнаруживают, предотвращают несанкционированный доступ и разрешают законный доступ в автоматизированных системах.

Система управления ключами [key management system] — подсистема системы ключевой, определяющая порядок регистрации ключей, их использования, смены, хранения и архивирования, резервного копирования и восстановления, замены или изъятия из обращения скомпрометированных, а также уничтожения старых ключей. Целью управления ключами является нейтрализация таких угроз, как компрометация и несанкционированное использование ключей, например использование ключа, срок действия которого истек.

Система шифрования [cryptosystem, cipher, син. — шифросистема] — система криптографическая, предназначенная для защиты информации от лиц, не имеющих права доступа к ней. Защита обеспечивается путем зашифрования информации. Математическая модель С.ш. включает способ кодирования исходной и выходной информации, шифр и систему ключевую. Основными требованиями, определяющими качество С.ш., являются: стойкость криптографическая, имитостойкость, помехоустойчивость шифра и др.

Сообщение открытое [plaintext, cleartext] — в широком смысле — данные, представленные в виде последовательности над конечным множеством (буквы, цифры и другие символы) или непрерывного сигнала (звуки, изображения и др.), подлежащие зашифрованию. В более узком смысле — аналогичные данные, обладающие доступным семантическим (смысловым) содержанием и предназначенные для хранения, преобразования или передачи.

Сообщение шифрованное [ciphertext] — сообщение, полученное в результате зашифрования сообщения открытого.

Способ шифрования [encryption method (cipher type)] — способ преобразования множества сообщений открытых в множество сообщений шифрованных и обратно. Основные известные С.ш. реализуются шифром гаммирования, шифром замены простой и шифром перестановки, а также их комбинациями.

Средства криптографические [cryptographic tools, cryptographic mechanisms] — в широком смысле — средства обеспечения безопасности информации, использующие функции криптографические. В узком смысле — средства, реализованные в виде документов, механических, электромеханических, электронных технических
устройств или программ, предназначенные для выполнения функций системы криптографической.

Средства криптографические аппаратные [cryptographic hardware (device, facility)] — средства криптографическая, реализованные в виде специальных технических устройств. Реализуют одну или несколько функций криптографических или их частей.

Стойкость доказуемая [provable security] — см. Стоякость криптографическая теоретическая.

Стойкость криптографическая [cryptographic security] — фундаментальное понятие криптографии — свойство криптосистемы (криптопротокола), характеризующее ее (его) способность противостоять атакам противника и/или нарушителя, как правило имеющих целью получить ключ секретный или сообщение открытого. Развиваются два основных подхода к определению и оценке стойкости — стойкость теоретическая и стойкость практическая.

Стойкость (криптосистемы) практическая [practical security (of the cryptosystem)] — вычислительная сложность алгоритма, реализующего наилучшую в определенном смысле атаку на криптосистему. Чаще всего под С.п. понимают временную сложность выполнения успешной атаки на криптосистему наиболее быстрым из известных алгоритмов при реальных предположениях о свойствах криптосистемы и ее применении, а также о вычислительных машинах, на которых она будет реализовываться. Такой подход с учетом перспектив развития вычислительных машин позволяет оценить время, в течение которого данная криптосистема будет обеспечивать защитенность информации. См. также Стоякость криптографическая.

Стойкость (шифросистемы) совершенная [perfect secrecy] — свойство системы шифрования, заключающееся в том, что текст шифрованный не содержит информации о ключе и тексте открытом, кроме, возможно, его длины. Например, таким свойством обладает шифросистема гаммирования, если применяемая гамма является реализацией последовательности случайной идеальной.

Стойкость (криптосистемы) теоретическая [theoretical security] — стойкость криптографическая, определяемая в рамках некоторой математической модели. Основные подходы к определению С.т. в настоящее время — стойкость теоретико-информационная и стойкость теоретико-сложностная. Рассмотрение С.т. в рамках абстрактных математических моделей позволяет говорить о стойкости доказуемой.

