Белорусский национальный технический университет
Факультет Международный институт дистанционного образования
Кафедра «Информационные системы и технологии»

ЭЛЕКТРОННЫЙ УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

«ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ»

для специальности:
1-40 01 01 «Программное обеспечение информационных технологий»

Составитель:
Макареня Сергей Николаевич, доцент кафедры ИСиТ

Минск БНТУ 2023

СОДЕРЖАНИЕ

Перечень материалов

Электронный учебно-методический комплекс включает:

— теоретический раздел (конспект лекций);

— практический раздел (лабораторные занятия);

— контроль знаний (задания контрольной работе);

— вспомогательный раздел (программа дисциплины, литература, список вопросов для зачета).

Пояснительная записка

Электронный учебно-методический комплекс разработан для студентов специальности 1-40 01 01 «Программное обеспечение информационных технологий». Информационное наполнение ЭУМК соответствует программе дисциплины «Основы информационной безопасности».

ЭУМК может использоваться как при проведении занятий по дисциплине «Основы информационной безопасности», так и для организации самостоятельной работы студентов. Внедрение ЭУМК будет способствовать более эффективному овладению теоретическими и практическими основами обеспечения информационной безопасности.

Информация в ЭУМК хорошо структурирована. Теоретический раздел включает основные темы курса. Лабораторный практикум содержит необходимый базовый методический материал (цель лабораторной работы, краткие теоретические сведения, задания на лабораторную работу, контрольные вопросы). В ЭУМК приводится также список литературы и актуальная программа дисциплины. Модуль контроля знаний содержит список вопросов к зачету.

ЭУМК разработан в виде pdf-файла и не требует установки специального программного обеспечения.

Учебная программа составлена на основе образовательного стандарта ОСВО 1-40 01 01-2021 и учебного плана специальности 1-40 01 01 «Программное обеспечение информационных технологий» специализации 1-40 01 01 01 «Веб-технологии и программное обеспечение мобильных систем».

РАЗДЕЛ 1. ТЕОРЕТИЧЕСКИЙ

Лекция 1. Введение в защиту информации.

Ученые, анализируя тот или иной отрезок истории развития человеческого общества, присваивают ему краткое наименование, в основе которого лежит наиболее характерное свойство, присущее именно данному отрезку истории. Известны различные классификации, например, по классовым признакам, по технологическим и т. д. Если следовать технологической классификации, то сегодня человечество переходит от индустриального общества к информационному. Информация из абстрактного «знания» превращается в материальную силу. Информационные технологии коренным образом изменяют облик материального производства, позволяют экономить материальные ресурсы, создавать новые приборы и системы, в буквальном смысле изменили наши представления о времени и пространстве.

Однако широкое внедрение в жизнь информационных технологий, управляющих жизненно важными процессами, к сожалению, сделало их достаточно уязвимыми со стороны естественных воздействий среды и искусственных воздействий со стороны человека. Возникла проблема обеспечения безопасности информационных систем в широком смысле слова или защиты информации в более узкой постановке.

1.1 Цели защиты информации. Основные понятия.

Целями защиты являются: предотвращение утечки, хищения, утраты, искажения, подделки, несанкционированных действий по уничтожению, модификации, копированию, блокированию документированной информации и иных форм незаконного вмешательства в информационные системы.

Под информацией будем понимать сведения о лицах, предметах, фактах, событиях, явлениях и процессах.

Информация может существовать в виде документа (бумажного), в виде физических полей и сигналов (электро-магнитных, акустических, тепловых и т.д.), в виде биологических полей (память человека).

В дальнейшем будем рассматривать информацию в документированной (на бумаге, дискете и т.д.) форме, в форме физических полей (радиосигналы, акустические сигналы). Среду, в которой информация либо создается, либо передается, обрабатывается, хранится будем называть информационным объектом.

Под безопасностью информационного объекта (ИО) будем понимать его защищенность от случайного или преднамеренного вмешательства в нормальный процесс его функционирования.

Природа воздействия на ИО может быть двух видов: непреднамеренной (стихийные бедствия, отказы, ошибки персонала и т.д.); преднамеренной (действия злоумышленников). Все воздействия могут привести к последствиям (ущербу) трех видов: нарушению конфиденциальности; нарушению целостности; нарушению доступности.

Нарушение конфиденциальности – нарушение свойства информации быть известной только определенным субъектам.

Нарушение целостности – несанкционированное изменение, искажение, уничтожение информации.

Нарушение доступности (отказ в обслуживании) – нарушается доступ к информации, нарушается работоспособность объекта, доступ в который получил злоумышленник.

В отличие от разрешенного (санкционированного) доступ к информации в результате преднамеренных действий злоумышленник получает несанкционированный доступ (НСД). Суть НСД состоит в получении нарушителем доступа к объекту в нарушении установленных правил.

Под угрозой информационной безопасности объекта будем понимать возможные воздействия на него, приводящие к ущербу.

Некоторое свойство объекта, делающее возможным возникновение и реализацию угрозы, будем называть уязвимостью.

Действие злоумышленника, заключающееся в поиске и использовании той или иной уязвимости, будем называть атакой.

Целью защиты ИО является противодействие угрозам безопасности.

Защищенный ИО – это объект со средствами защиты, которые успешно и эффективно противостоят угрозам безопасности.

Комплексная защита ИО – совокупность методов и средств (правовых, организационных, физических, технических, программных).

Политика безопасности – совокупность норм, правил, рекомендаций, регламентирующих работу средств защиты ИО от заданного множества угроз безопасности.

Схематично основное содержание предмета защиты информации представлено на рис.1.1

1.2 Классификация угроз, методов и средств защиты информации.

Под угрозой информационной безопасности объекта будем понимать возможные воздействия на него, приводящие к ущербу. К настоящему времени известно большое количество угроз. Приведем упрощенную их классификацию. Угрозы делятся по свойству информации, против которого они направлены:

угрозы физической и логической целостности (уничтожение или искажение информации); угрозы конфиденциальности информации; угрозы доступности (работоспособности); угрозы праву собственности.

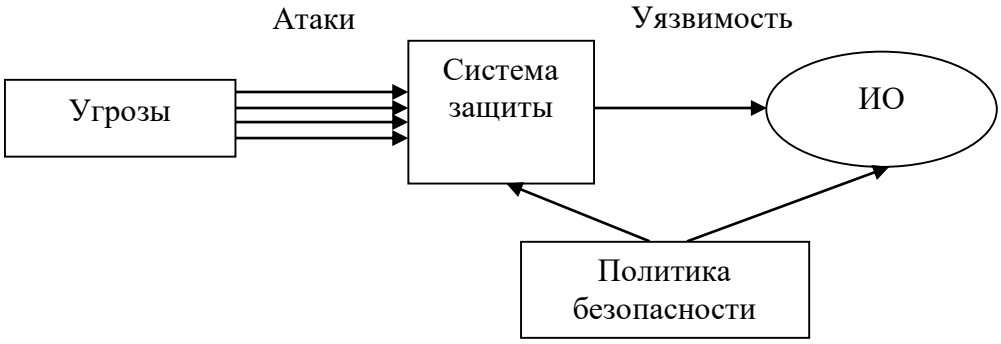


Рис.1.1. Содержание предмета защиты информации

По происхождению: случайные (отказы, сбои, ошибки, стихийные явления); преднамеренные (злоумышленные действия людей).

По источникам: люди (персонал, посторонние); технические устройства; модели, алгоритмы, программы; внешняя среда (состояние атмосферы, побочные шумы, сигналы и наводки).

Рассмотрим более подробно перечисленные угрозы.

Случайные угрозы обусловлены недостаточной надежностью аппаратуры и программных продуктов, недопустимым уровнем внешних воздействий, ошибок персонала. Методы оценки воздействия этих угроз рассматриваются в других дисциплинах (теории надежности, программировании, инженерной психологии и т.д.).

Преднамеренные угрозы связаны с действиями людей. Это и работники спецслужб, хакеры, работники самого объекта. Огромное количество разнообразных ИО, делает бессмысленным перечисление всех возможных угроз для информационной безопасности, поэтому в дальнейшем при изучении того или иного раздела мы будем рассматривать основные угрозы для конкретных объектов. Например, для несанкционированного доступа к информации вычислительной системы злоумышленник может воспользоваться штатными каналами доступа, если нет никаких мер защиты: через терминалы пользователей; через терминал администратора системы; через удаленные терминалы.

И через нештатные каналы: побочное э/м излучение информации с аппаратуры системы; побочные наводки информации по сети электропитания и заземления; побочные наводки информации на вспомогательных коммуникациях; подключение к внешним каналам связи.

1.3 Классификация методов защиты информации.

Все методы защиты информации по характеру проводимых действий можно разделить на: законодательные (правовые); организационные; технические; комплексные.

Для обеспечения защиты объектов информационной безопасности должны быть соответствующие правовые акты, устанавливающие порядок защиты и ответственность за его нарушение. Законы должны давать ответы на следующие вопросы: что такое информация, кому она принадлежит, как может с ней поступать собственник, что является посягательством на его права, как он имеет право защищаться, какую ответственность несет нарушитель прав собственника информации.

Установленные в законах нормы реализуются через комплекс организационных мер, проводимых, прежде всего государством, ответственным за выполнением законов, и собственниками информации. К таким мерам относятся и издание подзаконных актов, регулирующих конкретные вопросы по защите информации (положения, инструкции, стандарты т.д.), и государственное регулирование сферы через систему лицензирования, сертификации, аттестации.

Поскольку в настоящее время основное количество информации генерируется, обрабатывается, передается и хранится с помощью технических средств, то для конкретной ее защиты в информационных объектах необходимы технические устройства. В силу многообразия технических средств нападения приходится использовать обширный арсенал технических средств защиты.

Наибольший положительный эффект достигается в том случае, когда все перечисленные способы, применяются совместно, т.е. комплексно.

Лекция 2. Методы и средства защиты от утечки по техническим каналам.

2.1. Технические каналы утечки информации.

Под техническими каналами понимаются каналы доступа к информации, возникающие вследствие либо естественных физических явлений, сопровождающих работу информационных объектов, либо создаваемых искусственно нарушителем. Наиболее часто технические каналы используются для овладения компьютерной, акустической, телефонной и визуальной информацией.

2.1.1. Канал побочных электромагнитных излучений и наводок (ПЭМИН).

Работа средств вычислительной техники сопровождается электромагнитными излучениями и наводками на соединительные проводные линии, цепи "питания", "земля", возникающими вследствие электромагнитных воздействий в ближней зоне излучения, в которую могут попадать также провода вспомогательной и посторонней аппаратуры. В некоторых случаях информацию, обрабатываемую компьютерами, можно восстановить путем анализа электромагнитных излучений и наводок. В персональном компьютере основными источниками электромагнитных излучений являются устройства ввода и вывода информации совместно с их адаптерами (монитор, принтер, клавиатура, печатающее устройство и т. д.), а также центральный процессор. Утечке информации в ПК способствует применение коротких видеоимпульсов прямоугольной формы и высокочастотных коммутирующих сигналов. Исследования показывают, что излучение видеосигнала монитора является достаточно мощным, широкополосным и охватывает диапазон метровых и дециметровых волн. Применение в ПК импульсных сигналов прямоугольной формы и высокочастотной коммутации приводит к тому, что в спектре излучений будут компоненты с частотами вплоть до СВЧ. Хотя энергетический спектр сигналов убывает с ростом частоты, но эффективность излучения при этом увеличивается, и уровень излучений может оставаться постоянным до частот нескольких гигагерц. Резонансы из-за паразитных связей могут вызывать усиление излучения сигналов на некоторых частотах спектра.

Основными цепями распространения опасного сигнала являются: информационные цепи видеосигнала; цепи обмена информацией с жестким диском; цепи обмена информацией с ОЗУ; цепи ввода информации с клавиатуры; цепи вывода информации на принтер; цепи питания; вынос

наведенных информационных сигналов через подключенные к системному блоку устройства, непосредственно не участвующие или не участвующие в данный момент времени в процессе передачи информации (мышь, клавиатура, информационный кабель принтера).

Наиболее опасными с точки зрения осуществления возможного несанкционированного съема обрабатываемой информации за счет ПЭМИН являются цепи, в которых сигналы имеют периодический характер. Примером таких цепей являются информационные цепи видеосигнала, представляющего регулярный сигнал в последовательном коде. Менее опасные - сигналы в цепях обмена информацией с жестким и гибким дисками, цепях клавиатуры, т.к. они носят не регулярный характер. Еще менее опасны сигналы в цепи вывода информации на принтер, в цепи обмена информацией с ОЗУ, передаваемые в параллельном коде.

2.1.2. Канал утечки акустической информации

Наиболее простым способом перехвата речевой информации является подслушивание (прямой перехват). Разведываемые акустические сигналы могут непосредственно приниматься ухом человека, реагирующим на изменение звукового давления, возникающего при распространении звуковой волны в окружающем пространстве. Диапазон частот акустических колебаний, слышимых человеком, простирается от 16…25 Гц до 18…20 кГц в зависимости от индивидуальных особенностей слушателя. Человек воспринимает звук в очень широком диапазоне звуковых давлений. Одной из опорных величин этого диапазона является стандартный порог слышимости. Под ним условились понимать эффективное значение звукового давления, создаваемое гармоническим звуковым колебанием частоты F=1000 Гц, едва слышимым человеком со средней чувствительностью слуха. Порогу слышимости соответствует звуковое давление $P=2\times10^{-5}$ Па. В случаях, когда уровни звукового давления, создаваемого звуковой волной, ниже порога слышимости, когда нет возможности непосредственно прослушивать речевые сообщения, или, когда требуется их зафиксировать (записать), используют микрофоны и радиомикрофоны. Микрофон является преобразователем акустических колебаний в электрические сигналы, а радиопередатчик позволяет передавать эти сигналы на значительные расстояния.

Кроме того, если источник акустической информации находится внутри помещения, то акустическая информация может быть перехвачена с помощью электронного стетоскопа. Акустические колебания источника возбуждают в ограждающих конструкциях помещения (стенах, полу, потолке) микроскопические вибрации, эти вибрации улавливаются и преобразуются в

электрические сигналы. Электронный стетоскоп представляет собой вибродатчик, усилитель, оконечное устройство.

Вибродатчик - преобразователь вибрации в электрический сигнал. Вибродатчик прикрепляется к стене, потолку, можно в соседней комнате. Размеры несколько см., вес десятки-сотни граммов, коэффициент усиления десятки тысяч. Может соединяться с наушниками, записывающей аппаратурой, с передатчиком.

2.1.3. Канал утечки телефонной информации

Канал утечки речевой информации, передаваемой по телефонным каналам связи, возникает за счет несанкционированного подключения злоумышленника к проводным линям связи или за счет перехвата сообщений, передаваемых по радиоканалу. Подключение к проводным линям связи может быть контактным и бесконтактным. В качестве оконечных устройств используют специальные магнитофоны, включающиеся по командам (ключевое слово, появление сигнала, команда управления). Часто записывающее устройство может находиться на расстоянии. В этом случае используется телефонный ретранслятор, представляющий собой радиопередатчик, питающийся от телефонной сети. Телефонные ретрансляторы могут быть закамуфлированы под различные элементы телефонной сети: розетка, конденсатор, фильтр, реле, и т.д. Дальность действия несколько сот метров. Для перехвата сообщений, передаваемых по радиоканалу, используется сканирующий приемник, компьютер со специальным программным обеспечением.

2.1.4. Канал утечки визуальной информации

Визуальная информация может быть перехвачена с помощью оптической аппаратуры с большого расстояния (с искусственных спутников земли, с удаленных наблюдательных пунктов), а также с помощью миниатюрной аппаратуры в непосредственной близи от источника информации (фотоаппарат, видеокамера).

2.2 Методы и средства защиты информации от утечки по техническим каналам

2.2.1. Защита информации от утечки по каналу побочных электромагнитных излучений и наводок

Для защиты информации от утечки по каналу побочных электромагнитных излучений применяют проведение защитных мероприятий помещений в целом путем их экранирования, защиту излучающей аппаратуры, а если этих мер недостаточно, то используют электромагнитное зашумление.

Экранирование помещений можно выполнить, используя различные материалы: листовую сталь, проводящую медную сетку, алюминиевую фольгу. Расчеты показывают, что медная сетка с ячейкой 2,5 мм даст приемлемую эффективность экранирования. Достаточно эффективный и, немаловажно, дешевый экран получается при использовании алюминиевой фольги. Экранировать нужно все помещение полы, стены, потолки, двери. На практике это может выглядеть следующим образом, выбирается одна наиболее удобно расположенная комната, желательно не имеющая стен, смежных с неконтролируемыми помещениями, а также без вентиляционных отверстий. На пол, например, под линолеум, укладывается фольга, сетка и т. д., стены под обоями или панелями покрываются фольгой. Потолки можно сделать алюминиевыми подвесными, а на окнах использовать алюминиевые жалюзи, специальные проводящие стекла или проводящие шторы. Важно обеспечить электрический контакт экранов пола, потолка, стен и т. д. по всему периметру помещения.

Для снижения уровня побочных излучений аппаратуры при ее изготовлении используют специальные конструкторские и технологические меры. Например, у персональных компьютеров корпуса системного блока выполняются в виде замкнутого электромагнитного экрана. Крышка системного блока в нижней части имеет специальные пазы, которые при установленной крышке играют роль предельных волноводов. Корпуса разъемов клавиатуры и мыши изнутри корпуса системного блока заключены в электромагнитные экраны. В цепи элементов индикации системного блока включены ферритовые фильтры. Корпус видеомонитора выполняется из металла или пластмассы с напылением токопроводящего материала, экран покрывается прозрачной токопроводящей пленкой. Соединительные кабели помещаются в экран. Необходимо качественное заземление всех экранов. Снижения мощности излучаемых сигналов может быть достигнуто и за счет разрушения периодичности видеосигналов путем введения случайности развертки видеомонитора. Практически без ухудшения воспроизведения изображения возможно снизить амплитуды гармоник излучения. Этот способ получил название способ снятия повторяемости.

Если конструкторскими и технологических мерами не удается обеспечить требуемые характеристики, возможно применение генераторов шума. Генератор шума представляет собой источник электромагнитных колебаний, спектр которых перекрывает весь частотный диапазон возможных опасных излучений, а мощность достаточна для маскировки полезного сигнала. Конструктивно он может выполняться в виде платы, вставляемой в слот компьютера, или в виде автономного блока. Серьезную проблему представляет защита проводных

линий, выходящих за пределы помещений, в которых находится компьютерное оборудование (цепи электропитания, линии телефонной связи, цепи пожарной и охранной сигнализации). Экранирование таких линий позволяет защититься от наводок, распространяемых по этим линиям за пределами защищаемых помещений. Наиболее экономичным способом экранирования считается размещение информационных кабелей в экранирующий распределительный короб. Когда такого короба не имеется, то приходится экранировать отдельные линии связи. Для этого используют либо провода в экранирующей оболочке, либо помещать в такую оболочку (например, фольгу) существующие провода. Эффективно применять при этом скрутку двух проводов (бифиляр) или трех проводов (трифиляр), уменьшающую излучение. При использовании трифиляра третий провод заземляется и служит экраном. Очень эффективен экранированный коаксиальный кабель. Необходимо проследить за тем, чтобы кабели разных линий связи были максимально разнесены для уменьшения взаимных наводок. После проведения работ по экранированию помещения необходимо выполнить работы по заземлению экранов. Обычно это делается путем параллельного подключения к существующему контуру заземления, предварительно проверив его сопротивление (оно должно быть не более 4 Ом).

2.2.2. Защита акустической информации.

Для защиты акустической информации от прослушивания обычными микрофонами, находящимися за пределами помещения, необходимо при проектировании и строительстве помещения обеспечить требуемую звукоизоляцию, это достигается использованием двойных рам (стеклопакетов), дверей, звукоизоляционных материалов в стенах, потолках, полах.

Для защиты речевой информации от микрофонов и радиомикрофонов, установленных внутри помещений, применяются генераторы акустических помех. Существуют два типа генераторов акустических помех: генераторы шумовых помех и генераторы звукоподобных сигналов. Оба типа генераторов маскируя полезный сигнал, создают трудности при разговоре. С точки зрения снижения помех и санкционированным слушателям предпочтительнее генераторы звукоподобных сигналов, т.к. при меньшем уровне сигнала они обеспечивают требуемое маскирование полезного сигнала.

Однако наиболее эффективным методом защиты является периодический поиск и изъятие радиомикрофонов. Для этого используют: детекторы радиоизлучений, сканирующие приемники, поисковые комплексы, нелинейные локаторы.

Основой обнаружения работающих радиомикрофонов может быть любой приемник радиосигналов, позволяющий определить факт наличия в помещении

источника излучения и определить его местоположение. Простейшим радиопеленгатором является детектор (индикатор) радиоизлучений. Детектор представляет собой широкополосный приемник радиосигнала со световой или звуковой индикацией. Недостатками такого устройства являются трудности в определении место излучения; помехи фоновых радиоизлучений промышленных источников.

Существует набор тактических приемов преодоления недостатков (отключение электросети, изменение ориентации, измерение фона). Лучшими характеристиками обладают более сложные детекторы с положительной обратной связью (ПОС) по звуковой частоте. Принцип работы детектора с ПОС по звуковой частоте заключается в следующем.

Оконечное устройство детектора изготавливается в виде динамика на длинном проводе. Сигнал звуковой частоты, излучаемый динамиком принимается радиомикрофоном, преобразуется в радиосигнал, принимается детектором усиливается и вновь излучается динамиком. Таким образом, возникает ПОС в контуре. Контур возбуждается при наличии работающего радиомикрофона. Перемещая динамик в пространстве легко определить направление на источник радиоизлучения.

Более эффективным средством является сканирующий по частоте приемник. Поскольку рабочая частота радиомикрофона неизвестна и может находиться в диапазоне от десятков МГц до 1ГГц, то для ее обнаружения необходим приемник с перестраиваемой полосой приема. Такие приемники получили название сканирующих. Существуют малогабаритные сканирующие приемники с телескопическими антеннами, с автоматической и ручной перестройкой частоты в широком диапазоне, с запоминанием частот и другим набором сервисных функций. На базе таких приемников изготавливаются поисковые комплексы, содержащие портативный компьютер, генератор звукового теста, коррелятор. Такие комплексы в автоматическом режиме за несколько минут определяют частоты, и уровни излучений, определяют направление и расстояние до скрытых средств.