Стойкость теоретико-информационная (шенноновская) [information-theoretic (Shannon) security] — вид стойкости теоретической,
определяемый с точки зрения математической теории информации. С.т.-и. криптосистемы обычно характеризуется количеством информации (в смысле К. Шеннона) относительно неизвестного противнику и/или нарушителю элемента криптосистемы, содержащимся в перехваченном тексте шифрованном или других доступных данных и вычисленным в рамках той или иной вероятностной модели. Говорят также, что С.т.-и. криптосистемы характеризует ее способность противостоять атакам со стороны противника и/или нарушителя, располагающих неограниченными вычислительными ресурсами.


Схема подписи цифровой [digital signature scheme] — состоит из двух алгоритмов: алгоритма формирования подписи цифровой и алгоритма проверки подписи цифровой. Надежность С.п.ц. определяется сложностью решения следующих трех задач для лица, не являющегося владельцем ключа секретного: подделки подписи цифровой, т.е. вычисления значения подписи под заданным документом; создания подписанного сообщения, т.е. нахождения хотя бы одного сообщения с правильным значением подписи (подделка подписи цифровой экзистенциальная); подмены сообщения, т.е. подбора двух различных сообщений с одинаковыми значениями подписи.

Т

Текст открытый [plaintext] — сообщение открытое, представленное в виде последовательности над конечным алфавитом.

Текст шифрованный (шифротекст) [ciphertext] — текст, полученный в результате зашифрования текста открытого.
Угроза [threat] — потенциальная опасность нарушения одного или нескольких свойств системы криптографической (протокола криптографического), например конфиденциальности, целостности, аутентификации, невозможности отказа, неотслеживаемости.

Угроза активная [active threat] — угроза, которая может быть реализована путем намеренного несанкционированного вмешательства в работу криптосистемы (протокола криптографического).

Усложнение последовательности линейной рекуррентной [linear recurrent sequence confusion] — преобразование последовательности линейной рекуррентной с целью усложнения ее аналитического строения. Одним из распространенных способов У.п.л.р. является ее преобразование с помощью генератора фильтрующего.

Устройство криптографическое [cryptographic device] — средство криптографическое аппаратное, выполненное в виде отдельного устройства.

Участник (протокола) [party] — субъект, участвующий в той или иной форме в выполнении протокола.

Участник нечестный [dishonest party] — см. Нарушитель.

Участник честный [honest party] — участник протокола криптографического, владеющий всей необходимой информацией, в том числе, если требуется, ключами секретными, и выполняющий действия в соответствии с протоколом.

Ф


Функция зашифрования [encryption function] — функция зашифрования описывает процесс зашифрования и осуществляет зависящее от ключа отображение последовательностей блоков текста (сообщения) открытого в последовательности блоков текста (сообщения) шифрованного. Доопределяется на множество всех текстов (сообщений) открытых и реализуется алгоритмом зашифрования.

Функция криптографическая [cryptographic function] — функция, необходимая для реализации системы криптографической. К таким
функциям относятся: генерация ключей, генерация последовательностей псеvдослучайных, функция шифрования, вычисление и проверка значений кода аутентичности сообщения и подписи цифровой, вычисление значения хэш-функции и др.

Функция одномоправленная [one-way function] — см. Функция односторонняя.

Функция односторонняя [one-way function, син. — функция одномоправленная] — отображение множества всех слов конечной длины \( n \) над конечным алфавитом, для которого существует такое \( \gamma < \infty \), что образ любого слова длины \( n \) можно вычислить за \( O(n^\gamma) \) операций, но ни для какого \( \beta < \infty \) не существует алгоритма, вычисляющего для любого слова длины \( n \) его прообраз за \( O(n^\beta) \) операций. Понятие Ф.о. используется в основном при конкретизации определения стойкости криптографической шифросистеме асимметричных. Вопрос о существовании Ф.о. является открытым. Доказательство односторонности какой-либо функции означало бы доказательство гипотезы \( NP \neq P \) в теории сложности алгоритмов. Различают Ф.о. сильные, Ф.о. слабые, функции дистрибутивно односторонние, подлинные и др. Иногда Ф.о. называют эффективно вычислимую функцию, для которой неизвестен эффективный алгоритм обращения.

Функция равновероятная [balanced function] — см. Функция сбалансированная.

Функция расшифрования [decryption function] — функция, описывающая процесс расшифрования и осуществляющая отображение, обратное к функции зашифрования. Доопределяется на множество всех текстов (сообщений) шифрованных и реализуется алгоритмом расшифрования.

Функция с запретами [function with interdictions] — функция дискретная, для которой существует запрет функции. Выходная последовательность генератора фильтрующего, построенного с помощью Ф.с. з., является последовательностью пseвдослучайной, в которой отсутствуют некоторые многограммы символов, поэтому она может быть отображена набором тестов статистических.

Функция с секретом [trapdoor function] — функция дискретная, зависящая от параметра (секрета, описание секрета). Знание параметра позволяет эффективно (с полиномиальной сложностью) вычислять и инвертировать данную функцию. Если параметр неизвестен, то не существует эффективного алгоритма инвертирования функции. Семейство Ф. с с. обладает свойствами функции односторонней. Применение Ф. с с. предполагает построение соответствующего генератора функций с секретом, т.е. эффективного алгоритма,
порождающего пары (функция, секрет). Например, см. Шифросистема асимметричная.

Функция сбалансированная [balanced function, син. — функция равномерная] — отображение сбалансированное, у которого значение параметра \( m \) равно единице.

Функция сжатия [compression function] — функция, отображающая входные данные, состоящие из текущего хэш-значения и очередного блока хэшируемого сообщения, в новое хэш-значение. Используется в интерактивных конструкциях хэш-функций, позволяющих хэшировать сообщения произвольной длины.

Хэш-значение [hash-code, hash-result, hash-value, hash, imprint, digital fingerprint, message digest]—значение хэш-функции для данного аргумента.

Хэш-функция [hash function] — функция, отображающая входное слово конечной длины в конечном алфавите в слово заданной, обычно фиксированной длины.

Хэш-функция криптографическая [cryptographic hash function] — хэш-функция, сочетающая в себе свойства хэш-функции односторонней, хэш-функции с преобразами вторыми трудно обнаружимыми и хэш-функции с коллизиями трудно обнаружимыми. Особо выделяют хэш-функции криптографические, задаваемые ключом, имеющие другое содержание.

Хэш-функция криптографическая, задаваемая ключом [cryptographic hash function with key] — хэш-функция криптографическая, реализуемая алгоритмом кодирования имитозащищающего или кодом аутентификации и предназначенная для обеспечения невозможности для противника и/или нарушителя создавать новые или модифицировать передаваемые (или хранимые) сообщения.

Хэш-функция односторонняя [one-way hash function (OWHF)] — хэш-функция, для которой задача поиска преобразов заданных значений является вычислительно трудной.

Хэш-функция с коллизиями трудно обнаружимыми [collision-intractable hash function] — хэш-функция, для которой задача поиска коллизий является вычислительно трудной.

Хэш-функция с преобразами вторыми трудно обнаружимыми [second preimage resistant hash function] — хэш-функция, для которой задача поиска коллизий преобраза второго является вычислительно трудной.
Целостность [integrity] — отсутствие изменений в передаваемой или хранимой информации по сравнению с ее исходной записью. Необходимым условием соблюдения Ц. является защищенность сообщения от преднамеренной или случайной несанкционированной модификации или уничтожения.

Целостность выделенных полей [selective field integrity] — обеспечение возможности проверки того, что выделенные поля передаваемых данных не подверглись несанкционированной модификации или уничтожению.