Описанные ранее средства позволяют обнаружить радиомикрофоны в работающем состоянии (в активном). Если радиомикрофон управляется дистанционно или по программе и на момент поиска выключен (пассивен), то его можно обнаружить с помощью устройства, получившего название нелинейный локатор (НЛ). Принцип действия НЛ основан на преобразовании спектра гармонического сигнала локатора нелинейным элементом радиомикрофона (диодом, транзистором). Что приводит к появлению в спектре отраженного сигнала, гармоники с удвоенной частотой.

Для защиты акустической информации от утечки по вибрационным каналам используются системы виброакустического зашумления, состоящие из генераторов электрических колебаний и преобразователей.

Отличие генераторов вибропомех заключается в мощности колебаний и преобразователей электрических сигналов в вибросигналы. Генераторы выполняются аналогично генераторам акустических помех, а в качестве преобразователей используются устройства, преобразующие электрические сигналы в энергию упругих колебаний среды.

При возбуждении конструкций, имеющих высокое акустическое сопротивление (кирпичные стены, бетонные перекрытия), согласование в широком частотном диапазоне проще осуществляется с устройствами, имеющими высокий механический импеданс подвижной системы. Такими устройствами являются пьезокерамические преобразователи. Электромагнитные преобразователи имеют худшие характеристики.

Наводимые в конструкциях вибрационные шумы должны маскировать вибрации от полезного речевого сигнала, создавая при этом минимум помех для людей, работающих в помещении.

Помехи создаются за счет акустического сигнала вибродатчика и за счет вибрационных колебаний конструкций. Основная доля при этом падает на акустический паразитный сигнал. Наряду с выбором типа вибратора, необходимо вибропреобразователи располагать не на поверхности конструкции, а в специально изготовленной нише, тщательно заделанной после установки преобразователя.

2.2.3. Защита телефонной информации.

Для защиты информации, открыто передаваемой по проводным линиям связи, необходимо иметь технические средства, обнаруживающие факт прослушивания линии. Для этого существует ряд приборов
от простейших индикаторов подключения до очень сложных анализаторов характеристик линий. Работа всех этих приборов основана на фиксации изменений параметров телефонной сети при подключении к ним устройств прослушивания. Простейшие приборы реагируют на изменение напряжения в линии или сопротивления, более сложные фиксируют частотные и импульсную характеристики.

При этом, однако, надо помнить, что необходимо иметь значения характеристик до подключения устройств прослушивания, сеть должна иметь стабильные характеристики, подключение устройств должно изменять параметры линии.

Наиболее эффективным способом защиты речевой информации в каналах связи являются методы, основанные на преобразовании речевой информации по определенному алгоритму, делающие невозможным понимание речи.

Получили распространение два способа скремблирования речи: аналоговое скремблирование (аналоговое преобразование) и цифровое криптопреобразование (цифровое скремблирование).

Суть аналогового скремблирования заключается в аналоговом преобразовании речевого сигнала после которого он становится неразборчивым. Для его понимания необходимо провести обратное преобразование. Такое преобразование осуществляется чаще всего за счет инвертирования частотного спектра речевого сигнала в соответствии с условленным порядком.

Самым надежным способом защиты является криптографическое преобразование речевого сигнала. Речевой аналоговый сигнал преобразуется в цифровой, затем в специальном вычислителе производится специальное математическое преобразование (шифрование) с использованием секретного ключа, зашифрованная информация передается в канал связи. Второй абонент с помощью своего секретного ключа (такого же, что и у первого) расшифровывает информацию и преобразует ее в аналоговый сигнал.

Аппарат, содержащий обе линейки, позволяет осуществлять дуплексную связь. Наличие компрессора и декомпрессора обусловлено низкой скоростью передачи информации по реальным телефонным каналам 2400 - 9600 бит/сек. На выходе же АЦП при стандартной частоте дискретизации 8кГц и 8 битовом представлении выбором скорости составляет 64 кбит/сек. Компрессор реализует один из алгоритмов сжатия LPC, GSM, CELP. Важными проблемами в такой связи являются синхронизация работы двух аппаратов и распределение ключевой информации.

Лекция 3. Основные функции системы защиты от НСД в компьютерных системах.

В общем случае для типовой компьютерной системы (КС) система защиты от НСД должна обеспечивать:

1. Идентификацию и аутентификацию пользователя при начале работы в КС.
2. Управление доступом к ресурсам и процессам КС.
3. Контроль целостности объектов КС.
4. Мониторинг процессов и событий КС.
5. Управление безопасностью КС.

В КС защита от НСД осуществляется программными, аппаратными или аппаратно-программными средствами.

3.1 Аутентификация пользователя

Аутентификация означает установление подлинности. Обеспечивает работу в сети только санкционированных пользователей. Чаще всего проводится при входе в сеть, но может проводиться и во время работы. Обычно проводится после процесса идентификации, во время которого пользователь сообщает свой идентификатор (называет себя). В процедуре аутентификации участвуют две стороны: пользователь доказывает свою подлинность, а сеть проверяет это доказательство и принимает решение. В качестве доказательства используют: знание секрета (пароля); владения уникальным предметом (физическим ключом); предъявления биометрической характеристики (отпечатка пальца, рисунка радужной оболочки глаза, голоса).

Наиболее распространенное средство аутентификации – пароль. Используется как при входе в систему, так и в процессе работы. Пароль может вводится с клавиатуры, или с различных носителей цифровой информации или комбинировано. При использовании паролей необходимо соблюдать необходимые требования: по правилам генерации (длина, случайность символов), хранения (хранить в защищенном месте), использования (в зашифрованном виде), отзыва.

В качестве субъектов аутентификации могут выступать не только пользователи, но и различные устройства или процессы. Причем процесс аутентификации может носить обоюдный характер. Обе стороны должны доказать свою подлинность. Например, пользователь, обращающийся к корпоративному серверу, должен убедится, что имеет дело с сервером своего предприятия. В этом случае процедура называется – взаимная аутентификация.

3.2. Управление доступом к ресурсам и процессам КС.

Допущенным в КС субъектам должны предоставляться различные права по доступу к информационному ресурсу и по возможным действиям с ним. Например, доступ к каталогам, файлам, принтерам, доступ к системным функциям: доступ к серверу, установка системного времени, создание резервных копий данных, выключение сервера и т.д. Эти права могут задаваться по-разному. В основном их можно разбить на два класса: произвольный доступ и мандатный доступ.

Произвольный доступ реализуется в операционной системе общего назначения. Задаются определенные операции над определенным ресурсом одному пользователю или группе пользователей, явно указанным своими идентификаторами. При этом пользователь может передавать свои полномочия некоторому процессу и если этим же процессом управляет другой пользователем, то в итоге права одного пользователя становятся доступными другому. Основой произвольного доступа является матрица прав доступа, строки которой соответствуют субъектам (пользователи, процессы и т.д.), а столбцы – объектам (файлы, каталоги и т.д.). В ячейках матрицы содержатся права доступа субъектов к объектам. Пример матрицы прав доступа приведен в табл. 3.1, где R- права доступа пользователя по чтению, W- права доступа пользователя по записи; C- управление доступом для других пользователей.

Табл. 3.1.

Пользователи/Файлы	F1	F2	F3	F4	F5	
Петров	R	W	W		RW	
Иванов	RW		R	R		
Сидоров		RW			R	
Федоров	C	C	C	C	C	

В зависимости от способа представления матрицы прав доступа в ОС различают несколько способов реализации. Наиболее распространенным являются списки прав доступа, биты доступа, парольная защита.

Списки прав доступа. С каждым объектом ассоциируется список пользователей с указанием их прав доступа к объекту. При принятии решения о доступе соответствующий объект проверяется на наличие прав, ассоциированных с идентификатором пользователя, запрашивающего доступ.

Биты доступа вместо списка пользователей с объектом связываются

биты доступа. Например, в ОС UNIX организованы три категории пользователей. Каждой группе разрешены определенные действия. Каждый пользователь получает определенный бит (номер), который определяет к какой группе он относится и какими полномочиями наделен.

При реализации мандатного доступа вся информация делится по уровням конфиденциальности, а все пользователи также делятся на группы по уровням допуска к информации различного уровня конфиденциальности. При этом пользователи не имеют возможности изменять уровень доступности информации. Нормативное управление доступом основано на модели Белла-ЛаПадула, которая описывает правила документооборота, принятые в правительственных учреждениях США. Основным наблюдением, сделанным Беллом и ЛаПадулой, является то, что в официальном документообороте, всем субъектам и объектам присваивается уровень (метка) безопасности. Для предотвращения утечки информации к неуполномоченным субъектам с низкими уровнями безопасности не позволяется читать информацию из объектов с высокими уровнями безопасности. Субъектам не позволяется размещать информацию или записывать ее в объекты с более низким уровнем безопасности. Мандатный доступ является более строгим, исключает волюнтаризм со стороны пользователей. Реализуется в ОС специального назначения.

3.3. Контроль целостности

Подсистема контроля целостности должна контролировать как целостность информационных ресурсов сети, так и целостность системы защиты. Злоумышленник может установить вредоносные программы, типа «троянский конь», поправить системные файлы, предоставить себе несанкционированные полномочия, отключить систему защиты, т.е. нарушить целостность установленного программного обеспечения. Подсистема контроля целостности должна работать пассивно, не мешая работе контролируемой системы. В ходе контроля вычисляется некоторая относительно короткая величина (слепок), зависящая от контролируемых данных, причем любое их изменение должно приводить к существенному изменению этой величины. В простейшем случае в качестве такой величины может использоваться контрольная сумма, более стойкий результат дает использование криптографических методов. Подсистема контроля целостности работает по замкнутому циклу, обрабатывая файлы, системные объекты и атрибуты системных объектов с целью вычисления слепка, затем сравнивает его с предыдущим слепком. При обнаружении расхождения сигнализирует администратору безопасности.

3.4. Аудит

Подсистема аудита предназначена для фиксации событий, связанных с доступом к защищаемым ресурсам. Для этого средствами ОС и или прикладных программ ведутся журналы регистрации событий безопасности, в которые записывается информация о времени, источнике, категории события, коде события, субъект, компьютер и т. д. Многие компоненты КС имеют подобные журналы. Например, коммуникационное оборудование-маршрутизаторы Cisco, межсетевые экраны-Check Point и сетевые ОС начиная с W-NT. Данные журналы ведутся и системами обнаружения атак, входящими в состав системы защиты информации.

3.5 Управление безопасностью в КС

Система защиты информации КС должна быть управляемой. Управляемость необходима для обеспечения ее текущего функционирования (смена паролей, криптографических ключей, изменения списков пользователей, изменения каталогов защищаемых файлов и т.д.) и адаптации к изменившейся политики безопасности сети (изменение правил доступа, изменения длины паролей или ключей, замена используемых криптопротоколов и алгоритмов). При этом функции оперативного управления являются обязательными, т.к. без них систему защиты не возможно эксплуатировать. Функции адаптации к политике безопасности желательны и должны закладываться на этапе проектирования.

3.6. Программные средства защиты от НСД

3.6.1. Защита средствами операционной системы

Для защиты информации от несанкционированного доступа одиночных ПК (не входящих в локальные или глобальные сети) используются программные средства операционных систем или прикладные программы, а также специальные программно-аппаратные средства.

Исторически сложилось так, что до недавнего времени при разработке ОС всеобщего назначения (Windows, Unix) вопросам защиты от НСД внимания уделялось мало. И это вполне объяснимо. Они были рассчитаны на применение пользователями, не обрабатывающими конфиденциальную информацию. В современных версиях ОС эти недостатки по заявлению разработчиков устранены, т.е. большинство описанных выше функций защиты могут быть реализованы. Однако остается открытым вопрос о степени доверия к надежности встроенных функций защиты, поскольку программные коды, например, Windows не известны.

3.6.2. Защита средствами прикладных программ.

Существует ряд программных продуктов, работающих под управлением незащищенных ОС, выполняющих защиту от НСД в КС. Эти программы в основном выполняют функции, сформулированные ранее, и отличаются друг от друга непринципиально. Поэтому коротко рассмотрим наиболее разрекламированное средство – Sekret Net.

Основные выполняемые функции: идентификация пользователей до загрузки ОС (возможно с использованием аппаратных средств); регламентация доступа к физическим и логическим устройствам (дискам, портам, файлам, папкам); контроль целостности ПО защиты; организация и ведение журнала безопасности (регистрация всех событий, относящихся к безопасности).

Средства защиты от НСД, реализованные в виде программ и не могут обеспечить защиту от квалифицированного злоумышленника. Существует достаточно атак, в результате которых злоумышленник перехватывает пароли, списки, присваивает себе полномочия администратора, выдает себя за санкционированного пользователя, модифицирует ОС в части управления и контроля доступа. Для надежной защиты информационных ресурсов ПК необходимо, чтобы критичные для безопасности операции осуществлялись в изолированной операционной среде, недоступной злоумышленнику. Такая среда создается путем использования аппаратно-программных средств (АПС) защиты от НСД.

3.7 Аппаратно- программные средства защиты от НСД.

АПС защиты обеспечивает выполнение всех функций по защите в своей, изолированной операционной среде, недоступной злоумышленнику. По своей архитектуре наиболее эффективные устройства представляют собой автономный компьютер с собственным процессором, памятью, BIOS, операционной системой. АПС защиты управляет включением защищаемого компьютера, процессом авторизации (идентификации и аутентификации), процессом допуска пользователей и процессов к информационным ресурсам ПК и т.д. Современные АПС защиты реализуют криптографические алгоритмы шифрования и протоколы аутентификации и контроля целостности. Наиболее известным средством защиты является Аппаратно-программный комплекс «Аккорд». Аппаратно-программный комплекс Аккорд имеет большое количество модификаций, это наиболее надежное и дорогостоящее изделие, разработано в России ОКБ «САПР» на белорусском рынке представляется фирмой «Марфи».

Аккорд включает три подсистемы: управления доступом, регистрации и учета, обеспечения целостности.

Подсистема управления доступом осуществляет идентификацию, проверку подлинности и контроль допуска субъектов в систему, к внешним устройствам, к файлам, папкам, каталогам.

Подсистема регистрации и учета осуществляет указанные функции в отношении:

- входа (выхода) субъектов доступа в (из) системы;
- запуска (завершения) программ и процессов (заданий, задач);
- доступа программ субъектов к защищаемым файлам, включая их создание и удаление;
- доступа программ субъектов доступа к внешним устройствам ПЭВМ; изменений полномочий субъектов доступа;
- создаваемых защищаемых объектов доступа.

Подсистема обеспечения целостности отвечает за целостность программных средств и обрабатываемой информации.

Лекция 4. Атаки в компьютерных сетях.

4.1. Общие сведения об атаках

Атакой на КС называется действие или последовательность действий нарушителя, которые приводят к реализации угроз, путем использования уязвимостей этой КС. Уязвимости делят на: уязвимости за счет наличия недостатков в аппаратно-программном продукте по вине разработчика; уязвимости, добавленные администратором при настройке КС; уязвимости, внесенные пользователем КС (короткий пароль, игнорирование политики безопасности). Атака состоит из следующих этапов: сбор информации; реализация атаки, завершение атаки.

Сбор информации.

Изучение окружения атакуемой системы (определяется провайдер жертвы, адреса доверенных узлов, трафик, режим работы организации, телефонные номера и т.д.); идентификация топологии сети (определяется количество компьютеров, способ их соединения, организация выхода в глобальную сеть); идентификация узлов (проводится разведка IP-адреса узла, его доступности); идентификация сервисов и портов (определяется наличие установленных сервисов типа Telnet, FTP, Web- сервера и наличие доступа к ним, открытость портов); идентификации ОС (определяется тип ОС); определение роли узла (маршрутизатор, МСЭ, сервер); определение уязвимостей узла (на основе собранной информации определяется наличие уязвимостей).

Реализация атаки.

Реализация атаки заключается в проникновении в систему и установления контроля над ней. Контроль может быть непосредственный, например, через Telnet или с помощью установленной программы.

Завершение атаки.

На этом этапе Злоумышленник убирает следы своей атаки с целью невозможности его идентификации. Для этого используют: подмену адреса источника атаки путем создания пакетов с фальшивыми адресами источника; проводят очистку журнала регистрации событий. Либо атаку проводят с уже взломанных промежуточных серверов или proxy-серверов. Маскируют внедренные программы путем присоединения их к стандартным, либо присвоением им названий, похожих на названия стандартных программ. Изменяют контрольные суммы файлов и папок.

Большинство известных атак можно разбить на следующие группы: удаленное проникновение (атака, в результате которой реализуется удаленное управление компьютером через сеть); локальное проникновение (внедряется

программа, которая управляет компьютером(Get Admin); удаленный отказ в обслуживании (перегрузка потоком сообщений узла, который не в состоянии их переработать); локальный отказ в обслуживании (узел занят обработкой некоторой задачи и все остальные игнорирует(зацикливание)); сетевое сканирование (сеть подвергается запросам программы, анализирующей топологию, доступные сервисы и уязвимости (nmap, Satan)); взлом паролей (запуск программы, подбирающие пароли пользователей (Crack)); анализ протоколов (с помощью анализатора протоколов просматривается трафик. Извлекаются идентификаторы, пароли).

4.2. Технология обнаружения атак

Технология обнаружения атак основывается на: признаках, описывающих нарушения политики безопасности (что); источниках информации, в которых ищутся признаки нарушения политики безопасности (где); методах анализа информации, получаемой из соответствующих источников (как).

Признаками атак являются: повтор определенных событий; неправильные или несоответствующие текущей ситуации команды; признаки работы средств анализа уязвимостей; несоответствующие параметры сетевого трафика; непредвиденные атрибуты; необъяснимые проблемы.

Повтор определенных событий. Злоумышленник, пытаясь осуществить несанкционированное проникновение, вынужден совершать определенные действия несколько раз, т.к. с одного раза он не достигает своей цели. Например, подбор пароля при аутентификации; сканирование портов с целью обнаружения открытых.

Неправильные или несоответствующие текущей ситуации команды. Обнаружение неправильных запросов или ответов, ожидаемых от автоматизированных процессов и программ. Например, в процессе аутентификации почтовых клиентов системы вместо традиционных процедур вдруг обнаружены иные команды, оказалось, что свидетельствует о попытке злоумышленника получить доступ к файлу паролей почтового шлюза.

Признаки работы средств анализа уязвимостей. Имеется ряд средств автоматизированного анализа уязвимостей сети: nmap, Satan, Internet Scanner, которые в определенном порядке обращаются к различным портам с очень небольшим интервалом времени. Такие обращения являются признаками атак.

Несоответствующие параметры сетевого трафика. Например, некорректные параметры входного и выходного трафика (в ЛВС приходят из внешней сети пакеты, имеющие адреса источника, соответствующие диапазону адресов внутренней сети. Из ЛВС выходят пакеты с адресом источника, находящегося во внешней сети. Адрес источника запрещен, адрес источника и

получателя совпадают); некорректные значения параметров различных полей сетевых пакетов (взаимоисключающие флаги); аномалии сетевого трафика (параметры сетевого трафика отличатся от традиционных: коэффициент загрузки, размер пакета, среднее число фрагментированных пакетов, использование нетипичного протокола); непредвиденные атрибуты (запросы пользователей, их действия характеризуются неким типовым профилем, отклонения от него это признак атаки, например, работа в нерабочее врем в выходные, в отпуске; нетипичное местоположения пользователя, нетипичные запросы сервисов и услуг).

Необъяснимые проблемы. Проблемы с программным и аппаратным обеспечением, с системными ресурсами, с производительностью.

Источники информации об атаках являются журналы регистрации событий (ЖРС) или сетевой трафик.

Журналы регистрации событий ведутся рабочими станциями, серверами, межсетевыми экранами (МСЭ), системами обнаружения атак. Типовая запись в таком журнале ведется по следующей форме:

Таблица 4.1.

Дата	Время	Источник (программа, которая регистрирует событие)	Категория (название события: вх., вых., изм.политики доступа к объекту)	Код события	Пользователь (субъект, с которым связано событие: Ad, User, system)	Компьютер (место, на котором произошло событие)

Изучение сетевого трафика позволяет проводить анализ содержания пакетов или последовательностей пакетов.

Примеры обнаружения атак по ЖРС и сетевому трафику.

Обнаружение сканирования портов: Отслеживая записи в ЖРС замечаем, что идет поток запросов из одного адреса через короткие промежутки времени (5-10 запросов в сек.) к портам, номера которых перебираются последовательно (это признак простейшего сканирования). В более сложном сканировании признаки маскируют: увеличивают временные интервалы между запросами и номера портов изменяют по случайному закону.

***Обнаружение подмены адреса источника сообщения*:** Каждому пакету присваивается уникальный идентификатор и если пакеты исходят из одного источника, то очередной пакет получает номер на 1 больше. Если приходят пакеты из разных источников, а их идентификаторы последовательно нарастают, то это свидетельствует о фальшивом адресе источника.

Аналогично можно использовать поле времени жизни. Пакеты, отправленные из различных источников, при приеме в узле имеют одинаковые значения (примерно) оставшегося времени жизни, хотя оно должно быть разным. Следовательно, они отправлены из одного источника.

Обнаружение идентификации типа ОС: Специальной программой формируются пакеты уровня TCP в заголовках, которых используются комбинации флагов, не соответствующие стандартам. По реакции узла на эти пакеты определяется тип ОС. Данная комбинация флагов и является признаком идентификации типа ОС.

Обнаружение троянских программ: При передаче троянской программы идет обращение к портам с вполне определенными номерами. Поэтому если получены пакеты с этими номерами портов, то это свидетельствует о возможном наличии в передаваемых данных троянской программы. Кроме того, троянские программы могут быть распознаны по наличию ключевых слов в поле данных.

Обнаружение атак «Отказ в обслуживании»: обнаружение производится по превышению числа запросов в 1 времени; по совпадению адресов отправителя и получателя; по номерам портов, указанных в пакетах (пересылка пакета с 19 на 17 или 13 на 37 зацикливает атакуемый компьютер).

Для координации деятельности мирового сообщества по защите в сети Интернет создан Координационный центр CERT/CC. Он собирает всю информацию об атаках и дает рекомендации пользователям. Адрес этого центра в Интернете: WWW.cept.org.