Центр доверия [trusted entity, trusted authority, authority, trusted third party] — особый участник протокола криптографического, которому доверяют все остальные его участники, введенный в протокол для усиления его безопасности. Различают следующие виды Ц.д.: центр регистрации, центр распределения ключей, центр сертификации, центр установки меток временных, центр нотаризации и т.д.

Центр распределения ключей [key distribution center] — центр доверия, распределяющий среди участников (протокола) ключи секретные в шифросистемах симметричных или шифросистемах асимметричных.

Центр регистрации [registration center] — центр доверия, который работает вместе с центром сертификации, выполняя роль местного автономного центра хранения реестра сертификатов ключей. Основные функции Ц.р. — регистрация пользователей в системе и присвоение им уникальных идентификаторов, оптимизация управления реестром сертификатов при большом числе запросов, позволяющая масштабировать систему управления сертификатами для большого числа пользователей на большой территории, одновременно сдвига процесс подтверждения ближе к пользователям.

Центр сертификации (ключей открытых) [certification center] — центр доверия, обеспечивающий аутентичность ключей открытых путем придания им сертификатов ключей, заверенных подписью цифровой.

Цикл (раунд) (протокола криптографического) [round, pass (of cryptographic protocol)] — в протоколах криптографических с двумя участниками — временной интервал, в котором активен только один из участников. Другое название — проход (pass) протокола. Ц.(р.) завершается формированием и отсылкой сообщения с последующим переходом активного участника в состояние ожидания и передачей активности другому участнику. В протоколах с тремя и более участниками в синхронном случае цикл — период времени между двумя
точками синхронизации. К очередной точке синхронизации каждый участник должен отослать все сообщения, которые ему предписано передать другим участникам в текущем Ц. В протоколах доказательства интерактивного ц.(р.) часто называют комбинацию из трех шагов: заявка, запрос, ответ. В асинхронном случае понятие Ц.(р.) условно.

Цикл жизненный ключ [key lifetime] — последовательность стадий, которые проходят ключи от момента генерации до момента уничтожения. Включает такие основные стадии, как: генерация, регистрация ключей (и пользователей), инициализация, период действия, хранение, смена, архивирование, уничтожение и восстановление.

Цикл (раунд) шифрования [round] — один шаг (из ряда однородных шагов) в алгоритме шифрования итеративном.

Шаг (протокола) [step (of a protocol), protocol action] — конкретное законченное действие, выполняемое участником (протокола) во время одного цикла (раунда) протокола, например вычисление значения некоторой функции, проверка правильности сертификата ключа, генерация случайного числа, отправка сообщения и т.п.

Шифр [cipher] — семейство обратимых отображений множества последовательностей блоков текстов (сообщений) открытых в множество последовательностей блоков текстов (сообщений) шифрованных и обратно, задаваемых функцией шифрования. Каждое из отображений определяется некоторым параметром, называемым ключом, и описывается некоторым алгоритмом шифрования, реализующем один из режимов шифрования. Математическая модель Ш. включает алгоритм зашифрования, алгоритм расшифрования, определение режима шифрования, а также модель множества текстов открытых. В зависимости от способа представления текстов открытых (сообщений) различают блочные, поточные и другие Ш. Основными требованиями, определяющими качество Ш., являются: стойкость криптографическая, имитостойкость, помехоустойчивость шифра и др.

Шифр гаммирования [keystream cipher] — шифр, в котором функция зашифрования осуществляет гаммирование.

Шифр замены простой [substitution cipher] — шифр, в котором функция зашифрования состоит в замене блоков текста (сообщения) открытого блоками текста (сообщения) шифрованного в соответствии с ключом, представляющим собой подстановку на множестве блоков текста. См. также Ключ коммутаторный.
Шифр перестановки [permutation cipher] — шифр, в котором текст (сообщение) шифрованный получается из текста (сообщения) открытого перестановкой блоков текста (сообщения) открытого.