4.3. Методы анализа информации при обнаружении атак

4.3.1. Способы обнаружения атак.

Способы обнаружения атак можно разделить на: обнаружение признаков аномального поведения защищаемой системы; обнаружение признаков злоумышленных действий субъектов.

Обнаружение признаков аномального поведения защищаемой системы.

Составляется совокупность признаков нормального (без вмешательства злоумышленника) поведения системы - эталон признаков нормального поведения. Реальное поведение непрерывно или дискретно сравнивается с эталонным, если они не совпадают, то возможно это следствие злоумышленных действий. Таким образом, отклонение реального поведения от эталонного – признак атаки. Структурная схема системы для обнаружения признаков аномального поведения защищаемой системы изображена на рис. 4.1.

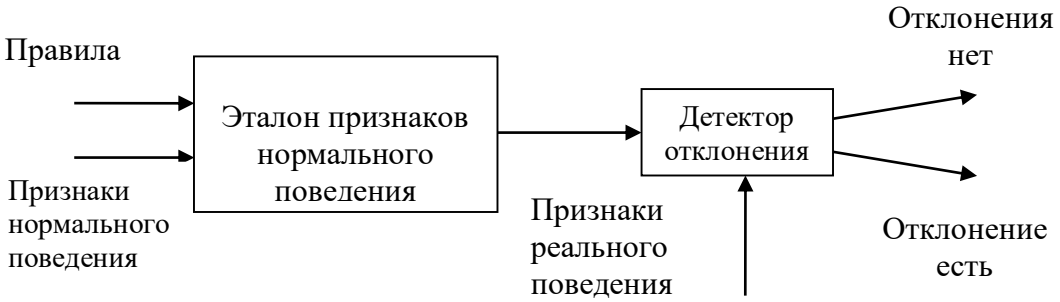


Рис. 4.1. Обнаружение признаков аномального поведения защищаемой системы

При данном способе обнаруживаются любые атаки (в том числе и неизвестные), приводящие к отклонению поведения от нормального. При этом необходимо иметь эталон признаков нормального поведения. Это возможно, если процесс функционирования системы является стабильным (каждый день решаются одни и те же задачи, например, в супермаркете, банке) и описывается некой устойчивой совокупностью признаков. Все изменения в поведении таких систем - плановые, прогнозируемые и могут быть учтены путем корректировки эталона (подключение филиала банка). Режим работы систем жестко регламентируем. Для таких систем целесообразно использовать описанный способ. Например, сотрудники организации используют электронную почту только в рабочее время: $\mathbf{t} \in$ [9-00, 18-00]- эталон; если $\mathbf{t_P}$ =23-15, то имеет место отклонение от эталона.

Обнаружение признаков злоумышленных действий субъектов.

Создается база шаблонов признаков злоумышленных действий. Реальные действия субъекта сравниваются с шаблонами признаков злоумышленных действий. При обнаружении совпадений делается вывод о наличии атаки. Таким образом, совпадение действия субъекта с одним из шаблонов – признак атаки рис. 4.2

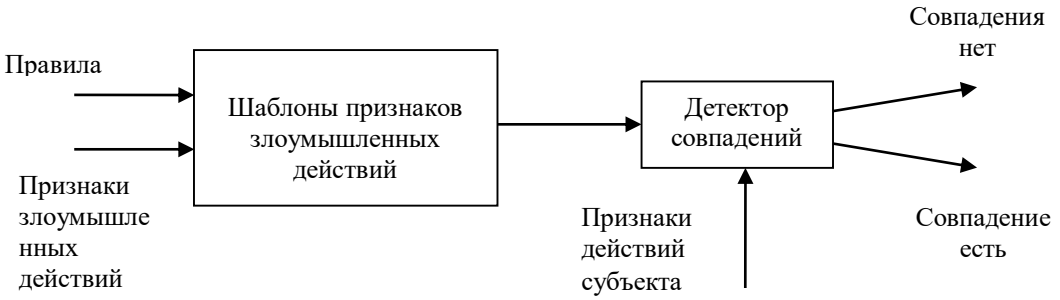


Рис. 4.2. Обнаружение признаков злоумышленных действий субъектов.

Данный способ должен применяться для систем, решающих разнообразные задачи, взаимодействуя с различными узлами, поведение

которых является нестабильным и для которых невозможно составить эталон нормального поведения. Он требует наличия базы шаблонов признаков злоумышленных действий и не пригоден для обнаружения неизвестных атак.

4.3.2. Методы анализа информации при обнаружении атак

Составление эталона признаков нормального поведения системы - сложная задача, т.к. в компьютерной системе протекает большое количество процессов, она взаимодействует с различными пользователями, действия которых трудно формализуются. Аналогичные трудности встречаются и при реализации второго способа обнаружения атак.

Принята следующая классификация признаков (параметров): числовые параметры (размер сообщения, длительность временного интервала); - категориальные параметры (имя файла, команда, ключевое слово); - параметры активности (количество соединений в единицу времени).

Чем больше признаков используется, тем больше шансов обнаружить атаку, но с другой стороны анализ слишком большого количества параметров требует больших вычислительных ресурсов при этом производительность контролируемого узла, объем операционной и дисковой памяти снижаются. Большинство числовых параметров поведения системы носят случайный характер и имеют разброс значений от одного наблюдения к другому. Поэтому при составлении эталона необходимо оперировать с вероятностными характеристиками этих случайных величин (МОЖ, дисперсия, квантиль, закон распределения). Следовательно, при таком подходе задача сравнения эталона с реальным поведением может рассматриваться как задача статистической классификации. Например, как задача проверки статистической гипотезы или задача распознавания образов.

При использовании аппарата проверки статистической гипотезы выдвигается гипотеза (одномерная), что среднее значение эталонного признака $\overline{X}_{Эj}$, равно среднему значению реального признака $\overline{X}_{Pj}$ т.е. H_0: $\overline{X}_{Эj} = \overline{X}_{Pj}$ при альтернативе H_1: $\overline{X}_{Эj} \neq \overline{X}_{Pj}$. Наблюдая реальные значения X_{PJ} и имея решающее правило, гипотеза принимается или отвергается с заданной вероятностью.

Ограничения. Необходимо знать законы распределения величин X_{PJ}, $X_{ЭJ}$. Особенно сложно определить $f(X_{ЭJ}/H_1)$. Для этого необходимо имитировать атаку на систему и определить условную плотность вероятности (т.е. обучить систему обнаружения).

Описанную процедуру следует применять для всех признаков поведения. И если хотя бы по одному из них результат отрицателен, то принимается решение о наличии атаки. При этом существуют ошибки: ложная тревога и

пропуск атаки. Вероятность ошибок тем больше, чем реальные вероятностные характеристики признаков отличаются от гипотетических.

Перспективным способом анализа информации при обнаружении атак можно считать теорию нейронных сетей.

К настоящему времени информационное сообщество накопило большое количество информации о злоумышленных действиях. Известно, что негативные действия сопровождаются определенными признаками. Поскольку в сетях все действия осуществляются посредством генерации битовых потоков (сигнатур), то по многим повторяющимся атакам имеется банк сигнатур (строка символов, определенные команды, последовательность команд). В задачах обнаружения признаков злоумышленных действий эти сигнатуры играют роль шаблонов. Сетевой трафик анализируется на наличие в нем сигнатур атак. Эта задача - детерминированная. Детектор обнаружения ищет совпадение сигнатур трафика с сигнатурами атак. Например, ищет некорректные значения полей в заголовке пакетов. При этом обеспечивается простота реализации, высокая скорость функционирования, отсутствие ложных тревог, однако при этом невозможно обнаружить неизвестные атаки (шаблоны отсутствуют), небольшая модификация атаки делает ее не обнаруживаемой.

Лекция 5. Межсетевые экраны

5.1. Общие сведения

При подключении любой закрытой компьютерной сети к открытым сетям, например, к сети Internet, высокую актуальность приобретают угрозы несанкционированного вторжения в закрытую сеть из открытой, а также угрозы несанкционированного доступа из закрытой сети к ресурсам открытой. Подобный вид угроз характерен также для случая, когда объединяются отдельные сети, ориентированные на обработку конфиденциальной информации совершенно разного уровня секретности. При ограничении доступа этих сетей друг к другу возникают угрозы нарушения установленных ограничений.

Неправомерное вторжение во внутреннюю сеть из внешней может выполняться как с целью несанкционированного использования ресурсов внутренней сети, например, хищения информации, так и с целью нарушения ее работоспособности.

Угрозы несанкционированного доступа во внешнюю сеть из внутренней сети актуальны в случае ограничения разрешенного доступа во внешнюю сеть правилами, установленными в организации. Такое ограничение, что особенно характерно для взаимодействия с открытыми сетями, может понадобиться в следующих случаях: для предотвращения утечки конфиденциальных данных; при запрете доступа, например, в учебных заведениях; к информации нецензурной и нежелательной направленности; в случае запрета служебного доступа к развлекательным компьютерным ресурсам в рабочее время.

Бороться с рассмотренными угрозами безопасности межсетевого взаимодействия средствами универсальных операционных систем не представляется возможным. Универсальная операционная система — это слишком большой и сложный комплекс программ, который, с одной стороны, может содержать внутренние ошибки и недоработки, а с другой — не всегда обеспечивает защиту от ошибок администраторов и пользователей.

Поэтому проблема защиты от несанкционированных действий при взаимодействии с внешними сетями успешно может быть решена только с помощью специализированных программно-аппаратных комплексов, обеспечивающих целостную защиту компьютерной сети от враждебной внешней среды. Такие комплексы называют межсетевыми экранами, брандмауэрами или системами Fire Wall. Межсетевой экран устанавливается на стыке между внутренней и внешней сетями и функции противодействия несанкционированному межсетевому доступу берет на себя.

5.2. Функции межсетевого экранирования

Для противодействия несанкционированному межсетевому доступу брандмауэр должен располагаться между защищаемой сетью организации, являющейся внутренней, и потенциально враждебной внешней сетью (рис.5. 1). При этом все взаимодействия между этими сетями должны осуществляться только через межсетевой экран. Организационно экран входит в состав защищаемой сети.

Межсетевой экран должен учитывать протоколы информационного обмена, положенные в основу функционирования внутренней и внешней сетей. Если эти протоколы отличаются, то брандмауэр должен поддерживать многопротокольный режим работы, обеспечивая протокольное преобразование отличающихся по реализации уровней модели OSI для объединяемых сетей. Чаще всего возникает необходимость в совместной поддержке стеков протоколов SPX/IPX и TCP/IP.

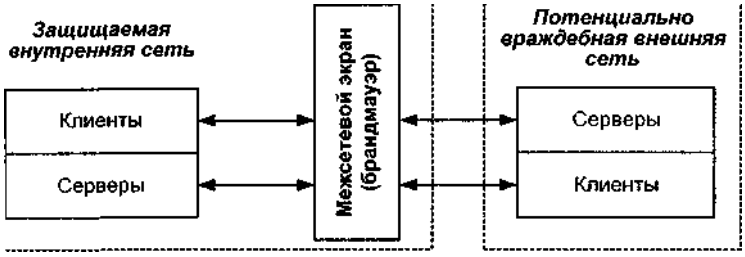


Рис. 5.1. Схема подключения межсетевого экрана

Брандмауэр не является симметричным. Для него отдельно задаются правила, ограничивающие доступ из внутренней сети во внешнюю сеть и наоборот. В общем случае работа межсетевого экрана основана на динамическом выполнении двух групп функций:

- фильтрации проходящих через него информационных потоков;
- посредничества при реализации межсетевых взаимодействий.

В зависимости от типа экрана эти функции могут выполняться с различной полнотой. Простые межсетевые экраны ориентированы на выполнение только одной из данных функций. Комплексные экраны обеспечивают совместное выполнение указанных функций защиты. Собственная защищенность брандмауэра достигается с помощью тех же средств, что и защищенность универсальных систем.

Чтобы эффективно обеспечивать безопасность сети, комплексный брандмауэр обязан управлять всем потоком, проходящим через него, и отслеживать свое состояние. Для принятия управляющих решений по используемым сервисам межсетевой экран должен получать, запоминать, выбирать и обрабатывать информацию, полученную от всех коммуникационных уровней и от других приложений. Недостаточно просто проверять пакеты по отдельности.

Устройство, подобное межсетевому экрану, может использоваться и для защиты отдельного компьютера. В этом случае экран, уже не являющийся межсетевым, устанавливается на защищаемый компьютер. Такой экран, называемый брандмауэром компьютера или системой сетевого экранирования, контролирует весь исходящий и входящий трафик независимо от всех прочих системных защитных средств. При экранировании отдельного компьютера поддерживается доступность сетевых сервисов, но уменьшается или вообще ликвидируется нагрузка, индуцированная внешней активностью. В результате снижается уязвимость внутренних сервисов защищаемого таким образом компьютера, поскольку первоначально сторонний злоумышленник должен преодолеть экран, где защитные средства сконфигурированы особенно тщательно и жестко.

5.2.1. Фильтрация трафика

Фильтрация информационных потоков состоит в их выборочном пропускании через экран, возможно, с выполнением некоторых преобразований и извещением отправителя о том, что его данным в пропуске отказано. Фильтрация осуществляется на основе набора правил, предварительно загруженных в экран и являющихся выражением сетевых аспектов принятой политики безопасности. Поэтому межсетевой экран удобно представлять как последовательность фильтров, обрабатывающих информационный поток. Каждый из фильтров предназначен для интерпретации отдельных правил фильтрации путем выполнения следующих стадий:

- анализа информации по заданным в интерпретируемых правилах критериям, например, по адресам получателя и отправителя или по типу приложения, для которого эта информация предназначена;
- принятия на основе интерпретируемых правил одного из следующих решений не пропустить данные;
- обработать данные от имени получателя и возвратить результат отправителю.

Правила фильтрации могут задавать и дополнительные действия, которые относятся к функциям посредничества, например, преобразование данных, регистрация событий и др. Соответственно правила фильтрации определяют перечень условий, по которым с использованием указанных критериев анализа осуществляется:

- разрешение или запрещение дальнейшей передачи данных;
- выполнение дополнительных защитных функций.

В качестве критериев анализа информационного потока могут использоваться следующие параметры:

- служебные поля пакетов сообщений, содержащие сетевые адреса, идентификаторы, адреса интерфейсов, номера портов и другие значимые данные;

– непосредственное содержимое пакетов сообщений, проверяемое, например, на наличие компьютерных вирусов; внешние характеристики потока информации, например, временные, частотные характеристики, объем данных и т. д.

Используемые критерии анализа зависят от уровней модели OSI, на которых осуществляется фильтрация. В общем случае, чем выше уровень модели OSI, на котором брандмауэр фильтрует пакеты, тем выше и обеспечиваемый им уровень защиты.

5.2.2. Выполнение функций посредничества

Функции посредничества межсетевой экран выполняет с помощью специальных программ, называемых экранирующими агентами или просто программами-посредниками. Данные программы являются резидентными и запрещают непосредственную передачу пакетов сообщений между внешней и внутренней сетью.

При необходимости доступа из внутренней сети во внешнюю сеть или наоборот вначале должно быть установлено логическое соединение с программой-посредником, функционирующей на компьютере экрана. Программа-посредник проверяет допустимость запрошенного межсетевого взаимодействия и при его разрешении сама устанавливает отдельное соединение с требуемым компьютером. Далее обмен информацией между компьютерами внутренней и внешней сети осуществляется через программного посредника, который может выполнять фильтрацию потока сообщений, а также осуществлять другие защитные функции.

Функции фильтрации межсетевой экран может выполнять без применения программ-посредников, обеспечивая прозрачное взаимодействие между внутренней и внешней сетью. Вместе с тем программные посредники могут и не осуществлять фильтрацию потока сообщений.

В общем случае экранирующие агенты, блокируя прозрачную передачу потока сообщений, могут выполнять следующие функции: идентификацию и аутентификацию пользователей; проверку подлинности передаваемых данных; разграничение доступа к ресурсам внутренней сети; разграничение доступа к ресурсам внешней сети; фильтрацию и преобразование потока сообщений, например, динамический поиск вирусов и прозрачное шифрование информации; трансляцию внутренних сетевых адресов для исходящих пакетов сообщений; регистрацию событий, реагирование на задаваемые события, а также анализ зарегистрированной информации и генерацию отчетов; кэширование данных, запрашиваемых из внешней сети.

Идентификация и аутентификация пользователей необходима не только при их доступе из внешней сети во внутреннюю, но и наоборот. Распространенным

способом аутентификации является использование одноразовых паролей. Пароль не должен передаваться в открытом виде через общедоступные коммуникации. Это предотвратит получение несанкционированного доступа путем перехвата сетевых пакетов, что возможно, например, в случае стандартных сервисов типа Telnet. Удобно и надежно также применение цифровых сертификатов, выдаваемых доверительными органами, например, центром распределения ключей. Большинство программ-посредников разрабатываются таким образом, чтобы пользователь аутентифицировался только в начале сеанса работы с межсетевым экраном. После этого от него не требуется дополнительная аутентификация в течение времени, определяемого администратором.

Проверка подлинности получаемых и передаваемых данных необходима не только для аутентификации электронных сообщений, но и мигрирующих программ (Java, ActiveX Controls), по отношению к которым может быть выполнен подлог. Проверка подлинности сообщений и программ заключается в контроле их цифровых подписей. Для этого также могут применяться цифровые сертификаты.

Разграничение доступа к ресурсам внутренней или внешней сети. Способы разграничения к ресурсам внутренней сети ничем не отличаются от способов разграничения, поддерживаемых на уровне операционной системы. При разграничении доступа к ресурсам внешней сети чаще всего используется один из следующих подходов:

- разрешение доступа только по заданным адресам во внешней сети;
- фильтрация запросов на основе обновляемых списков, недопустимых адресов и блокировка поиска информационных ресурсов по нежелательным ключевым словам;
- накопление и обновление администратором санкционированных информационных ресурсов внешней сети в дисковой памяти брандмауэра и полный запрет доступа во внешнюю сеть.

Фильтрация и преобразование потока сообщений выполняется посредником на основе заданного набора правил. Здесь следует различать два вида программ посредников: экранирующие агенты, ориентированные на анализ потока сообщений для определенных видов сервиса, например, FTP, HTTP, Telnet; универсальные экранирующие агенты, обрабатывающие весь поток сообщений, например, агенты, ориентированные на поиск и обезвреживание компьютерных вирусов или прозрачное шифрование данных.

Программный посредник анализирует поступающие к нему пакеты данных, и если какой-либо объект не соответствует заданным критериям, то посредник либо блокирует его дальнейшее продвижение, либо выполняет соответствующие

преобразования, например, обезвреживание обнаруженных компьютерных вирусов. При анализе содержимого пакетов важно, чтобы экранирующий агент мог автоматически распаковывать проходящие файловые архивы.

Брандмауэры с посредниками позволяют также организовывать защищенные виртуальные сети (Virtual Private Network — VPN), например, безопасно объединить несколько локальных сетей, подключенных к Internet, в одну виртуальную сеть. VPN обеспечивают прозрачное для пользователей соединение локальных сетей, сохраняя секретность и целостность передаваемой информации путем ее динамического шифрования. При передаче по Internet возможно шифрование не только данных пользователей, но и служебной информации — конечных сетевых адресов, номеров портов и т. д.

Трансляция внутренних сетевых адресов. Данная функция реализуется по отношению ко всем пакетам, следующим из внутренней сети во внешнюю. Для этих пакетов посредник выполняет автоматическое преобразование IP-адресов компьютеров-отправителей в один "надежный" IP-адрес, ассоциируемый с брандмауэром, из которого передаются все исходящие пакеты. В результате все исходящие из внутренней сети пакеты оказываются отправленными межсетевым экраном, что исключает прямой контакт между авторизованной внутренней сетью и являющейся потенциально опасной внешней сетью. IP-адрес брандмауэра становится единственным активным IP-адресом, который попадает во внешнюю сеть.

При таком подходе топология внутренней сети скрыта от внешних пользователей, что усложняет задачу несанкционированного доступа. Кроме повышения безопасности трансляция адресов позволяет иметь внутри сети собственную систему адресации, не согласованную с адресацией во внешней сети, например, в сети Internet. Это эффективно решает проблему расширения адресного пространства внутренней сети и дефицита адресов внешней сети.

Регистрация событий, реагирование на задаваемые события, а также анализ зарегистрированной информации и составление отчетов. В качестве обязательной реакции на обнаружение попыток выполнения несанкционированных действий должно быть определено уведомление администратора, т. е. выдача предупредительных сигналов. Многие межсетевые экраны содержат мощную систему регистрации, сбора и анализа статистики. Учет может вестись по адресам клиента и сервера, идентификаторам пользователей, времени сеансов, времени соединений, количеству переданных/принятых данных, действиям администратора и пользователей. Системы учета позволяют произвести анализ статистики и предоставляют администраторам подробные отчеты. За счет использования специальных протоколов посредники могут выполнить удаленное оповещение об определенных событиях в режиме реального времени.

Кэширование данных, запрашиваемых из внешней сети. При доступе пользователей внутренней сети к информационным ресурсам внешней сети вся информация накапливается на пространстве жесткого диска брандмауэра, называемого в этом случае proxy-сервером. Поэтому если при очередном запросе нужная информация окажется на proxy-сервере, то посредник предоставляет ее без обращения к внешней сети, что существенно ускоряет доступ. Администратору следует позаботиться только о периодическом обновлении содержимого proxy-сервера. За счет использования специальных протоколов посредники могут выполнить удаленное оповещение об определенных событиях в режиме реального времени.

Кэширование данных, запрашиваемых из внешней сети. При доступе пользователей внутренней сети к информационным ресурсам внешней сети вся информация накапливается на пространстве жесткого диска брандмауэра, называемого в этом случае proxy-сервером. Поэтому если при очередном запросе нужная информация окажется на proxy-сервере, то посредник предоставляет ее без обращения к внешней сети, что существенно ускоряет доступ. Администратору следует позаботиться только о периодическом обновлении содержимого proxy-сервера. Функция кэширования успешно может использоваться для ограничения доступа к информационным ресурсам внешней сети. В этом случае все санкционированные информационные ресурсы внешней сети накапливаются и обновляются администратором на proxy-сервере. Пользователям внутренней сети разрешается доступ только к информационным ресурсам proxy-сервера, а непосредственный доступ к ресурсам внешней сети запрещается. Экранирующие агенты намного надежнее обычных фильтров и обеспечивают большую степень защиты. Однако они снижают производительность обмена данными между внутренней и внешней сетями и не обладают той степенью прозрачности для приложений и конечных пользователей, которая характерна для простых фильтров.

5.2.3. Особенности межсетевого экранирования на различных уровнях модели OSI

Брандмауэры поддерживают безопасность межсетевого взаимодействия на различных уровнях модели OSI. При этом функции защиты, выполняемые на разных уровнях эталонной модели, существенно отличаются друг от друга. Поэтому комплексный межсетевой экран удобно представить в виде совокупности неделимых экранов, каждый из которых ориентирован на отдельный уровень модели OSI. Чаще всего комплексный экран функционирует на сетевом, сеансовом и прикладном уровнях эталонной модели. Соответственно различают такие неделимые брандмауэры (рис. 5.2), как экранирующий маршрутизатор, экранирующий

транспорт (шлюз сеансового уровня), а также экранирующий шлюз (шлюз прикладного уровня).

Учитывая, что используемые в сетях протоколы (TCP/IP, SPX/IPX) не однозначно соответствуют модели OSI, то экраны перечисленных типов при выполнении своих функций могут охватывать и соседние уровни эталонной модели. Например, прикладной экран может осуществлять автоматическое зашифровывание сообщений при их передаче во внешнюю сеть, а также автоматическое расшифровывание криптографически закрытых принимаемых данных. В этом случае такой экран функционирует не только на прикладном уровне модели OSI, но и на уровне представления. Шлюз сеансового уровня при своем функционировании охватывает транспортный и сетевой уровни модели OSI. Экранирующий маршрутизатор при анализе пакетов сообщений проверяет их заголовки не только сетевого, но и транспортного уровня.

Межсетевые экраны каждого из типов имеют свои достоинства и недостатки. Многие из используемых брандмауэров являются либо прикладными шлюзами, либо экранирующими маршрутизаторами, не поддерживая полную безопасность межсетевого взаимодействия. Надежную же защиту обеспечивают только комплексные межсетевые экраны, каждый из которых объединяет экранирующий маршрутизатор, шлюз сеансового уровня, а также прикладной шлюз.

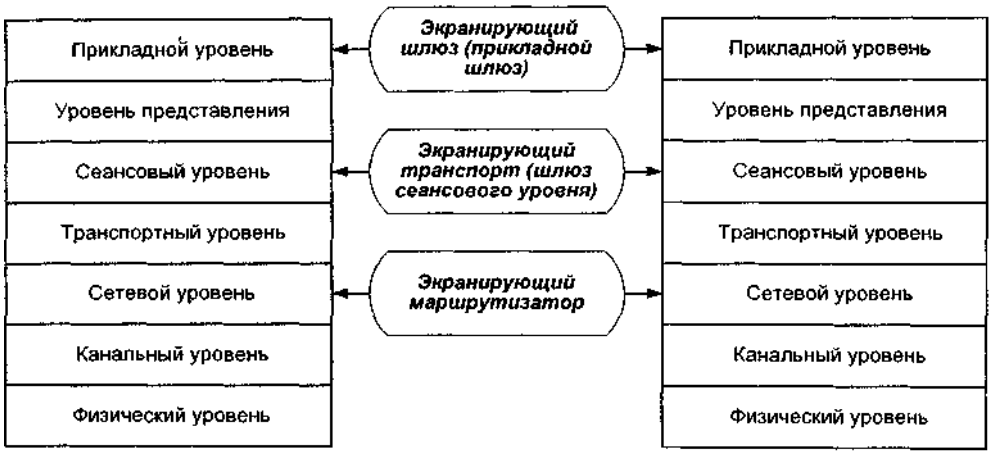


Рис. 5.2. Типы межсетевых экранов, функционирующих на отдельных уровнях модели OSI.

5.3. Экранирующий маршрутизатор

Экранирующий маршрутизатор, называемый еще пакетным фильтром, предназначен для фильтрации пакетов сообщений и обеспечивает прозрачное взаимодействие между внутренней и внешней сетями. Он функционирует на сетевом уровне модели OSI, но для выполнения своих отдельных функций может охватывать и транспортный уровень эталонной модели. Решение о том, пропустить или отбраковать данные, принимается для каждого пакета независимо на основе

заданных правил фильтрации. Для принятия решения анализируются заголовки пакетов сетевого и транспортного уровней. В качестве анализируемых полей IP- и TCP (UTP) - заголовков каждого пакета выступают: адрес отправителя; адрес получателя; тип пакета; флаг фрагментации пакета; номер порта источника; номер порта получателя.

Адреса отправителя и получателя являются IP-адресами. Эти адреса заполняются при формировании пакета и остаются неизменными при передаче его по сети.

Поле типа пакета содержит код протокола ICMP, соответствующего сетевому уровню, либо код протокола транспортного уровня (TCP или UDP), к которому относится анализируемый IP-пакет.

Флаг фрагментации пакета определяет наличие или отсутствие фрагментации IP-пакетов. Если флаг фрагментации для анализируемого пакета установлен, то данный пакет является подпакетом фрагментированного IP-пакета.

Номера портов источника и получателя добавляются драйвером TCP или UDP к каждому отправляемому пакету сообщения и однозначно идентифицируют приложение-отправитель, а также приложение, для которого предназначен этот пакет. Например, при использовании протокола передачи файлов FTP реализация данного протокола на сервере по умолчанию получает номер TCP-порта 21. Каждый Telnet-сервер по умолчанию имеет TCP-порт 23. Для возможности фильтрации пакетов по номерам портов необходимо знание принятых в сети соглашений относительно выделения номеров портов протоколам высокого уровня.

При обработке каждого пакета экранирующий маршрутизатор последовательно просматривает заданную таблицу правил, пока не найдет правила, с которым согласуется полная ассоциация пакета. Здесь под ассоциацией понимается совокупность параметров, указанных в заголовках данного пакета. Если экранирующий маршрутизатор получил пакет, не соответствующий ни одному из табличных правил, он применяет правило, заданное по умолчанию. Из соображений безопасности это правило обычно указывает на необходимость отбраковки всех пакетов, не удовлетворяющих ни одному из других правил.

В качестве пакетного фильтра может использоваться как обычный маршрутизатор, так и работающая на сервере программа, сконфигурированные таким образом, чтобы фильтровать входящие и исходящие пакеты. Современные маршрутизаторы, например, маршрутизирующие устройства компаний Cisco,

позволяют связывать с каждым портом несколько десятков правил и фильтровать пакеты как на входе, так и на выходе.

К достоинствам экранирующих маршрутизаторов относятся: простота самого экрана, а также процедур его конфигурирования и установки; прозрачность для программных приложений и минимальное влияние на производительность сети; низкая стоимость, обусловленная тем, что любой маршрутизатор в той или иной степени представляет возможность фильтрации пакетов.

Однако экранирующие маршрутизаторы не обеспечивают высокой степени безопасности, так как проверяют только заголовки пакетов и не поддерживают многие необходимые функции защиты, например, аутентификацию конечных узлов, криптографическое закрытие пакетов сообщений, а также проверку их целостности и подлинности. Экранирующие маршрутизаторы уязвимы для таких распространенных сетевых атак, как подделка исходных адресов и несанкционированное изменение содержимого пакетов сообщений. "Обмануть" межсетевые экраны данного типа не составляет труда достаточно сформировать заголовки пакетов, которые удовлетворяют разрешающим правилам фильтрации.

5.4. Шлюз сеансового уровня

Шлюз сеансового уровня, называемый еще экранирующим транспортом, предназначен для контроля виртуальных соединений и трансляции IP-адресов при взаимодействии с внешней сетью. Он функционирует на сеансовом уровне модели OSI, охватывая в процессе своей работы также транспортный и сетевой уровни эталонной модели. Защитные функции экранирующего транспорта относятся к функциям посредничества.

Контроль виртуальных соединений заключается в контроле квитирования связи, а также контроле передачи информации по установленным виртуальным каналам.

При контроле квитирования связи шлюз сеансового уровня следит за установлением виртуального соединения между рабочей станцией внутренней сети и компьютером внешней сети, определяя, является ли запрашиваемый сеанс связи допустимым. Такой контроль основывается на информации, содержащейся в заголовках пакетов сеансового уровня протокола TCP. Однако если пакетный фильтр при анализе TCP-заголовков проверяет только номера портов источника и получателя, то экранирующий транспорт анализирует другие поля, относящиеся к процессу квитирования связи.

Чтобы определить, является ли запрос на сеанс связи допустимым, шлюз сеансового уровня выполняет следующие действия. Когда рабочая станция (клиент) запрашивает связь с внешней сетью, шлюз принимает этот запрос, проверяя, удовлетворяет ли он базовым критериям фильтрации, например,

может ли DNS-сервер определить IP-адрес клиента и ассоциированное с ним имя. Затем, действуя от имени клиента, шлюз устанавливает соединение с компьютером внешней сети и следит за выполнением процедуры квитирования связи по протоколу TCP.

5.5. Прикладной шлюз

Прикладной шлюз, называемый также экранирующим шлюзом, функционирует на прикладном уровне модели OSI, охватывая также уровень представления, и обеспечивает наиболее надежную защиту межсетевых взаимодействий. Защитные функции прикладного шлюза, как и экранирующего транспорта, относятся к функциям посредничества. Однако прикладной шлюз, в отличие от шлюза сеансового уровня, может выполнять существенно большее количество функций защиты, к которым относятся следующие:

– идентификация и аутентификация пользователей при попытке установления соединений через брандмауэр;

– проверка подлинности информации, передаваемой через шлюз;

– разграничение доступа к ресурсам внутренней и внешней сетей;

– фильтрация и преобразование потока сообщений, например, динамический поиск вирусов и прозрачное шифрование информации;

– регистрация событий, реагирование на задаваемые события, а также анализ зарегистрированной информации и генерация отчетов;

– кэширование данных, запрашиваемых из внешней сети.

Учитывая, что функции прикладного шлюза относятся к функциям посредничества, он представляет собой универсальный компьютер, на котором функционируют программные посредники (экранирующие агенты) — по одному для каждого обслуживаемого прикладного протокола (HTTP, FTP, SMTP, NNTP и др).

Посредник каждой службы TCP/IP ориентирован на обработку сообщений и выполнение функций защиты, относящихся именно к этой службе. Так же, как и шлюз сеансового уровня, прикладной шлюз перехватывает с помощью соответствующих экранирующих агентов входящие и исходящие пакеты, копирует и перенаправляет информацию через шлюз, и функционирует в качестве сервера-посредника, исключая прямые соединения между внутренней и внешней сетью. Однако посредники, используемые прикладным шлюзом, имеют важные отличия от канальных посредников шлюзов сеансового уровня. Во-первых, посредники прикладного шлюза связаны с конкретными приложениями (программными серверами), а во-вторых, они могут фильтровать поток сообщений на прикладном уровне модели OSI.

Прикладные шлюзы используют в качестве посредников специально разработанные для этой цели программные серверы конкретных служб TCP/IP — серверы HTTP, FTP, SMTP, NNTP и др. Эти программные серверы функционируют на брандмауэре в резидентном режиме и реализуют функции защиты, относящиеся к соответствующим службам TCP/IP. Трафик UDP обслуживается специальным транслятором содержимого UDP-пакетов.

Как и в случае шлюза сеансового уровня, для связи между рабочей станцией внутренней сети и компьютером внешней сети соответствующий посредник прикладного шлюза образует два соединения: от рабочей станции до брандмауэра и от брандмауэра до места назначения. Но, в отличие от канальных посредников, посредники прикладного шлюза пропускают только пакеты, сгенерированные теми приложениями, которые им поручено обслуживать. Например, программа-посредник службы HTTP может обрабатывать лишь трафик, генерируемый этой службой. Если в сети работает прикладной шлюз, то входящие и исходящие пакеты могут передаваться лишь для тех служб, для которых имеются соответствующие посредники. Так, если прикладной шлюз использует только программы-посредники HTTP, FTP и Telnet, то он будет обрабатывать лишь пакеты, относящиеся к этим службам, блокируя при этом пакеты всех остальных служб.

Фильтрация потоков сообщений реализуется прикладными шлюзами на прикладном уровне модели OSI. Соответственно посредники прикладного шлюза, в отличие от канальных посредников, обеспечивают проверку содержимого обрабатываемых пакетов. Они могут фильтровать отдельные виды команд или информации в сообщениях протоколов прикладного уровня, которые им поручено обслуживать. Например, для службы FTP возможно динамическое обезвреживание компьютерных вирусов в копируемых из внешней сети файлах. Кроме того, посредник данной службы может быть сконфигурирован таким образом, чтобы предотвращать использование клиентами команды PUT, предназначенной для записи файлов на FTP-сервер. Такое ограничение уменьшает риск случайного повреждения хранящейся на FTP-сервере информации и снижает вероятность заполнения его гигабайтами ненужных данных.

При настройке прикладного шлюза и описании правил фильтрации сообщений используются такие параметры, как название сервиса, допустимый временной диапазон его использования, ограничения на содержимое сообщений, связанных с данным сервисом, компьютеры, с которых можно пользоваться сервисом, идентификаторы пользователей, схемы аутентификации и др.

5.6. Установка и конфигурирование межсетевых экранов

Для эффективной защиты межсетевого взаимодействия система FireWall должна быть правильно установлена и сконфигурирована. Данный процесс осуществляется путем последовательного выполнения следующих этапов: разработки политики межсетевого взаимодействия; определения схемы подключения, а также непосредственного подключения межсетевого экрана; настройки параметров функционирования брандмауэра.

Перечисленные этапы отражают системный подход к установке любого программно-аппаратного средства, предполагающий, начиная с анализа, последовательную детализацию решения стоящей задачи.

5.6.1. Разработка политики межсетевого взаимодействия

Политика межсетевого взаимодействия является той частью политики безопасности в организации, которая определяет требования к безопасности информационного обмена с внешним миром. Данные требования обязательно должны отражать два аспекта: политику доступа к сетевым сервисам; политику работы межсетевого экрана.

Политика доступа к сетевым сервисам определяет правила предоставления, а также использования всех возможных сервисов защищаемой компьютерной сети. Соответственно в рамках данной политики должны быть заданы все сервисы, предоставляемые через сетевой экран, и допустимые адреса клиентов для каждого сервиса. Кроме того, должны быть указаны правила для пользователей, описывающие, когда и какие пользователи каким сервисом и на каком компьютере могут воспользоваться. Отдельно определяются правила аутентификации пользователей и компьютеров, а также условия работы пользователей вне локальной сети организации.

Политика работы межсетевого экрана задает базовый принцип управления межсетевым взаимодействием, положенный в основу функционирования брандмауэра. Может быть выбран один из двух таких принципов:

- запрещено все, что явно не разрешено;
- разрешено все, что явно не запрещено.

В зависимости от выбора, решение может быть принято, как в пользу безопасности в ущерб удобству использования сетевых сервисов, так и наоборот. В первом случае межсетевой экран должен быть сконфигурирован таким образом, чтобы блокировать любые явно неразрешенные межсетевые взаимодействия. Учитывая, что такой подход позволяет адекватно реализовать принцип минимизации привилегий, он, с точки зрения безопасности, является лучшим. Здесь администратор не сможет по забывчивости оставить

разрешенными какие-либо полномочия, так как по умолчанию они будут запрещены. Доступные лишние сервисы могут быть использованы во вред безопасности, что особенно характерно для закрытого и сложного программного обеспечения, в котором могут быть различные ошибки и некорректности. Принцип "запрещено все, что явно не разрешено", в сущности, является признанием факта, что незнание может причинить вред.

При выборе принципа "разрешено все, что явно не запрещено" межсетевой экран настраивается таким образом, чтобы блокировать только явно запрещенные межсетевые взаимодействия. В этом случае повышается удобство использования сетевых сервисов со стороны пользователей, но снижается безопасность межсетевого взаимодействия. Администратор может учесть не все действия, которые запрещены пользователям. Ему приходится работать в режиме реагирования, предсказывая и запрещая те межсетевые взаимодействия, которые отрицательно воздействуют на безопасность сети.

5.6.2. Определение схемы подключения межсетевого экрана

Для подключения межсетевых экранов могут использоваться различные схемы, которые зависят от условий функционирования, а также количества сетевых интерфейсов брандмауэра.

Брандмауэры с одним сетевым интерфейсом (рис. 5.3) не достаточно эффективны как с точки зрения безопасности, так и с позиций удобства конфигурирования. Они физически не разграничивают внутреннюю и внешнюю сети, а соответственно не могут обеспечивать надежную защиту межсетевых взаимодействий. Настройка таких межсетевых экранов, а также связанных с ними маршрутизаторов представляет собой довольно сложную задачу, цена решения которой превышает стоимость замены брандмауэра с одним сетевым интерфейсом на брандмауэр с двумя или тремя сетевыми интерфейсами. Поэтому рассмотрим лишь схемы подключения межсетевых экранов с двумя и тремя сетевыми интерфейсами. При этом защищаемую локальную сеть будем рассматривать как совокупность закрытой и открытой подсетей. Здесь под открытой подсетью понимается подсеть, доступ к которой со стороны потенциально враждебной внешней сети может быть полностью или частично открыт. В открытую подсеть могут, например, входить общедоступные WWW-, FTP- и SMTP-серверы, а также терминальный сервер с модемным пулом.

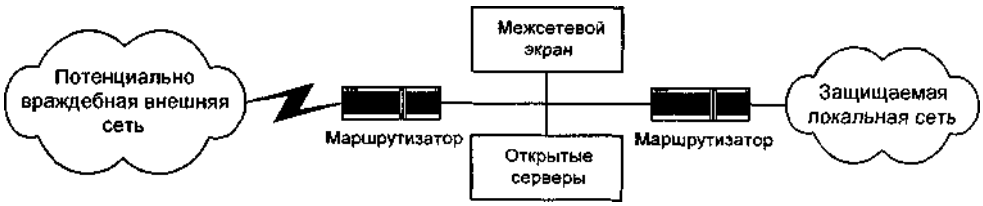


Рис. 5.3. Защита локальной сети брандмауэром с одним сетевым интерфейсом

Среди всего множества возможных схем подключения брандмауэров типовыми являются следующие: схема единой защиты локальной сети; схема *с* защищаемой закрытой и не защищаемой открытой подсетями; схема с раздельной защитой закрытой и открытой подсетей.

Схема единой защиты локальной сети является наиболее простым решением (рис. 5.4), при котором брандмауэр целиком экранирует локальную сеть от потенциально враждебной внешней сети. Между маршрутизатором и брандмауэром имеется только один путь, по которому идет весь трафик. Обычно маршрутизатор настраивается таким образом, что брандмауэр является единственной видимой снаружи машиной. Открытые серверы, входящие в локальную сеть, также будут защищены межсетевым экраном. Однако объединение серверов, доступных из внешней сети, вместе с другими ресурсами защищаемой локальной сети существенно снижает безопасность межсетевых взаимодействий. Поэтому данную схему подключения брандмауэра можно использовать лишь при отсутствии в локальной сети открытых серверов или, когда имеющиеся открытые серверы делаются доступными из внешней сети только для ограниченного числа пользователей, которым можно доверять.

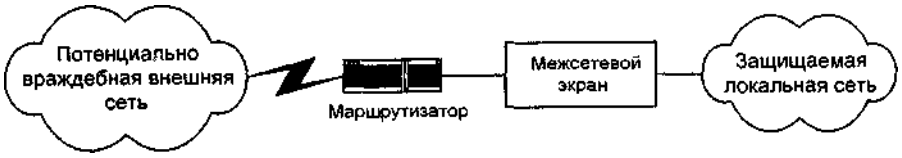


Рис. 5.4. Схема единой защиты локальной сети.

При наличии в составе локальной сети общедоступных открытых серверов их целесообразно вынести как открытую подсеть до межсетевого экрана (рис. 5.5). Данный способ обладает более высокой защищенностью закрытой части локальной сети, но обеспечивает пониженную безопасность открытых серверов, расположенных до межсетевого экрана. Некоторые брандмауэры позволяют разместить эти серверы на себе. Но такое решение не является лучшим с точки зрения загрузки компьютера и безопасности самого брандмауэра. Учитывая вышесказанное, можно сделать вывод, что схему подключения брандмауэра с защищаемой закрытой подсетью и не защищаемой открытой подсетью

целесообразно использовать лишь при невысоких требованиях по безопасности к открытой подсети.

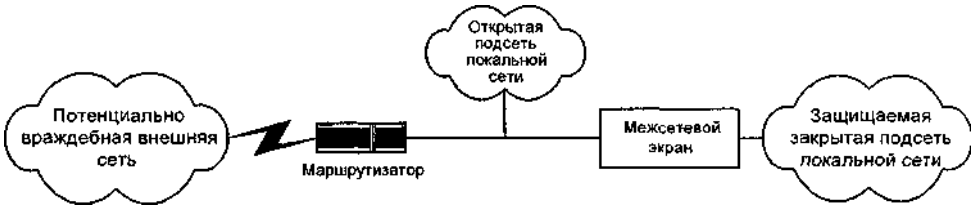


Рис. 5.5. Схема с защищаемой закрытой и не защищаемой открытой подсетями

В случае же, когда к безопасности открытых серверов предъявляются повышенные требования, то необходимо использовать схему с раздельной защитой закрытой и открытой подсетей. Такая схема может быть построена на основе одного брандмауэра с тремя сетевыми интерфейсами (рис. 5.6) или на основе двух брандмауэров с двумя сетевыми интерфейсами (рис. 5.7). В обоих случаях доступ к открытой и закрытой подсетям локальной сети возможен только через межсетевой экран. При этом доступ к открытой подсети не позволяет осуществить доступ к закрытой подсети.

Из последних двух схем большую степень безопасности межсетевых взаимодействий обеспечивает схема с двумя брандмауэрами, каждый из которых образует отдельный эшелон защиты закрытой подсети. Защищаемая открытая подсеть здесь выступает в качестве экранирующей подсети. Обычно экранирующая подсеть конфигурируется таким образом, чтобы обеспечить доступ к компьютерам подсети как из потенциально враждебной внешней сети, так и из закрытой подсети локальной сети. Однако прямой обмен информационными пакетами между внешней сетью и закрытой подсетью невозможен.