Шифр совершенный [perfect cipher] — шифр, при использовании которого текст шифрованный не дает противнику, не знающему ключа секретного, никакой информации о тексте открытом, т.е. условное распределение на множестве текстов открытых при заданном тексте шифрованном совпадает с безусловным распределением на множестве текстов открытых.

Шифрование [encryption, enciphering] — термин, объединяющий термины зашифрование и расшифрование.

Шифрование аппаратное [hardware encryption] — шифрование, выполняемое с применением средств криптографических аппаратных.

Шифрование программное [software encryption] — шифрование, выполняемое только с применением средств криптографических программных.

Шифрование сеанса [session encryption] — способ реализации сеанса связи между двумя сторонами, при котором все передаваемые в процессе его выполнения сообщения шифруются на специально сгенерированном для данного сеанса ключе сеансовом.

Шифросистема [cryptosystem, cipher] — см. Система шифрования.

Шифросистема асимметричная [public-key cryptosystem, asymmetric cryptosystem, син. — шифросистема с ключом открытым] — система шифрования, в которой асимметричным образом используются ключи двух видов — ключи открытые и ключи секретные. Ключ открытый участника протокола задает процесс зашифрования направляемых в его адрес сообщений и является общедоступным. Ключ секретный участника протокола задает процесс расшифрования направляемых в его адрес сообщений и хранится им в тайне. Стоимость криптографическая Ш.а. определяется трудоемкостью, с которой противник и/или нарушитель может вычислить ключ секретный, исходя из знания ключа открытого и другой дополнительной информации о шифросистеме.

Шифросистема блочная [block ciphering system] — система шифрования, в которой функция зашифрования реализуется алгоритмом зашифрования блочным.

Шифросистема поточная [stream ciphering system] — система шифрования, в которой функция зашифрования реализуется алгоритмом зашифрования поточным.

Шифросистема с ключом открытым [public-key cryptosystem] — См. Шифросистема асимметричная.
Шифросистема с ключом секретным [private-key cryptosystem] — см. Шифросистема симметричная.

Шифросистема симметричная [secret key cryptosystem, symmetric cryptosystem, син. — шифросистема с ключом секретным] — система шифрования, в которой симметричным образом используются секретные ключи шифрования и ключи расшифрования. В Ш.с. ключи шифрования и расшифрования в большинстве случаев совпадают, а в остальных случаях один легко определяется по другому. Стоимость криптографическая Ш.с. определяется трудоемкостью, с которой противник и/или нарушитель может вычислить любой из секретных ключей, и оценивается при общепринятом допущении, что противнику и/или нарушителю известны все элементы шифросистемы, за исключением ключа секретного (правило Керкгоффса).

Шифросистема RSA [RSA cryptosystem] — шифросистема асимметричная, реализующая алгоритм шифрования RSA.

Шифротекст [ciphertext] — текст, полученный в результате шифрования текста открытого.

Э

Энтропия [entropy] — теоретико-информационная характеристика распределения случайной величины. Энтропия (по К. Шеннону) дискретной случайной величины $S$ с распределением $(p_1, ..., p_n, ...)\) равна $H(S) = -\sum p_i \cdot \log(p_i)$.

Энтропия алгоритмическая [algorithmic entropy, kolmogorov complexity entropy] — введенная А.Н. Колмогоровым мера количества информации, необходимого для описания конечного объекта. Под Э.а. двоичного слова понимают сложность этого слова относительно оптимального способа его описания. (См. сложность последовательности по Колмогорову.) Понятие алгоритмической энтропии связано с понятием энтропии случайной величины (по К. Шеннону).