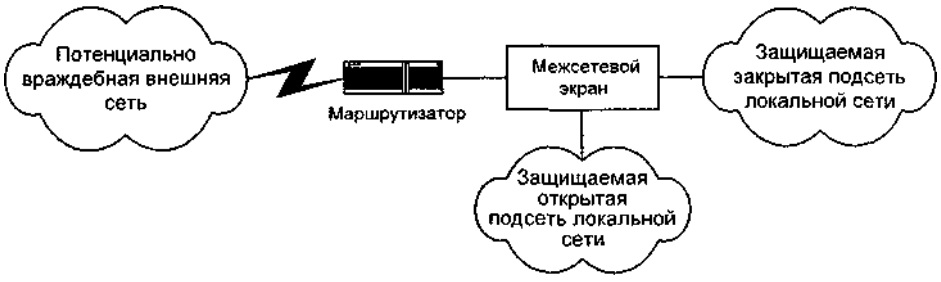


Рис. 5.6. Схема с раздельной защитой закрытой и открытой подсетей на основе одного брандмауэра с тремя сетевыми интерфейсами.

5.7. Настройка параметров функционирования межсетевого экрана

Межсетевой экран представляет собой программно-аппаратный комплекс защиты, состоящий из компьютера, а также функционирующих на нем операционной системы (ОС) и специального программного обеспечения. Следует отметить, что это специальное программное обеспечение часто также называют брандмауэром.

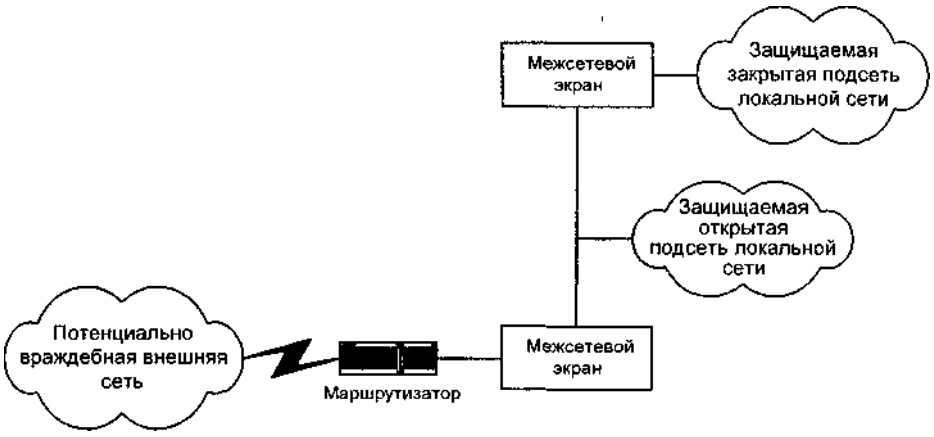


Рис. 5.7. Схема с раздельной защитой закрытой и открытой подсетей на основе двух брандмауэров с двумя сетевыми интерфейсами

Компьютер брандмауэра должен быть достаточно мощным и физически защищенным, например, находиться в специально отведенном и охраняемом помещении. Кроме того, он должен иметь средства защиты от загрузки ОС с несанкционированного носителя. После установки на компьютер брандмауэра выбранной операционной системы, ее конфигурирования, а также инсталляции специального программного обеспечения можно приступать к настройке параметров функционирования всего межсетевого экрана. Этот процесс включает следующие этапы:

– выработку правил работы межсетевого экрана в соответствии с разработанной политикой межсетевого взаимодействия и описание правил в интерфейсе брандмауэра;

– проверку заданных правил на непротиворечивость;

– проверку соответствия параметров настройки брандмауэра разработанной политике межсетевого взаимодействия.

Формируемая на первом этапе база правил работы межсетевого экрана представляет собой формализованное отражение разработанной политикой межсетевого взаимодействия. Компонентами правил являются защищаемые объекты, пользователи и сервисы. В число защищаемых объектов могут входить обычные

компьютеры с одним сетевым интерфейсом, шлюзы (компьютеры с несколькими сетевыми интерфейсами), маршрутизаторы, сети, области управления. Защищаемые объекты могут объединяться в группы. Каждый объект имеет набор атрибутов, таких как сетевой адрес, маска подсети и т. п. Часть этих атрибутов следует задать вручную, остальные извлекаются автоматически из информационных баз, например, NIS/NIS+, SNMP MIB, DNS. Следует обратить внимание на необходимость полного описания объектов, так как убедиться в корректности заданных правил экранирования можно только тогда, когда определены все сетевые интерфейсы шлюзов и маршрутизаторов. Подобную информацию можно получить автоматически от SNMP-агентов. При описании правил работы межсетевого экрана пользователи наделяются входными именами и объединяются в группы. Для пользователей указываются допустимые исходные и целевые сетевые адреса, диапазон дат и времени работы, а также схемы и порядок аутентификации. Определение набора используемых сервисов выполняется на основе встроенной в дистрибутив брандмауэра базы данных, имеющей значительный набор TCP/IP сервисов. Нестандартные сервисы могут задаваться вручную с помощью специальных атрибутов. Прежде чем указывать сервис при задании правил, необходимо определить его свойства. Современные брандмауэры содержат предварительно подготовленные определения всех стандартных TCP/IP-сервисов, разбитых на четыре категории — TCP, UDP, RPC, ICMP.

Лекция 6. Виртуальные защищенные сети

6.1. Принципы построения

Распределенные корпоративные сети могут создаваться на базе отдельных компьютеров или локальных вычислительных сетей (ЛВС) посредством соединения их через специально проложенные каналы связи, через каналы связи общего пользования и через открытые компьютерные сети типа Интернет.

Корпоративные сети на базе Интернет наиболее привлекательны, т.к. обладают рядом преимуществ: относительно малой стоимостью; высокой пропускной способностью; высокой масштабируемостью (размеры сети не ограничиваются каналами передачи данных). Но при этих преимуществах имеется серьезная проблема – обеспечение безопасности информации. В этих сетях необходимо защищать информацию в процессе передачи ее по открытым каналам связи, а также обеспечивать защиту ЛВС и отдельных компьютеров от несанкционированного допуска со стороны внешней среды (пользователей Интернет).

Защита информации при передаче ее по открытым каналам связи требует: аутентификации взаимодействующих сторон; шифрования информации; подтверждения подлинности и целостности доставленной информации; защите от повтора, задержки и удаления сообщений; защите от отрицательных фактов отправления и приема сообщений.

ЛВС и отдельные компьютеры, объединенные через открытую глобальную сеть в единую компьютерную сеть, обеспечивающую защищенность передаваемой и хранимой информации, называются виртуальной частной сетью (Vitual Private Network-VPN). Открытая глобальная сеть может быть основой огромного числа виртуальных частных сетей (рис.6.1).

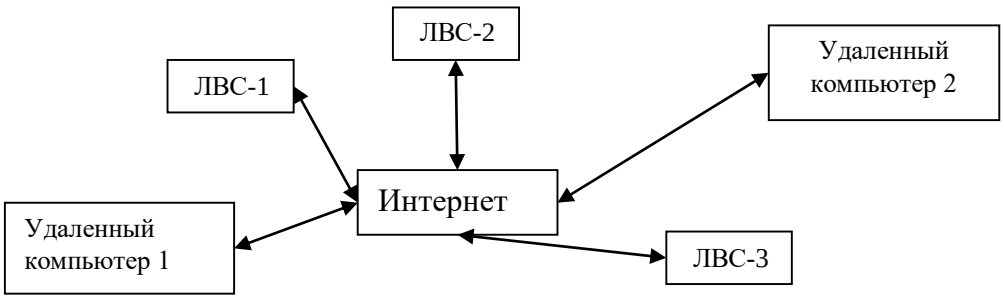


Рис. 6.1. Пример VPN.

Защита информации при передаче по открытым каналам основана на создании защищенных виртуальных каналов связи, крипто защищенных туннелей или туннелей VPN. Каждый туннель VPN – соединение, проведенное

через открытую сеть, по которому передаются криптографически защищенные пакеты сообщений (рис.6.2, маршрут ЛВС-1—1—2—3—9—8—ЛВС-2).

Защищенные виртуальные каналы могут прокладываться (рис.6.3): от каждого компьютера ЛВС-1 до каждого компьютера ЛВС-2, если внутри ЛВС нужно также обеспечит защиту; от пограничного маршрутизатора или МСЭ ЛВС-1 до пограничного маршрутизатора или МСЭ ЛВС-2; от провайдера Интернет ЛВС-1 до провайдера Интернет ЛВС-2(если можно быть уверенным, в том, что в проводных каналах опасность несанкционированного доступа гораздо меньше, чем в каналах с разделением пакетов).

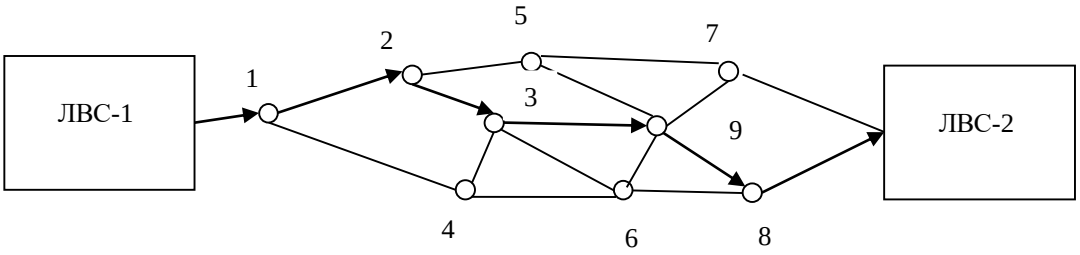


Рис.6.2. Образование туннеля VPN

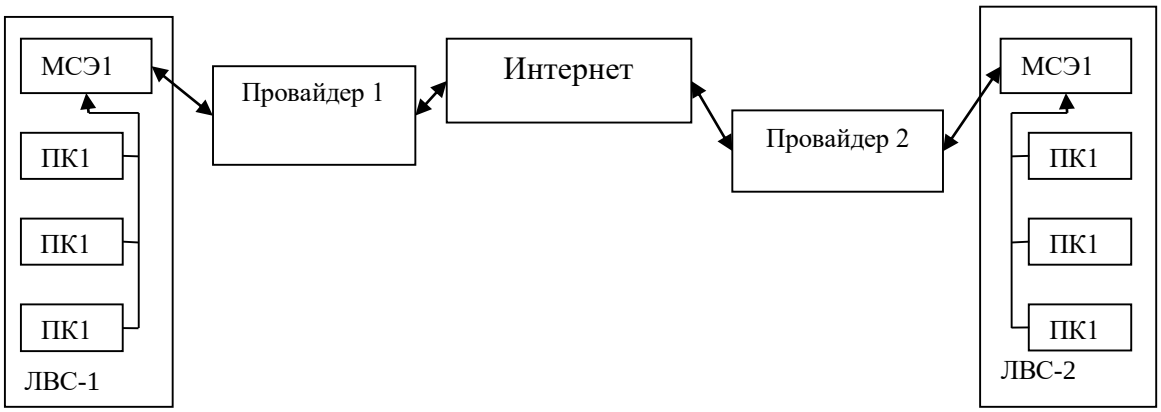


Рис.6.3. Возможные каналы VPN.

Создание защищенного туннеля выполняют компоненты виртуальной сети, функционирующие в узлах, между которыми создается туннель. Эти узлы называются инициатор и терминатор туннеля. Инициатор инкапсулирует (встраивает) пакеты в новый пакет, содержащий в качестве адреса отправителя – адрес инициатора, адреса получателя – адрес терминатора; вложенный пакет полностью шифруется и подписывается ЭЦП. Терминатор получив пакет извлекает из него зашифрованный пакет, расшифровывает его и отправляет конечному адресату в ЛВС. Инициатор и терминатор должны использовать

одинаковые крипто протоколы и поддерживать протокол безопасного распределения ключей.

6.2. Протоколы VPN-сетей

Протоколы VPN появились сравнительно недавно. Они являются свободными для распространения и реализации. Для независимости от прикладных протоколов и приложений протоколы VPN работают на одном уровне из более низких уровней: канальном, сетевом и сеансовом.

Канальному уровню соответствуют протоколы PPTP, L2F, L2TP; сетевому - IPSec, SKIP; сеансовому - SSL/TLS, SOCKS. Чем ниже уровень протокола, тем он «прозрачней» для пользователя. Однако набор услуг безопасности при этом снижается. В VPN сетях криптозащита может одновременно выполняться на всех уровнях модели, однако при этом снижается скорость преобразования. Поэтому практически шифрование используется только на одном уровне.

6.2.1. Канальный уровень

Протокол PPTP (Point – to – Point Tunneling Protocol) представляет собой расширенный PPP. В этом протоколе конкретные алгоритмы шифрования и аутентификации не установлены. Клиенты удаленного доступа в операционных системах Windows поставляются с версией DES и называются MPPE (Microsoft Point – to – Point Encryption).

Протокол L2F (Layer – 2 Forwarding) позволяет использовать для удаленного доступа к провайдеру не только PPP, но и другие протоколы (SLTP), а для переноса по сети Интернет не только IP, как PPTP. В нем также не установлены конкретные алгоритмы шифрования и аутентификации. Протокол L2F является компонентом операционной системы IOS компании Cisco, которая устанавливается во все ее устройства межсетевого взаимодействия. С 1996 года эти протоколы объединены и названы протоколами туннелирования второго уровня L2TP (Layer – 2 Tunneling Protocol). Этот протокол поддерживают Microsoft, Cisco, 3Com. Протокол L2TP работает независимо от протоколов сетевого уровня, на которых функционируют различные ЛВС IP, IPX, NetBEUI. Пакеты этих протоколов шифруются, подписываются и инкапсулируются в пакеты Internet – IP и передаются по виртуальным каналам. Однако возникают сложности при организации и поддержке нескольких каналов в связи с необходимостью контроля состояния каждого канала. Поэтому протоколы канального уровня лучше всего подходят для создания защищенного удаленного доступа к ЛВС.

6.2.2. Сетевой уровень

Протокол Ipsec (Internet Protocol Security) входит в состав протокола IP v.6. Он предусматривает стандартные алгоритмы: аутентификации пользователей при инициализации туннеля, шифрования и электронной цифровой подписи, управления ключами. Туннель Ipsec между ЛВС поддерживает множество индивидуальных каналов передачи данных. Ipsec может работать только с IP, поэтому его используют вместе с L2TP, который не зависит от протокола транспортного уровня и работает с IPX.

Протокол SKIP (Simple Key management for Internet Protocol) управляет работой с ключами, использует алгоритм Диффи – Хелманна, но не поддерживает переговоров по поводу используемого алгоритма шифрования. Если получатель не смог расшифровать пакет, то он уже не сможет этого сделать. Версия протокола ISAKMP не содержит этот недостаток и включена в Ipsec. В IPv.4 может применяться как SKIP, так и ISAKMP.

6.2.3. Сеансовый уровень

Протоколы сеансового уровня (посредники) шифруют и ретранслируют трафик из защищаемой сети в открытую сеть Internet для каждого сокета в отдельности.

Protocol SSL/TLS (Security Sokets Lauer/Transport Lauer Security) создает защищенный туннель между конечными точками виртуальной сети, обеспечивая взаимную аутентификацию абонентов, шифрование и подлинность циркулирующих по туннелю данных. Он использует комплексно и симметричное шифрование и ассиметричное. Открытые ключи пользователей размещаются в цифровых сертификатах, заверенных ЭЦП Сертификационного центра.

Protocol SOCKS устанавливает аутентифицированный сеанс клиентского компьютера с сервером, исполняющим роль посредника. Посредник проводит любые операции, запрашиваемые клиентом, осуществляя при этом контроль за трафиком, и может блокировать конкретные приложения пользователей. Протоколы сеансового уровня по сравнению с канальными и сетевыми протоколами, которые просто открывают или закрывают канал для всего трафика в обоих направлениях, могут пропускать не весь трафик и ограничивать его направление.

Лекция 7. Создание защищенных компьютерных систем и оценка уровня их безопасности.

Общая методология создания защищенных компьютерных систем, и оценка уровня их безопасности изложена в группе международных стандартов ИСО/МЭК 15408-99 «Общие критерии оценки безопасности информационных технологий» и в их белорусских аналогах СТБ 34.101.1-3.2004 «Информационные технологии и безопасность. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель. Часть 2. Функциональные требования безопасности. Часть 3. Гарантийные требования безопасности». Эти стандарты формализуют процессы создания и оценки современных информационных систем. Поскольку безопасность таких систем невозможно оценить количественно, то предложено судить о безопасности по тому набору требований безопасности, который реализован в той или иной системе. В стандартах изложены процедуры формирования таких наборов, их оформление и использование.

7.1. Требования безопасности.

Требования безопасности - это все требования, реализация которых необходима для обеспечения безопасности информационной системы. Эти требования диктуются необходимым уровнем конфиденциальности, доступности и целостности информации, обрабатываемой в системе, а также уровнем опасности среды, в которой используется система.

Требования безопасности подразделяют на функциональные и гарантийные. Функциональные требования - это совокупность необходимых функций системы, обеспечивающих безопасность информации. Гарантийные требования - это совокупность требований, подтверждающих то, что функциональные требования сформированы правильно, реализованы в полном объеме и корректно.

7.1.1. Функциональные требования безопасности.

Все элементарные функциональные требования, называемые в стандарте элементами, образуют нижний уровень в иерархической структуре функциональных требований. Элементы группируются в компоненты, которые образуют следующий уровень иерархии. Объединение идет по общему доминирующему признаку. Функциональные компоненты объединяются в свою очередь в семейства, а семейства в – классы. Всего сформировано 11 классов, 66 семейств, 135 компонентов.

Класс формулирует одну из обобщенных функций безопасности. Ему присваивается название и маркировка. Например, класс FIA «Идентификация и

аутентификация». Маркировка трехбуквенная, «F» обозначает, что это класс функциональных требований, IA – аббревиатура названия.

Семейство формулирует некоторую часть класса. Имеет название и семибуквенную маркировку. Например, семейство FIA_UID «Идентификация пользователя».

Компонент-составная часть семейства. Имеет маркировку и название. Например, FIA _UID.1 «Выбор момента идентификации».

Элемент маркируется дополнительной цифрой и сопровождается описанием. Например, FIA _UID.1.2 «Каждый пользователь должен быть успешно идентифицирован до разрешения любого действия».

В описании любого класса имеется схема, показывающая иерархию компонентов внутри семейств (рис.7.1.). При последовательном соединении компонентов компонент, который находится выше по иерархии, обеспечивает больший уровень безопасности. Например, для семейства - 1 компонент-1 включает компоненты-2 или 3, компонент-2 включает компонент-1, но является только частью компонента-3. Для семейства – 2 компоненты-1 или 2 находятся на одном уровне иерархии и должны использоваться вместе, причем вместо компонента-2, можно использовать компонент-3, если его функции достаточно.

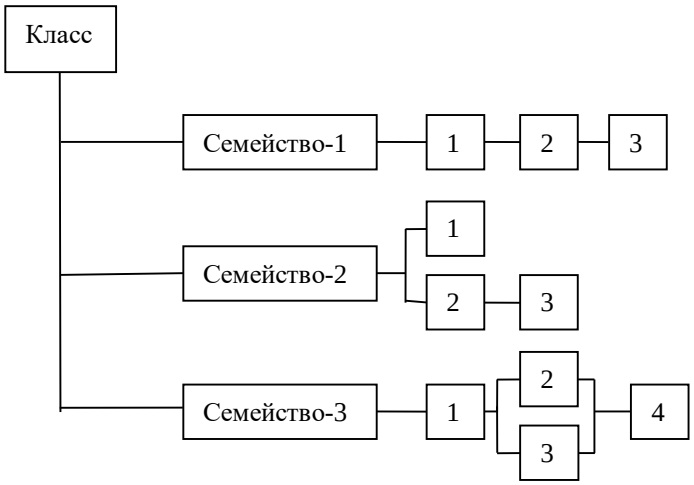


Рис.7.1. Пример структуры функциональных требований.

7.1.2 Гарантийные требования безопасности.

Гарантийные требования безопасности основываются на активном исследовании безопасности системы на всех этапах жизненного цикла. Не достижение целей безопасности возникает в следствие преднамеренного использования или случайной активизации уязвимостей. Причинами уязвимостей являются: неполнота выбранных функциональных требований;

некорректная реализация функциональных требований; несоответствующие условия эксплуатации.

Форма представления гарантийные требования безопасности аналогична форме представления функциональных требований: класс – семейство – компонент – элемент.

Каждый класс имеет название и маркировку из трех символов. Первый символ «А» означает, что это класс гарантийных требований. Например, класс ADF «Разработка».

Семейство охватывает некоторую часть класса. Имеет название и семибуквенную маркировку. Например, семейство ADF_FSP «Функциональная спецификация».

Компонент - составная часть семейства. Имеет маркировку и название. Например, ADF_FSP.1 «Неформальная функциональная спецификация».

Семейства гарантийных требований содержат только иерархические компоненты (компонент с большим номером содержит больше требований, чем компонент с меньшим).

Элемент гарантий маркируется дополнительной цифрой и буквой, сопровождается описанием. Например, ADF_FSP1.1E «Эксперт должен подтвердить, что представленная информация удовлетворяет всем требованиям к содержанию свидетельств».

Элемент гарантий принадлежит к одному из трех типов: элементы действий разработчика (маркируются символом D); элементы представления и содержания свидетельств (маркируются символом C); элементы действий эксперта (маркируются символом E).

Пример структуры гарантийных требований одного класса изображен на рис.7.2.

Всего сформировано 10 классов, 44 семейства, 93 компонентов. Перечислим классы гарантийных требований: ADF «Разработка», ALC «Поддержка жизненного цикла», ATE «Тестирование», AVA «Оценка уязвимостей», ADO «Поставка и эксплуатация», ACM «Управление конфигурацией», AGD «Руководства», AMA «Поддержка гарантий», APE «Оценка профиля защиты», ASE «Оценка задания по безопасности».

7.1.3 Оценочные уровни доверия

Гарантийные требования, предъявляемые к безопасности конкретной системы группируют в уровни гарантий. Всего предлагается 7 уровней. Каждый уровень включает компоненты из различных классов и семейств. Чем выше номер уровня гарантий, тем больше доверия к безопасности системы. Например, уровень 1 предусматривает только функциональное тестирование безопасности

системы. Может использоваться, когда достаточно только уверенности в правильном функционировании, а угрозы безопасности отсутствуют. Оценка может проводиться без помощи разработчика с минимальными затратами.

Уровень 7 предусматривает формальную верификацию проекта безопасности системы. Применим при разработке безопасных систем, работающих в условиях высокого риска. Компоненты этого уровня обеспечивают гарантию посредством анализа функциональной спецификации, полной спецификации интерфейсов, эксплуатационной документации, проектов верхнего и нижнего уровней, а также структурированного представления реализации.

7.2 Профиль защиты и задание по безопасности

Профили защиты (ПЗ) представляет собой независимый от реализации типовой набор требований безопасности для совокупности изделий определенного вида, отвечающий соответствующим целям безопасности. ПЗ предназначен для многократного использования, и определяет требования безопасности изделий, включая функциональные требования и гарантийные требования, в отношении которых установлено, что они являются достаточными и эффективными для достижения установленных целей безопасности.

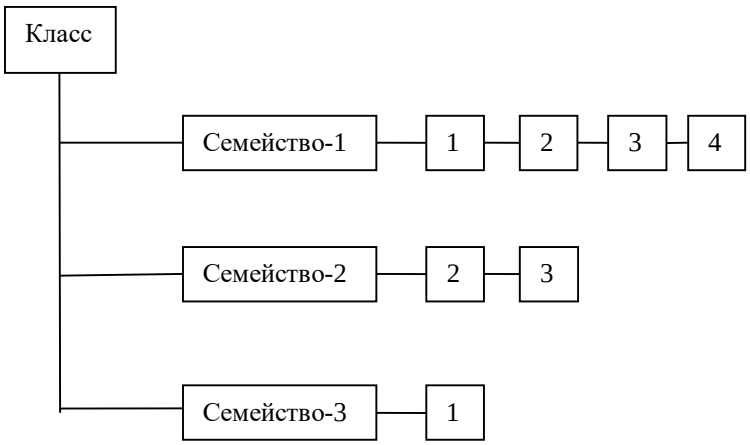


Рис.7.2. Пример структуры гарантийных требований.

Профили защиты разрабатываются и используются как стандартизованные наборы требований с целью повышения обоснованности задания требований безопасности изделий, оценки безопасности и возможности проведения сравнительного анализа уровня безопасности различных изделий.

Задание по безопасности (ЗБ) содержит совокупность требований безопасности для конкретного изделия, которые обеспечивают достижение

установленных целей безопасности. ЗБ представляет собой набор требований безопасности, которые могут быть определены ссылкой на профили защиты, ссылкой на отдельные стандартизованные требования или же содержать требования в явном виде.

ЗБ формируется разработчиком изделия и является основой для проведения оценки и сертификации изделия.

Структура профиля защиты строго регламентирована. Он содержит следующие разделы:

1. ВВЕДЕНИЕ ПЗ
1.1. Идентификация ПЗ
1.2. Аннотация ПЗ
2. ОПИСАНИЕ ОО
3. СРЕДА БЕЗОПАСНОСТИ ОО
3.1. Предположения безопасности
3.2. Угрозы
3.3. Политика безопасности организации
4. ЦЕЛИ БЕЗОПАСНОСТИ
4.1. Цели безопасности для ОО
4.2. Цели безопасности для среды
5. ТРЕБОВАНИЯ БЕЗОПАСНОСТИ ИТ
5.1. Функциональные требования безопасности ОО
5.2. Требования доверия к безопасности ОО
5.3. Требования безопасности для среды ИТ
6. ЗАМЕЧАНИЯ ПО ПРИМЕНЕНИЮ
7. ОБОСНОВАНИЕ
7.1. Логическое обоснование целей безопасности
7.2. Логическое обоснование требований безопасности.

Задание по безопасности по структуре во многом похоже на ПЗ и содержит дополнительную информацию, разъясняющую, каким образом требования ПЗ должны быть реализованы для конкретного изделия.

Лекция 8. Основы построения криптосистем

Криптография – это наука о методах, алгоритмах, программных и аппаратных средствах преобразования информации в целях сокрытия ее содержания, предотвращения видоизменения или несанкционированного использования.

Исторически сложилось так, что криптография длительное время использовалась, исключительно, как средство обеспечения конфиденциальности сообщений. Областью применения криптографии была область защиты государственной тайны: в военной, дипломатической и разведывательной сферах. Поэтому, естественно, криптография находилась в руках спецслужб. Всякие упоминания об этой науке в открытой печати были запрещены, хотя работы велись полным ходом, огромное число специалистов трудилось в этой области: математики, инженеры, разведчики. Криптографическая империя СССР противостояла аналогичной империи США. Часть специалистов трудилась над созданием стойких криптоалгоритмов, другая часть - над раскрытием чужих криптосистем.

В конце 20-го века обстановка в сфере использования криптографии коренным образом меняется. Здесь несколько причин. Главная - бурное развитие вычислительной техники, появление на этой базе информационных технологий. Доступность информационных технологий широкому кругу коммерческих компаний и частным лицам породила потребность, во-первых, обеспечивать конфиденциальность той информации, которая циркулирует в ТКС, во-вторых, обеспечивать ряд функций, таких как аутентификация субъектов системы, целостность сообщений, истинность документов и т.д. Оказалось, что все это можно обеспечить, используя принципы криптографии.

Криптография, обслуживающая задачи управления, бизнеса, телекоммуникаций, получила название открытой. Открытые криптотехнологии (ЭЦП, идентификация и аутентификация, защита от НСД) становятся коммерческими продуктами и распространяются без особых ограничений. Платежные системы: банковские, индивидуальные на основе пластиковых карт, локальные и корпоративные компьютерные сети - вот далеко неполный перечень применения криптографических технологий.

Наряду с решением задач обеспечения конфиденциальности, целостности и доступности информации существует задача анализа стойкости используемых криптопреобразований. Эта задача решается наукой, называемой криптоанализ. Криптография и криптоанализ составляют науку - криптологию.

8.1. Общие принципы криптографической защиты информации

Обобщенная схема криптографической системы, обеспечивающей шифрование передаваемой информации, имеет вид (рис.8.1).

Отправитель генерирует открытый текст исходного сообщения M, которое должно передаваться по открытому каналу. Отправитель шифрует текст с помощью обратимого преобразования E и ключа K: E_K и получает шифротекст

$C = E_K(M)$, который отправляет получателю. Получатель, приняв шифротекст C, расшифровывает его с помощью обратного преобразования $D = E_K^{-1}$ и получает исходное сообщение в виде открытого текста M: $M = D(C) = E_K^{-1}\big(E_K(M)\big)$.

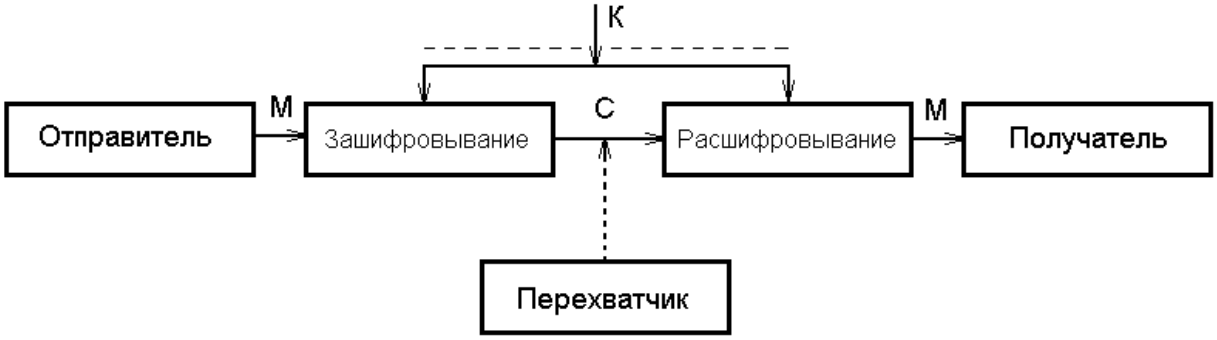


Рис.8.1. Обобщенная схема симметричной криптографической системы

Преобразование E_K выбирается из семейства криптографических преобразований, называемых криптоалгоритмами. Параметр, с помощью которого выбирается конкретное преобразование, называется криптографическим ключом K, Система, в которой осуществляется зашифровывание и расшифровывание сообщений, называется криптосистемой.

Формально криптосистема – это однопараметрическое семейство $(E_K)_{K\in\bar{K}}$ обратимых преобразований $E_K: \bar{M} \to \bar{C}$ из пространства $\bar{M}$ сообщений открытого текста в пространство $\bar{C}$ шифрованных текстов. Параметр K (ключ) выбирается из конечного множества $\bar{K}$, называемого пространством ключей. Криптосистема может иметь разные варианты реализации: набор инструкций; аппаратные или программные средства; аппаратно-программные средства.

Вообще говоря, преобразование зашифровывания может быть симметричным или асимметричным относительно преобразования расшифровывания. Поэтому различают два класса криптосистем: симметричные криптосистемы и асимметричные криптосистемы. Иногда их называют: одноключевые (с секретным ключом) и двухключевые (с открытым ключом). Схема симметричной криптосистемы с одним секретным ключом K была показана на рис.8.1. Обобщенная схема асимметричной криптосистемы с двумя разными ключами K_1 и K_2 показана на рис.8.2.

В этой криптосистеме один из ключей является открытым K_1, а другой K_2 – секретным. Для этой криптосистемы $C = E_{K_1}(M)$, а

$$M = D_{K_2}(C) = E_{K_2}^{-1}\left(E_{K_1}(M)\right)$$

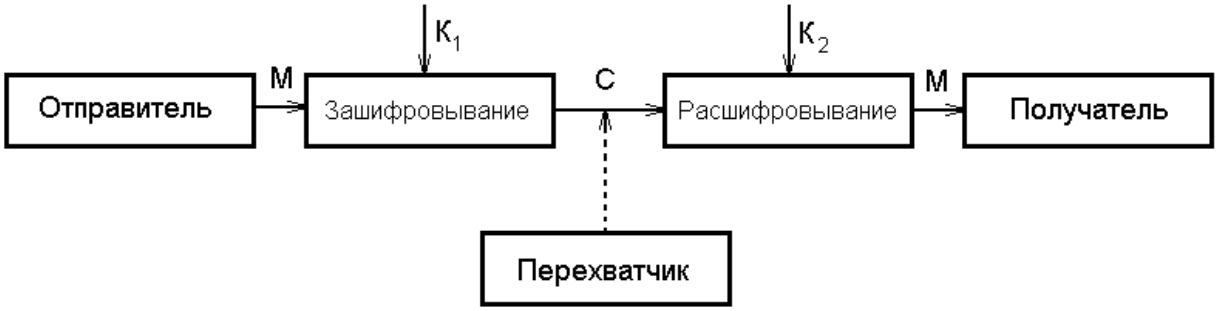


Рис.8.2. Обобщенная схема асимметричной криптографической системы

В симметричной криптосистеме секретный ключ надо передавать отправителю и получателю по защищенному каналу распространения ключей, например, спецсвязью. В асимметричной криптосистеме передают по незащищенному каналу только открытый ключ, а секретный ключ сохраняют в месте его генерации.

Злоумышленник при атаке на криптосистему может не только считывать шифротексты, передаваемые по каналу связи, но и пытаться их изменить по своему усмотрению.

Любая попытка со стороны злоумышленника расшифровать шифротекст C для получения открытого текста M или зашифровать свой собственный текст M^1 для получения правдоподобного шифротекста C^1, не имея подлинного ключа, называется криптоатакой.

Свойство криптосистемы, противостоять криптоатаке называется криптостойкостью. Оно измеряется в затратах злоумышленника, которые он несет, вскрывая криптосистему. Например, криптостойкость может выражаться в количестве машинного времени, затраченного на вскрытие криптосистемы.

Фундаментальное правило криптоанализа, впервые сформулированное голландцем А.Керкхоффом еще в XIX веке заключается в том, что стойкость шифра (криптосистемы) должна определяться только секретностью ключа. Иными словами, правило Керкхоффа состоит в том, что весь алгоритм шифрования, кроме значения секретного ключа, известен криптоаналитику противника. Это обусловлено тем, что криптосистема, реализующая семейство криптографических преобразований, обычно рассматривается как открытая система. Такой подход отражает очень важный принцип технологии защиты информации: защищенность системы не должна зависеть от секретности чего-либо такого, что невозможно быстро изменить в случае утечки секретной информации. Обычно криптосистема представляет собой совокупность аппаратных и программных средств, которую можно изменить только при значительных затратах времени и средств, тогда как ключ является легко изменяемым объектом. Именно поэтому стойкость криптосистемы определяется только секретностью ключа.

Другое почти общепринятое допущение в криптоанализе состоит в том, что криптоаналитик имеет в своем распоряжении шифротексты сообщений.

Существует четыре основных типа криптоаналитических атак. Конечно, все они формулируются в предположении, что криптоаналитику известны применяемый алгоритм шифрования и шифротексты сообщений. Перечислим эти криптоаналитические атаки.

1. Криптоаналитическая атака при наличии только известного шифртекста. Криптоаналитик имеет только шифротексты $C_1 C_2 \ldots C_i$,, нескольких сообщений, причем все они зашифрованы с использованием одного и того же алгоритма шифрования E_k. Работа криптоаналитика заключается в том, чтобы раскрыть исходные тексты $M_1 M_2 \ldots M_i$, по возможности большинства сообщений или, еще лучше, вычислить ключ K, использованный для зашифровывания этих сообщений, с тем, чтобы расшифровать и другие сообщения, зашифрованные этим ключом.

2. Криптоаналитическая атака при наличии известного открытого текста. Криптоаналитик имеет доступ не только к шифротекстам $C_1 C_2 \ldots C_i$, нескольких сообщений, но также к открытым текстам $M_1 M_2 \ldots M_i$ этих сообщений. Его работа заключается в нахождении ключа K, используемого при шифровании этих сообщений, или алгоритма расшифровывания D_k любых новых сообщении, зашифрованных тем же самым ключом.

3. Криптоаналитическая атака при возможности выбора открытого текста. Криптоаналитик не только имеет доступ к шифротекстам $C_1 C_2 \ldots C_i$, и связанным с ними открытым текстам, $M_1 M_2 \ldots M_i$ нескольких сообщений, но и может по желанию выбирать открытые тексты, которые затем получает в зашифрованном виде. Такой криптоанализ получается более мощным по сравнению с криптоанализом с известным открытым текстом, потому что криптоаналитик может выбрать для шифрования такие блоки открытого текста, которые дадут больше информации о ключе. Работа криптоаналитика состоит в поиске ключа K, использованного для шифрования сообщений, или алгоритма расшифровывания D_k НОВЫХ сообщений, зашифрованных тем же ключом.

4. Криптоаналитическая атака с адаптивным выбором открытого текста. Это - особый вариант атаки с выбором открытого текста. Криптоаналитик может не только выбирать открытый текст, который затем шифруется, но и изменять свой выбор в зависимости от результатов предыдущего шифрования. При криптоанализе с простым выбором открытого текста криптоаналитик обычно может выбирать несколько крупных блоков открытого текста для их шифрования, при криптоанализе с адаптивным выбором открытого текста он имеет возможность выбрать сначала более мелкий пробный блок открытого текста, затем выбрать следующий блок в зависимости от результатов первого выбора, и т.д. Эта атака предоставляет криптоаналитику еще больше возможностей, чем предыдущие типы атак.

Кроме перечисленных криптоаналитических атак существует силовая атака (перебор всех возможных значений ключа). С появлением мощных компьютеров и сетей этот вид атаки становится очень актуальным. Он может

сочетаться с перечисленными ранее аналитическими атаками. В связи с этим ключ криптосистемы должен обладать определенными свойствами: если рассматривать его совокупность двоичных знаков, то это должна быть случайная равномернораспределеная последовательность длины, которая делала бы перебор всех возможных значений ключа, практически невозможным.

8.2. Блочные и поточные шифры

Применение функции шифрования ко всему сообщению в целом реализуется очень редко. Практически все применяемые криптографические методы связаны с разбиением сообщения на большое число фрагментов (или знаков) фиксированного размера, каждый из которых шифруется отдельно. Такой подход существенно упрощает задачу шифрования, так как сообщения обычно имеют различную длину.

Можно выделить следующие характерные признаки методов шифрования данных:

- выполнение операций с отдельными битами или блоками.
- зависимость или независимость функции шифрования от результатов шифрования предыдущих частей сообщения.
- зависимость или независимость шифрования отдельных знаков от их положения в тексте.

В соответствии с этим различают три основных способа шифрования:

- поточные шифры;
- блочные шифры;
- блочные шифры с обратной связью.

Поточное шифрование

Поточное шифрование состоит в том, что каждый бит открытого текста и соответствующий бит ключа преобразовываются по определенному алгоритму (например, складываются по модулю 2). К достоинствам поточных шифров относятся высокая скорость шифрования, относительная простота реализации и отсутствие размножения ошибок. Недостатком является необходимость использовать для каждого сообщения другой ключ. Это обусловлено тем, что если два различных сообщения шифруются на одном и том же ключе, то эти сообщения легко могут быть расшифрованы,

$$C_1 = M_1 + K,\ C_2 = M_2 + K, C_1 + C_2 = M_1 + M_2$$

считая M_2 ключом можно вычислить M_1, т.к. M_2 не обладает свойствами ключа). Поэтому часто используют дополнительный, случайно выбираемый ключ сообщения, который передается в начале сообщения и применяется для модификации ключа шифрования. В результате разные сообщения будут шифроваться с помощью различных последовательностей. Это требует передачи информации синхронизации перед заголовком сообщения, которая должна быть принята до расшифровывания любого сообщения.

Поточные шифры широко применяются для шифрования преобразованных в цифровую форму речевых сигналов и цифровых данных, требующих оперативной доставки потребителю информации.

Блочное шифрование

При блочном шифровании открытый текст сначала разбивается на равные по длине блоки, затем применяется зависящая от ключа функция шифрования для преобразования блока открытого текста длиной $\boldsymbol{m}$ бит в блок шифротекста такой же длины. При блочном шифровании каждый бит блока шифротекста зависит от значений всех битов соответствующего блока открытого текста, и никакие два блока открытого текста не могут быть представлены одним и тем же блоком шифротекста. При этом небольшие изменения в шифротексте вызывают большие и непредсказуемые изменения в соответствующем открытом тексте, и наоборот. Вместе с тем применение блочного шифра имеет серьезные недостатки. Первый из них заключается в том, что вследствие детерминированного характера шифрования при фиксированной длине блока 64 бита можно осуществить криптоанализ шифротекста "со словарем" в ограниченной форме. Это обусловлено тем, что идентичные блоки открытого текста длиной 64 бита в исходном сообщении представляются идентичными блоками шифротекста, что позволяет криптоаналитику сделать определенные выводы о содержании сообщения. Другой потенциальный недостаток этого шифра связан с размножением ошибок. Результатом изменения только одного бита в принятом блоке шифротекста будет неправильное расшифровывание всего блока. Это, в свою очередь, приведет к появлению искаженных битов (от 1 до 64) в восстановленном блоке исходного текста.

Из-за отмеченных недостатков блочные шифры редко применяются в указанном режиме для шифрования длинных сообщений. Однако в финансовых учреждениях, где сообщения часто состоят из одного или двух блоков, блочные шифры широко используют в режиме прямого шифрования. Такое применение обычно связано с возможностью частой смены ключа шифрования, поэтому вероятность шифрования двух идентичных блоков открытого текста на одном и том же ключе очень мала.

Криптосистема с открытым ключом также является системой блочного шифрования и должна оперировать блоками довольно большой длины. Это обусловлено тем, что криптоаналитик знает открытый ключ шифрования и мог бы заранее вычислить и составить таблицу соответствия блоков открытого текста и шифротекста. Если длина блоков мала, например 30 бит, то число возможных блоков не слишком большое (при длине 30 бит это $2^{30} \approx 10^9$), и может быть составлена полная таблица, позволяющая моментально расшифровать любое сообщение с использованием известного открытого ключа.

Блочное шифрование с обратной связью

Как и при блочном шифровании, сообщения разбивают на ряд блоков, состоящих из $\boldsymbol{m}$ бит. Для преобразования этих блоков в блоки шифротекста,

которые также состоят из m бит, используются специальные функции шифрования. Однако если в блочном шифре такая функция зависит только от ключа, то в блочных шифрах с обратной связью она зависит как от ключа, так и от одного или более предшествующих блоков шифротекста.

Практически важным шифром с обратной связью является шифр со сцеплением блоков шифротекста. В этом случае $\boldsymbol{m}$ бит предыдущего шифротекста суммируются по модулю 2 со следующими $\boldsymbol{m}$ битами открытого текста, а затем применяется алгоритм блочного шифрования под управлением ключа для получения следующего блока шифротекста. Достоинством криптосистем блочного шифрования с обратной связью является возможность применения их для обнаружения манипуляций сообщениями, производимых активными перехватчиками. При этом используется факт размножения ошибок в таких шифрах, а также способность этих систем легко генерировать код аутентификации сообщений. Поэтому системы шифрования с обратной связью используют не только для шифрования сообщений, но и для их аутентификации. Криптосистемам блочного шифрования с обратной связью свойственны некоторые недостатки. Основным из них является размножение ошибок, так как один ошибочный бит при передаче может вызвать ряд ошибок в расшифрованном тексте. Другой недостаток связано тем, что разработка и реализация систем шифрования с обратной связью часто оказываются более трудными, чем систем поточного шифрования.

На практике для шифрования длинных сообщений применяют поточные шифры или шифры с обратной связью. Выбор конкретного типа шифра зависит от назначения системы и предъявляемых к ней требований.

Лекция 9. Симметричные криптосистемы

9.1. Основные понятия и определения

Большинство средств защиты информации базируется на использовании криптографических шифров и процедур зашифровывания - расшифровывания. В соответствии со стандартом ГОСТ 28147-89 под шифром понимают совокупность обратимых преобразований множества открытых данных на множество зашифрованных данных, задаваемых ключом и алгоритмом криптографического преобразования.