Я

Ящик черный [black-box] — конечный автомат, у которого известны только входной и выходной алфавиты и доступны для наблюдения выходные последовательности при произвольных входных последовательностях. Если дополнительно известна оценка числа состояний автомата, то говорят о ящике черном относительном.
СПИСОК ЛИТЕРАТУРЫ


28. Швечкова О.Г. Основы теории и практики реализации механизмов информационной безопасности: метод. указ. к лаб. работам / О.Г. Швечкова [и др.]; РГРТУ. — Рязань, 2008.
61. Информация с официального сайта Microsoft.
63. ГОСТ Р ИСО/МЭК 18028-1-2008: Информационная технология. Методы и средства обеспечения безопасности. Сетевая безопасность
информационных технологий. Ч. 1. Менеджмент сетевой безопасности.
64. ГОСТ Р 53114-2008: Защита информации. Обеспечение информационной безопасности в организации. Основные термины и определения.
71. Евдокимова Л.М., Электронный документооборот и обеспечение безопасными средствами WINDOWS [Текст]: учеб. пособие / Л.М. Евдокимова, В.В. Корабкин, А.Н. Пылькин, О.Г. Швечкова. — М.: КУРС. 2017.

165


83. Хореев П. Б. Методы и средства защиты информации в компьютерных системах [Текст]: учеб. пособие для студ. вузов / П. Б. Хореев. — М.: Академия, 2005.

84. Указ Президента РФ от 5 декабря 2016 г. № 646 «Об утверждении Доктрины информационной безопасности Российской Федерации».

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ................................................................................................................................................. 3
Основные положения Доктрины информационной безопасности РФ............................................. 4
Тема 1. ОСНОВНЫЕ ПОЛОЖЕНИЯ ТЕОРИИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ................................................................................................................................. 9

1.1. Понятие информационной безопасности. Задачи обеспечения информационной безопасности................................................................................................................................. 9

1.2. Угрозы информационной безопасности......................................................................................... 12
1.2.1. Основные угрозы конфиденциальности.................................................................................. 13
1.2.2. Основные угрозы целостности............................................................................................... 14
1.2.3. Основные угрозы доступности............................................................................................... 15

1.3. Роль криптографических протоколов в задаче обеспечения информационной безопасности ............................................................................................................................... 15

1.4. Общие сведения классической криптографии.............................................................................. 17
1.4.1. Основные понятия и определения классической криптографии.......................................... 17
1.4.2. Стохастость алгоритмов шифрования.................................................................................... 19
1.4.3. Основные методы криптографической защиты информации............................................. 22
1.4.4. Классификация криптографических алгоритмов.................................................................... 24

1.5. Аппаратная и программная реализация алгоритмов шифрования........................................ 33
1.5.1. Аппаратная реализация криптографических алгоритмов..................................................... 34
1.5.2. Программная реализация криптографических алгоритмов................................................ 35
1.5.3. Программно-аппаратная реализация криптографических алгоритмов............................ 35

Контрольные вопросы ................................................................................................................................. 36

Тема 2. ШИФРЫ ЗАМЕНЫ .......................................................................................................................... 38

2.1. Шифры простой замены.................................................................................................................. 39
2.1.1. Система шифрования Цезаря................................................................................................. 39
2.1.2. Аффинная система подстановки Цезаря................................................................................ 40
2.1.3. Лозунговый шифр.................................................................................................................... 42
2.1.4. Полибийский квадрат............................................................................................................. 44
2.1.5. Шифрующая таблица Триземуса........................................................................................... 45
2.1.6. Биграммный шифр Плейфера................................................................................................. 46
2.1.7. Система омофонов.................................................................................................................. 49
2.1.9. Тесты по теме «Алгоритмы простой замены» ..................................................................... 51

2.2. Шифры сложной замены.............................................................................................................. 55
2.2.1. Шифр Гронсфельда................................................................................................................ 56
2.2.2. Система шифрования Вижнера............................................................................................. 57
2.2.3. Шифр Вижнера с автоключом ................................................................................................ 58
2.2.4. Шифр Вижнера с перемешанным алфавитом....................................................................... 60
2.2.5. Двойной квадрат Уитстона.................................................................................................... 61
2.2.7. Тесты по теме «Алгоритмы сложной замены» .................................................................... 62
2.2.8. Задание для самостоятельной работы по практической реализации алгоритмов шифрования методами замены................................................................. 66