Ключ - это конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор только одного варианта из всех возможных для данного алгоритма. Основной характеристикой шифра является криптостойкость, которая определяет его стойкость к раскрытию методами криптоанализа. Обычно эта характеристика определяется интервалом времени, необходимым для раскрытия шифра. К шифрам, используемым для криптографической защиты информации, предъявляется ряд требований:

- достаточная криптостойкость (надежность закрытия данных);
- простота процедур шифрования и расшифровывания;
- незначительная избыточность информации за счет шифрования;
- нечувствительность к небольшим ошибкам шифрования и др.

В той или иной мере этим требованиям отвечают:

- шифры перестановок;
- шифры замены;
- шифры гаммирования;
- шифры, основанные на аналитических преобразованиях шифруемых данных.

Шифрование перестановкой заключается в том, что символы шифруемого текста переставляются по определенно правилу в пределах некоторого блока этого текста. При достаточной длине блока, в пределах которого осуществляется перестановка, и сложном неповторяющемся порядке перестановки можно достигнуть приемлемой для простых практических приложений стойкости шифра.

Шифрование заменой (подстановкой) заключается в том, что символы шифруемого текста заменяются символами того же или другого алфавита в соответствии с заранее обусловленной схемой замены.

Шифрование гаммированием заключается в том, что символы шифруемого текста складываются с символами некоторой случайной последовательности, именуемой гаммой шифра. Стойкость шифрования определяется в основном длиной (периодом) неповторяющейся части гаммы шифра. Поскольку с помощью ЭВМ можно генерировать практически бесконечную гамму шифра, то данный способ является одним из основных для шифрования информации в автоматизированных системах.

Шифрование аналитическим преобразованием заключается в том, что шифруемый текст преобразуется по некоторому аналитическому правилу (формуле). Например, можно использовать правило умножения вектора на матрицу, причем умножаемая матрица является ключом шифрования (поэтому ее размер и содержание должны хранится в секрете), а символами умножаемого вектора последовательно служат символы шифруемого текста.

Процессы зашифровывания и расшифровывания осуществляются в рамках некоторой криптосистемы. Характерной особенностью симметричной криптосистемы является применение одного секретного ключа как при зашифровывании, так и при расшифровывании сообщений.

Как открытый текст, так и шифротекст образуются из букв, входящих в конечное множество символов, называемое алфавитом. Примерами алфавитов являются конечное множество всех заглавных букв, конечное множество всех заглавных и строчных букв и цифр и т.п. В общем случае некоторый алфавит можно представить как $\sum = \{\boldsymbol{a_0}, \boldsymbol{a_1}, \dots, \boldsymbol{a_{m-1}}\}$.

Объединяя по определенному правилу буквы из алфавита $\sum$, можно создать новые алфавиты:

- Алфавит $\sum^2$, содержит m^2 биграмм $a_0a_0, a_0a_1, \dots, a_{m-1}a_{m-1}$;
- алфавит $\sum^3$, содержит m^3 биграмм $a_0a_0, a_0a_1, \dots, a_{m-1}a_{m-1}a_{m-1}$;

В общем случае, объединяя по n букв, получаем алфавит $\sum^n$, содержащий m^n - n-грамм. Например, английский алфавит $\{ABCD \dots XYZ\}$. объемом m = 26 букв позволяет сгенерировать $26^2 = 676$ биграмм $AA, AB, \dots, ZZ$,

$26^2 = 17576$ триграмм $AAA, AAB, \dots, ZZY, ZZZ$.

При выполнении криптографических преобразований полезно заменить буквы алфавита целыми числами 0,1,2,… Это позволяет упростить выполнение необходимых алгебраических манипуляций. Например, можно установить взаимно однозначное соответствие между русским алфавитом {АБВГ…ЮЯ} и множеством целых чисел $\sum = \{0,1,2, \dots, 31\}$, между английским алфавитом $\{ABCD \dots YZ\}$ и множеством целых чисел $\sum_{32} = \{0,1,2, \dots, 26\}$.

В дальнейшем будет обычно использоваться алфавит

$\overline{\sum_m} = \{0,1,2, \dots, m-1\}$.

содержащий m «букв» в виде чисел. Замена букв традиционного алфавита числами позволит более четко сформулировать основные концепции и приемы криптопреобразований. В тоже время в большинстве иллюстраций будет использоваться алфавит естественного языка.

Текст с n буквами из алфавита $\overline{\sum_m}$ можно рассматривать как n-грамму $\bar{X} = \{x_0, x_1, \dots, x_{n-1}\}$, где $x \in \overline{Z_m}$, для некоторого целого n=1,2,3…. Через $Z_{m,n}$ будем обозначать множество n-грамм, образованных из букв множества Z_m.

Криптографическое преобразование E представляет собой совокупность преобразований $E = \{E^{(n)}: 1 \le n < \infty\}$, $E^{(n)}: Z_{m,n} \to Z_{m,n}$. Преобразование $E^{(n)}$

определяет, как каждая n-грамма открытого текста $\bar{x} \in \overline{Z_{m,n}}$ заменяется n-граммой шифротекста $\bar{y}$, т.е. $\bar{y} = E^{(n)}(\bar{x})\, \bar{y} = E^{(n)}\left(\bar{x}\right)$, причем $\bar{x}$, $\bar{y} \in \overline{Z_{m,n}}$, при этом обязательным является требование взаимной однозначности преобразования, $E^{(n)}$ на множестве $\overline{Z_{m,n}}$.

Криптографическая система может трактоваться как семейство криптографических преобразований $E = \{E_K : K \in \bar{k}\}$, помеченных параметром K называется ключом. Множество значений ключа образуют ключевое пространство $\bar{k}$.

9.2. Традиционные симметричные криптосистемы

Традиционные (классические) методы шифрования, отличаются симметричной функцией шифрования. К ним относятся шифры перестановки, шифры простой и сложной замены, а также некоторые их модификации и комбинации. Следует отметить, что комбинации шифров перестановок и шифров замены образуют все многообразие применяемых на практике симметричных шифров.

Шифры перестановок

Правило перестановок символов - является ключом и задается различными предметами: цилиндром (скитала, древние греки), размером таблицы, условным словом или фразой (шифрующие таблицы в эпоху Возрождения), магическим квадратом в средние века.

Шифры простой замены

В шифрах простой замены каждый символ открытого текста заменяется символом того же или другого алфавита по определенному правилу. Широко известны и исследованы шифры Цезаря. Такие шифры имеют слабость по отношению к атакам на основе подсчета частот появления букв в шифротексте. Более устойчивыми являются биграммные шифры (замена двух букв) и n граммные шифры, позволяющие маскировать частоты появления букв.

Шифры сложной замены

Шифры сложной замены называют многоалфавитными, т.к. для шифрования каждого символа исходного сообщения применяют свой шифр простой замены. Многоалфавитная подстановка последовательно и циклически меняет используемые алфавиты. Например, в r - алфавитной подстановке символ x_0 исходного текста заменяется символом y_0 из алфавита B_0, x_1 на y_1 из B_1, x_{r-1} на y_{r-1} из B_{r-1}, x_r на y_r из B_r. Многоалфавитная подстановка маскирует естественную статистику исходного языка, т.к. конкретный символ из алфавита A может быть преобразован в несколько символов шифровальных алфавитов B. К шифры сложной замены относят шифры Гронсфельда, Вижинера, Вернама. В 20 годах были созданы первые шифровальные машины (электромеханические),

реализующие шифры сложной замены. Эти машины использовались до 60-х годов.

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

Следует отметить, что перед зашифровыванием открытые данные разбивают на блоки T_o^i одинаковой длины, обычно по 64 бита. Гамма шифра вырабатывается в виде последовательности блоков $\Gamma_{\text{ш}}^i$ аналогичной длины.

Уравнение зашифровывания можно записать в виде $T_{\text{ш}}^i = \Gamma_{\text{ш}}^i \oplus T_o^i,\ i = 1,2,\dots,M$, где T_o- i-й блок шифротекста, $\Gamma_{\text{ш}}^i$ - i -й блок гаммы шифра, T_o^i- i -й блок открытого текста, M - количество блоков открытого текста.

Процесс расшифровывания сводится к повторной генерации гаммы шифра и наложению этой гаммы на зашифрованные данные. Уравнение расшифровывания имеет вид: $T_o^i = \Gamma_{\text{ш}}^i \oplus T_{\text{ш}}^i$.

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

9.3. Современные симметричные криптосистемы

По мнению К. Шеннона, в практических шифрах необходимо использовать два общих принципа: рассеивание и перемешивание.

Рассеивание представляет собой распространение влияния одного знака открытого текста на много знаков шифротекста, что позволяет скрыть статистические свойства открытого текста.

Перемешивание предполагает использование таких шифрующих преобразований, которые усложняют восстановление взаимосвязи статистических свойств открытого и шифрованного текстов. Однако шифр должен не только затруднять раскрытие, но обеспечивать легкость зашифровывания и расшифровывания при известном пользователю секретном ключе.

Распространенным способом достижения эффектов рассеивания и перемешивания является использование составного шифра, т.е. такого шифра, который может быть реализован в виде некоторой последовательности простых

шифров, каждые из которых вносит свой вклад в значительное суммарное рассеивание и перемешивание.

В составных шифрах в качестве простых шифров чаще всего используются простые перестановки и подстановки. При перестановке просто перемешивают символы открытого текста, причем конкретный вид перемешивания определяется секретным ключом. При подстановке каждый символ открытого текста заменяют другим символом из того же алфавита, а конкретный вид подстановки также определяется секретным ключом. Следует заметить, что в современном блочном шифре блоки открытого текста и шифротекста представляют собой двоичные последовательности обычно длиной 64 бита. В принципе каждый блок может принимать 2^{64} значений. Поэтому подстановки выполняются в очень большом алфавите, содержащем до $2^{54} \approx 10^{19}$символов.

При многократном чередовании простых перестановок и подстановок, управляемых достаточно длинным секретным ключом, можно получить очень стойкий шифр с хорошим рассеиванием и перемешиванием.

Лекция 10. Стандарт шифрования данных ГОСТ 28147-89

ГОСТ 28147-89 представляет собой 64-битовый блочный алгоритм с 256-битовым ключом. Он предназначен для аппаратной и программной реализации, удовлетворяет криптографическим требованиям и не накладывает ограничений на степень секретности защищаемой информации.

Алгоритм предусматривает четыре режима работы:

– шифрование данных в режиме простой замены;

– шифрование данных в режиме гаммирования;

– шифрование данных в режиме гаммирования с обратной связью;

– выработка имитовставки.

Основными режимами шифрования являются режимы с использованием гаммирования, однако они базируются на использовании шифрования данных в режиме простой замены.

10.1. Режим простой замены

Шифрование открытых данных в режиме простой замены

Открытые данные, подлежащие шифрованию, разбивают на 64-разрядные блоки T_O. Процедура шифрования 64-разрядного блока T_O в режиме простой замены включает 32 цикла($j = 1,2, \dots ,32$). В ключевое запоминающее устройство вводят 256 бит ключа K в виде восьми 32-разрядных подключей (чисел) K_i

$K = K_7K_6K_5K_4K_3K_2K_1K_0$.

Последовательность битов блока

$T_0 = (a_1(0), a_2(0), \dots , a_{31}(0), a_{32}(0), b_1(0), b_2(0), \dots , b_{31}(0), b_{32}(0))$

разбивают на две последовательности по 32 бита:$b(0)$ и $a(0)$, где $b(0)$ - левые или старшие биты, $a(0)$ - правые или младшие биты.

Работа алгоритма в режиме простой замены изображена на рис. 4.1.

Обозначения на схеме:

N_1, N_2 32-разрядные накопители;

CM_1 32-разрядный сумматор по модулю 2^{32} $(+)$;

CM_2 32-разрядный сумматор по модулю$2(\oplus)$;

R 32-разрядный регистр циклического сдвига;

КЗУ - ключевое запоминающее устройство на 256 бит, состоящее из восьми 32-разрядных накопителей z;

S - блок подстановки, состоящий из восьми узлов замены (S-блоков замены) $S_1, S_2, S_3, \dots , S_8$.

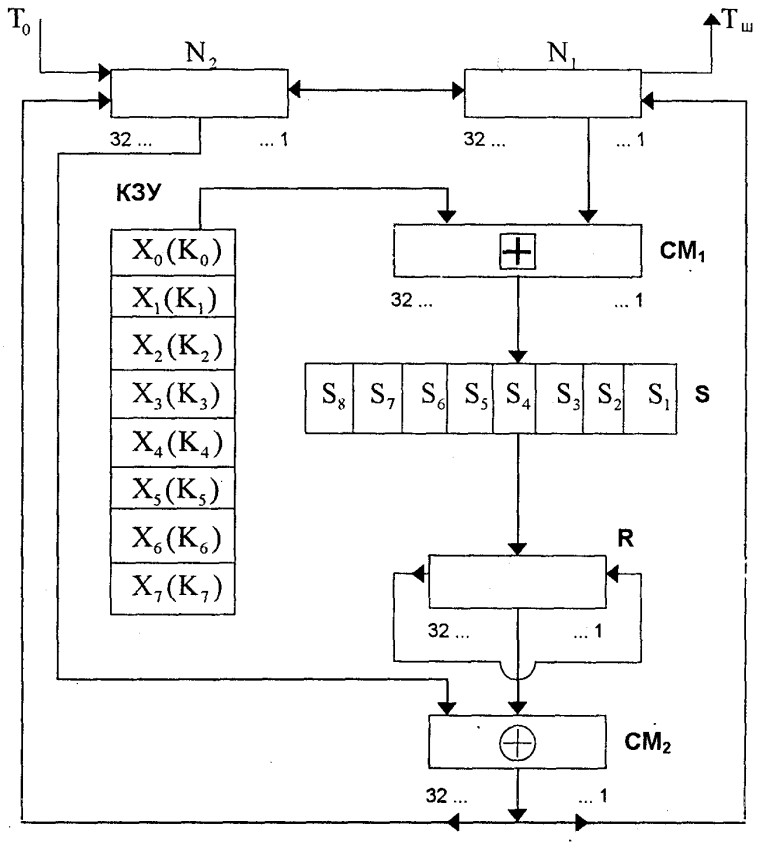


Рис. 10.1. Схема реализации режима простой замены

Эти последовательности вводят в накопителиN_1 и N_2 перед началом первого цикла шифрования. В результате начальное заполнение накопителя N_1:

$$a(0) = (a_{32}(0), a_{31}(0), \dots, a_2(0), a_1(0))$$

32, 31, … , 2, 1 ← номер разряда N_1

начальное заполнение накопителя N_2:

$$b(0) = (b_{32}(0), b_{31}(0), \dots, b_2(0), b_1(0))$$

32, 31, … , 2, 1 ← номер разряда N_2

Первый цикл $(j = 1)$ процедуры шифрования 64-разрядного блока открытых данных можно описать уравнениями:

$$\begin{cases} a(1) = f(a(0) + K_0) \oplus b(0), \\ b(1) = a(0). \end{cases}$$

Здесь $a(1)$ – заполнение N_1 после 1-го цикла шифрования; $b(1)$ - заполнение N_2 после 1-го цикла шифрования; f- функция шифрования.

Аргументом функции f является сумма по модулю 2^{32} числа $a(0)$ (начального заполнения накопителя N_1) и числа K_0 подключа, считываемого из накопителя X_0 КЗУ. Каждое из этих чисел равно 32 битам.

Функция f включает две операции над полученной 32-разрядной суммой $(a(0) + X_0)$.

Первая операция называется подстановкой (заменой) и выполняется блоком подстановки S. Блок подстановки S состоит из восьми узлов замены (S - блоков замены) $S_1, S_2, \dots, S_8$ с памятью 64 бит каждый. Поступающий из CM_2 на блок подстановки S 32-разрядный вектор разбивают на восемь последовательно идущих 4-разрядных векторов, каждый из которых преобразуется в четырехразрядный вектор соответствующим узлом замены. Каждый узел замены можно представить в виде таблицы-перестановки шестнадцати четырехразрядных двоичных чисел в диапазоне $0000 \dots 1111$. Входной вектор указывает адрес строки в таблице, а число в этой строке является выходным вектором. Затем четырехразрядные выходные векторы последовательно объединяют в 32-разрядный вектор. Узлы замены (таблицы-перестановки) представляют собой ключевые элементы, которые являются общими для сетей ТКС и редко изменяются. Эти узлы замены должны сохраняться в секрете.

Вторая операция - циклический сдвиг влево (на 11 разрядов) 32-разрядного вектора, полученного с выхода блока подстановки S. Циклический сдвиг выполняется регистром сдвига R. Затем результат работы функции шифрования f суммируют поразрядно по модулю 2 в сумматоре CM_2 с 32-разрядным начальным заполнением $b(0)$ накопителя N_2. Затем полученный на выходе CM_2 результат (значение $a(1)$) записывают в накопитель N_1, а старое значение N_1 (значение $a(0)$) переписывают в накопитель N_2 (значение $b(1) = a(0)$). Первый цикл завершен. Последующие циклы осуществляются аналогично, при этом во втором цикле из КЗУ считывают заполнение X_1 – подключ K_1, в третьем цикле - подключ K_2 и т.д., в восьмом цикле - подключ K_7. В циклах с 9-го по 16-й, а также в циклах с 17-го по 24-й подключи из КЗУ считываются в том же порядке: $K_0, K_1, K_2, \dots K_6, K_7$. В последних восьми циклах с 25-го по 32-й порядок считывания подключей из КЗУ обратный: $K_7, K_6, \dots K_2, K_1, K_0$. Таким образом, при шифровании в 32 циклах осуществляется следующий порядок выборки из КЗУ подключей:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7,$$
$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

В 32-м цикле результат из сумматора CM_2 вводится в накопитель N_2, а в накопителе N_1 сохраняется прежнее заполнение. Полученные после 32-го цикла шифрования заполнения накопителей N_1 и N_2. являются блоком шифрованных данных $\mathrm{T_ш}$, соответствующим блоку открытых данных $\mathrm{T_0}$.

Уравнения шифрования в режиме простой замены имеют вид

$$\begin{cases} a(j) = f(a(j-1) + K_{(j-1)mod8}) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 1 \dots 24,$$

$$\begin{cases} a(j) = f\left(a(j-1) + K_{(32-j)mod8}\right) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 25 \dots 31,$$

$$\begin{cases} a(j) = a(j-1) \\ b(j) = f(a(j-1) + K_0) \oplus b(j-1) \end{cases} \quad \text{при } j = 32.$$

где $a(j) = (a_{32}(j), a_{31}(j), \dots a_1(j))$ - заполнение N_1 после j-го цикла шифрования; $b(j) = (b_{32}(j), b_{31}(j), \dots, b_1(j))$ - заполнение N_2. после j-го цикла шифрования, $j = 1 \dots 32$..

Блок зашифрованных данных $\text{Т}_\text{ш}$ (64 разряда) выводится из накопителей N_1, N_2в следующем порядке: из разрядов $1 \dots 32$ накопителя N_1, затем из разрядов $1 \dots 32$ накопителя N_2, т.е. начиная с младших разрядов:

$\text{Т}_\text{ш} = (a_1(32), a_2(32), \dots a_{32}(32), b_1(32), b_2(32), \dots b_{32}(32))$.

Остальные блоки открытых данных зашифровываются в режиме простой замены аналогично.

Расшифрование в режиме простой замены

Криптосхема, реализующая алгоритм расшифрования в режиме простой замены, имеет тот же вид, что и при шифровании (см. рис. 10.1).

В КЗУ вводят 256 бит ключа, на котором осуществлялось шифрование. Зашифрованные данные, подлежащие расшифрованию, разбиты на блоки $\text{Т}_\text{ш}$, по 64 бита в каждом. Ввод любого блока

$\text{Т}_\text{ш} = (a_1(32), a_2(32), \dots a_{32}(32), b_1(32), b_2(32), \dots b_{32}(32))$.

в накопители N_1 и N_2 производят так, чтобы начальное значение накопителя N_1 имело вид

$a(0) = (a_{32}(32), a_{31}(32), \dots, a_2(32), a_1(32))$

32, 31, … , 2, 1 ← номер разряда N_1

а начальное заполнение накопителя N_2: $b(0) = (b_{32}(32), b_{31}(32), \dots, b_2(32), b_1(32))$

32, 31, … , 2, 1 ← номер разряда N_2

Расшифрование осуществляется по тому же алгоритму, что и шифрование, с тем изменением, что заполнения накопителей $X_0, X_1, X_2, \dots, X_7$ считываются из КЗУ в циклах расшифрования в следующем порядке:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0,$$
$$K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

Уравнения расшифрования имеют вид:

$$\begin{cases} a(32-j) = f\big(a(32-j+1) + K_{(j-1)}\big) \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad \text{при } j = 1 \dots 8,$$

$$\begin{cases} a(32-j) = f\big(a(32-j+1) + K_{(32-j)mod8}\big) \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \text{при } j = 9 \dots 31,$$

$$\begin{cases} a(0) = a(1) \\ b(0) = f(a(1) + K_0) \oplus b(1) \end{cases} \quad \text{при } j = 32.$$

Полученные после 32 циклов работы заполнения накопителей N_1 и N_2 образуют блок открытых данных

$$\mathrm{T}_0 = (a_1(0), a_2(0), \dots a_{32}(0), b_1(0), b_2(0), \dots b_{32}(0))$$

соответствующий блоку зашифрованных данных $T_ш$. При этом состояние накопителя N_1:

$$a(0) = (a_{32}(0), a_{31}(0), \dots, a_2(0), a_1(0))$$
$$32, \quad 31, \ \dots, \quad 2, \quad 1 \leftarrow \text{номер разряда } N_1$$

состояние накопителя N_2:

$$b(0) = (b_{32}(0), b_{31}(0), \dots, b_2(0), b_1(0))$$
$$32, \quad 31, \quad \dots, \quad 2, \quad 1 \quad \leftarrow \text{номер разряда } N_2$$

Аналогично расшифровываются остальные блоки зашифрованных данных.

Если алгоритм зашифрования в режиме простой замены 64-битового блока T_0 обозначить через A, то

$$A(\mathrm{T}_0) = A\big(a(0), b(0)\big) = (a(32), b(32)) = \mathrm{T}_{\text{ш}}.$$

Следует иметь в виду, что режим простой замены допустимо использовать для шифрования данных только в ограниченных случаях при выработке ключа и шифровании его с обеспечением имитозащиты для передачи по каналам связи или для хранения в памяти ЭВМ.

10.2. Режим гаммирования

Зашифровывание открытых данных в режиме гаммирования

Криптосхема, реализующая алгоритм шифрования в режиме гаммирования, показана на рис. 10.2. Открытые данные разбивают на 64-разрядные блоки

$\mathrm{T}_0^{(1)}, \mathrm{T}_0^{(2)}, \dots, \mathrm{T}_0^{(i)}, \dots, \mathrm{T}_0^{(m)}$,

где $\mathrm{T}_0^{(i)}$- i-й 64-разрядный блок открытых данных, $i = 1, \dots, m$, m - определяется объемом шифруемых данных.

Эти блоки поочередно зашифровываются в режиме гаммирования путем поразрядного сложения по модулю 2 в сумматоре CM_5 с гаммой шифра $\Gamma_{ш}$, которая вырабатывается блоками по 64 бита, т.е.

$$\Gamma_{ш} = (\Gamma_{ш}^{(1)}, \Gamma_{ш}^{(2)}, \dots, \Gamma_{ш}^{(i)}, \dots, \Gamma_{ш}^{(m)}$$

где $\Gamma_{ш}^{(i)}$ - i-й 64-разрядный блок, $i = 1, \dots, m$.

Число двоичных разрядов в блоке $T_0^{(m)}$ может быть меньше 64, при этом не использованная для зашифрования часть гаммы шифра из блока $\Gamma_{ш}^{(m)}$ отбрасывается.

Уравнение шифрования данных в режиме гаммирования имеет вид

$T_{ш}^{(i)} = T_0^{(i)} \oplus \Gamma_{ш}^{(i)}$,

где $\Gamma_{ш}^{(i)} = A(Y_{i-1} + C_{2,} Z_{i-1} + C_1), i = 1, \dots, m$; $T_{ш}^{(i)}$ - i-й блок 64-разрядного блока зашифрованного текста; $A(*)$ - функция шифрования в режиме простой замены; C_1, C_2 - 32-разрядные двоичные константы; Y_i, Z_i - 32-разрядные двоичные последовательности.

Величины Y_i, Z_i определяются итерационно по мере формирования гаммы $\Gamma_{ш}^{(i)}$ следующим образом:

$$(Y_0, Z_0) = A(\widetilde{S}),$$

где $\tilde{S}$ - синхропосылка (64-разрядная двоичная последовательность),

$(Y_0, Z_0) = (Y_{i-1} + C_2, Z_{i-1} + C_1), i = 1, \dots, m.$

Рассмотрим реализацию процедуры шифрования в режиме гаммирования. В накопители N_6 и N_5 заранее записаны 32-разрядные двоичные константы C_1 и C_2, имеющие следующие значения (в шестнадцатеричной форме):

$$C_1=01010104_{(16)}, \quad C_2=01010101_{(16)}$$

В КЗУ вводится 256 бит ключа; в накопители N_1 иN_2 - 64-разрядная двоичная последовательность (синхропосылка)

$$\tilde{S} = (S_1, S_2, \dots, S_{64}).$$

Синхропосылка $\tilde{S}$ является исходным заполнением накопителей N_1 и

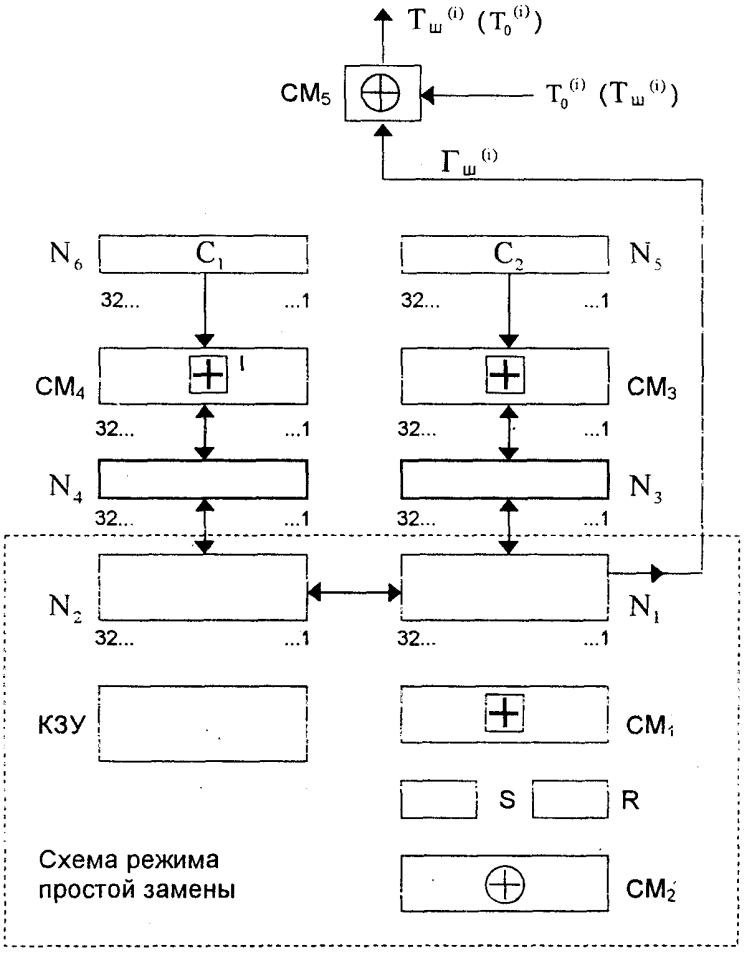


Рис. 10.2. Схема реализации режима гаммирования

N_2 для последовательной выработки m блоков гаммы шифра.

Исходное заполнение накопителяN_1:

$$(S_{32}, \ S_{31}, \ \dots, \ S_2, S_1)$$
$$32, \ \ 31, \ \dots, \ \ 2, \ 1 \leftarrow \text{номер разряда } N_1,$$

состояние накопителя N_2 :

$$(S_{64}, \ S_{63}, \ \dots, \ S_{34}, S_{33})$$
$$64, \ \ 63, \ \dots, \ \ 34, \ 33 \leftarrow \text{номер разряда } N_2,$$

Исходное заполнение N_1 и N_2 (синхропосылка $\tilde{S}$ шифруется в режиме простой замены. Результат шифрования

$$A(\tilde{S}) = (Y_0, Z_0)$$

переписывается в 32-разрядные накопители N_3 и N_4 так, что заполнение N_1 переписывается в N_3 , а заполнение N_2 - вN_4 .

Заполнение накопителяN_4 суммируют по модулю$2^{32}-1$ в сумматоре CM_4 с 32-разрядной константой C_1 из накопителя N_6. Результат записывается вN_4 . Заполнение накопителя N_3 суммируется по модулю2^{32} в сумматоре CM_3 с 32-разрядной константой C_2 из накопителя N_5. Результат записывается в N_3. Заполнение N_3 переписывают вN_1, а заполнение N_4 - в N_2, при этом заполнения N_3, N_4 сохраняются. Заполнение накопителей шифруется в режиме простой замены.

Полученное в результате шифрования заполнение накопителей N_1 и N_2 образует первый 64-разрядный блок гаммы шифра:

$$\Gamma_{\text{ш}}^{(1)} = (\gamma_1^{(1)}, \gamma_2^{(1)}, \dots, \gamma_{63}^{(1)}, \gamma_{64}^{(1)})$$

который суммируют поразрядно по модулю 2 в сумматоре CM_5 с первым 64-разрядным блоком открытых данных:

$$\text{T}_o^{(1)} = (t_1^{(1)}, t_2^{(1)}, \dots, t_{63}^{(1)}, t_{64}^{(1)})$$

В результате суммирования по модулю 2 значений $\Gamma_{\text{ш}}^{(1)}$ и $T_o^{(1)}$ получают первый 64-разрядный блок зашифрованных данных

$$\text{T}_{\text{ш}}^{(1)} = \Gamma_{\text{ш}}^{(1)} \oplus \text{T}_{\text{o}}^{(1)} = (\tau_1^{(1)}, \tau_2^{(1)}, \dots, \tau_{63}^{(1)}, \tau_{64}^{(1)})$$

,

где$\tau_i^{(1)} = t_i^{(1)} \oplus \gamma_i^{(1)}$, $i = 1 \dots 64$.

Для получения следующего 64-разрядного блока гаммы шифра $\Gamma_{\text{ш}}^{(2)}$ заполнение N_4 суммируется по модулю $(2^{32}-1)$ в сумматоре CM_4 с константой C_1 из N_6. Результат записывается в N_4. Заполнение N_3 суммируется по модулю 2^{32} в сумматоре CM_3 с константой C_2 из N_5. Результат записывается в N_3. Новое заполнение N_3 переписывают в N_1, а новое заполнение N_4 - в N_2, при этом заполнения N_3 и N_2 сохраняют. Заполнения N_1, N_2 шифруют в режиме простой замены.

Полученное в результате шифрования заполнение накопителей N_1 и N_2 образует второй 64-разрядный блок гаммы шифра $\Gamma_{\text{ш}}^{(2)}$, который суммируется поразрядно по модулю 2 в сумматоре CM_5 со вторым блоком открытых данных $\text{T}_o^{(2)}$

$\text{T}_{\text{ш}}^{(2)} = \Gamma_{\text{ш}}^{(2)} \oplus \text{T}_{\text{o}}^{(2)}$.

Аналогично вырабатываются блоки гаммы шифра $\Gamma_{\text{ш}}^{(3)}, \Gamma_{\text{ш}}^{(4)}, \dots, \Gamma_{\text{ш}}^{(m)}$и шифруются блоки открытых данных $\text{T}_o^{(3)}, \text{T}_o^{(3)}, \dots, \text{T}_o^{(m)}$.

В канал связи или память ЭВМ передаются синхропосылка $\tilde{\text{S}}$ и блоки зашифрованных данных:

$$\text{T}_{\text{Ш}}^{(1)}, \text{T}_{\text{Ш}}^{(2)}, \dots, \text{T}_{\text{Ш}}^{(m)}.$$

Расшифрование в режиме гаммирования

При расшифровании криптосхема имеет тот же вид, что и при шифровании (см. рис. 10.2).

Уравнение расшифрования

$$\mathrm{Т}_{О}^{(i)} = \mathrm{Т}_{\mathrm{Ш}}^{(i)} \oplus \Gamma_{\mathrm{Ш}}^{(i)} = \mathrm{Т}_{\mathrm{Ш}}^{(i)} \oplus \mathrm{A}(Y_{i-1} + \mathrm{C}_2, \mathrm{Z}_{i-1} + \mathrm{C}_1), i = 1 \dots m.$$

Следует отметить, что расшифрование данных возможно только при наличии синхропосылки, которая не является секретным элементом шифра и может храниться в памяти ЭВМ или передаваться по каналам связи вместе с зашифрованными данными.

Рассмотрим реализацию процедуры расшифрования. В КЗУ вводят 256 бит ключа, с помощью которого осуществляется шифрование данных $\mathrm{Т}_{О}^{(1)}, \mathrm{Т}_{О}^{(2)}, \dots, \mathrm{Т}_{О}^{(m)}$. В накопители N_1 и N_2 вводится синхропосылка и осуществляется процесс выработки m блоков гаммы шифра $\Gamma_{\mathrm{Ш}}^{(1)}, \Gamma_{\mathrm{Ш}}^{(2)}, \dots, \Gamma_{\mathrm{Ш}}^{(m)}$. Блоки зашифрованных данных $\mathrm{Т}_{\mathrm{Ш}}^{(1)}, \mathrm{Т}_{\mathrm{Ш}}^{(2)}, \dots, \mathrm{Т}_{\mathrm{Ш}}^{(m)}$ суммируются поразрядно по модулю 2 в сумматоре CM_5 с блоками гаммы шифра $\Gamma_{\mathrm{Ш}}^{(1)}, \Gamma_{\mathrm{Ш}}^{(2)}, \dots, \Gamma_{\mathrm{Ш}}^{(m)}$. В результате получаются блоки открытых данных $\mathrm{Т}_{0}^{(1)}, \mathrm{Т}_{0}^{(2)}, \dots, \mathrm{Т}_{0}^{(m)}$,при этом $\mathrm{Т}_{0}^{(m)}$может содержать меньше 64 разрядов.

10.3. Режим гаммирования с обратной связью

Криптосхема, реализующая алгоритм шифрования в режиме гаммирования с обратной связью, имеет вид, показанный на рис. 10.3.

Шифрование открытых данных в режиме гаммирования с обратной связью

Открытые данные, разбитые на 64-разрядные блоки $\mathrm{Т}_{0}^{(1)}, \mathrm{Т}_{0}^{(2)}, \dots, \mathrm{Т}_{0}^{(m)}$, шифруются в режиме гаммирования с обратной связью путем поразрядного сложения по модулю 2 с гаммой шифра $\Gamma_{\mathrm{Ш}}$, которая вырабатывается блоками по 64 бита: $\Gamma_{\mathrm{Ш}}^{(1)}, \Gamma_{\mathrm{Ш}}^{(2)}, \dots, \Gamma_{\mathrm{Ш}}^{(m)}$.

Число двоичных разрядов в блоке $\mathrm{Т}_{0}^{(m)}$ может быть меньше 64, при этом неиспользованная для шифрования часть гаммы шифра из блока $\Gamma_{\mathrm{Ш}}^{(m)}$ отбрасывается.

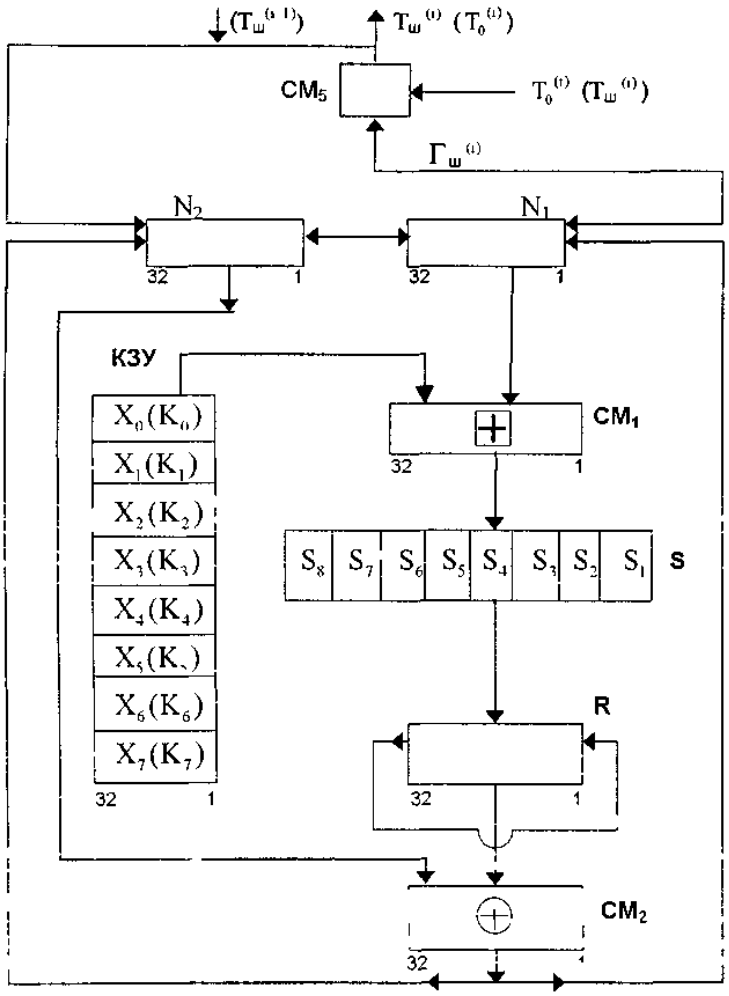


Рис. 10.3. Схема реализации режима гаммирования с обратной связью

Уравнения шифрования в режиме гаммирования с обратной связью имеют вид:

$$\mathrm{T}_{\mathrm{Ш}}^{(1)} = \mathrm{A}(\tilde{\mathrm{S}}) \oplus \mathrm{T}_{0}^{(1)} = \Gamma_{\mathrm{Ш}}^{(1)} \oplus \mathrm{T}_{0}^{(1)},$$

$$\mathrm{T}_{\mathrm{Ш}}^{(i)} = \mathrm{A}\left(\mathrm{T}_{\mathrm{Ш}}^{(i-1)}\right) \oplus \mathrm{T}_{0}^{(i)} = \Gamma_{\mathrm{Ш}}^{(i)} \oplus \mathrm{T}_{0}^{(i)}, i = 2 \ldots m.$$

Здесь $\mathrm{T}_{\mathrm{Ш}}^{(i)}$- i-й 64-разрядный блок шифрованного текста; $\mathrm{A}(*)$ - функция шифрования в режиме простой замены; mопределяется объемом открытых данных.

Аргументом функции $\mathrm{A}(*)$ на первом шаге итеративного алгоритма является 64-разрядная синхропосылка S, а на всех последующих шагах - предыдущий блок зашифрованных данных $\mathrm{T}_{\mathrm{Ш}}^{(i-1)}$.

Процедура шифрования данных в режиме гаммирования с обратной связью реализуется следующим образом. В КЗУ вводятся 256 бит ключа,в накопители N_1 и N_2 вводится синхропосылка $\tilde{\mathrm{S}} = (S_1, S_2, \ldots, S_{64})$ из 64 бит. Исходное заполнение накопителей N_1 и N_2 шифруется в режиме простой замены. Полученное в результате шифрования заполнение накопителей N_1 и

N_2 образует первый 64-разрядный блок гаммы шифра $\Gamma_{Ш}^{(1)} = \grave{A}(\tilde{S})$, который суммируется поразрядно по модулю 2 в сумматоре CM_5 с первым 64-разрядным блоком открытых данных

$$T_0^{(1)} = \left(t_1^{(1)}, t_2^{(1)}, \dots, t_{63}^{(1)}, t_{64}^{(1)}\right).$$

В результате получают первый 64-разрядный блок шифрованных данных

$$T_{Ш}^{(1)} = \Gamma_{Ш}^{(1)} \oplus T_0^{(1)},$$

где $T_{Ш}^{(1)} = \left(\tau_1^{(1)}, \tau_2^{(1)}, \dots, \tau_{63}^{(1)}, \tau_{64}^{(1)}\right)$.

Блок шифрованных данных $T_{Ш}^{(i)}$одновременно является также исходным состоянием накопителей N_1 и N_2 для выработки второго блока гаммы шифра $\Gamma_{Ш}^{(2)}$, и поэтому по обратной связи $T_{Ш}^{(1)}$ записывается в указанные накопители N_1 и N_2 .

Заполнение накопителя N_1:

$$\left(\tau_{32}^{(1)}, \tau_{31}^{(1)}, \dots, \tau_2^{(1)}, \tau_1^{(1)}\right)$$

$$32,\ 31, \dots, 2,\ 1 \leftarrow \text{номер разряда } N_1$$

Заполнение накопителя N_2 :

$$\left(\tau_{64}^{(1)}, \tau_{63}^{(1)}, \dots, \tau_{34}^{(1)}, \tau_{33}^{(1)}\right)$$

$$32,\ 31, \dots, 2,\ 1 \leftarrow \text{номер разряда } N_1$$

Заполнение накопителей N_1 и N_2 шифруется в режиме простой замены. Полученное в результате шифрования заполнение накопителей N_1 и N_2 образует второй 64-разрядный блок гаммы шифра $\Gamma_{Ш}^{(2)}$, который суммируется поразрядно по модулю 2 в сумматоре CM_5 со вторым блоком открытых данных $T_0^{(2)}$:

$$T_{Ш}^{(2)} = \Gamma_{Ш}^{(2)} \oplus T_0^{(2)}$$

Выработка последующих блоков гаммы шифра $\Gamma_{Ш}$ и шифрование соответствующих блоков открытых данных $T_0^{(i)}$ $(i = 3 \dots m)$ производятся аналогично. Если длина последнего m-го блока открытых данных $T_0^{(m)}$ меньше 64 разрядов, то из $\Gamma_{Ш}^{(m)}$ используется только соответствующее число разрядов гаммы шифра, остальные разряды отбрасываются.

В канал связи или память ЭВМ передаются синхропосылка $\tilde{S}$ и блоки зашифрованных данных $T_{Ш}^{(1)}, T_{Ш}^{(2)}, \dots, T_{Ш}^{(m)}$.

Расшифрование в режиме гаммирования с обратной связью

При расшифровывании криптосхема имеет тот же вид, что и при шифровании (см. рис.10 3).

Уравнения расшифрования:

$$T_0^{(1)} = A(\tilde{S}) \oplus T_{Ш}^{(1)} = \Gamma_{Ш}^{(1)} \oplus T_{Ш}^{(1)},$$

$$T_0^{(i)} = \Gamma_{Ш}^{(i)} \oplus T_{Ш}^{(i)} = A\left(T_{Ш}^{(i-1)}\right) \oplus T_{Ш}^{(i)}, i = 2 \dots m.$$

Реализация процедуры расшифрования шифрованных данных в режиме гаммирования с обратной связью происходит следующим образом. В КЗУ вводят 256 бит того же ключа, на котором осуществлялось шифрование открытых блоков $\mathrm{T}_0^{(1)}, \mathrm{T}_0^{(2)}, \ldots, \mathrm{T}_0^{(\mathrm{m})}$. В накопители N_1 и N_2 вводится синхропосылка $\tilde{\mathrm{S}}$. Исходное заполнение накопителей N_1 и N_2 (синхропосылка $\tilde{\mathrm{S}}$) шифруется в режиме простой замены. Полученное в результате шифрования заполнение N_1 и N_2 образует первый блок гаммы шифра

$$\Gamma_{\text{Ш}}^{(1)} = \mathrm{A}(\tilde{\mathrm{S}}),$$

который суммируется поразрядно по модулю 2 в сумматоре CM_5 с блоком шифрованных данных $\mathrm{T}_{\text{Ш}}^{(1)}$. В результате получается первый блок открытых данных

$$\mathrm{T}_0^{(2)} = \Gamma_{\text{Ш}}^{(2)} \oplus \mathrm{T}_{\text{Ш}}^{(2)}.$$

Блок шифрованных данных $\mathrm{T}_{\text{Ш}}^{(1)}$ является исходным заполнением накопителей N_1 и N_2 для выработки второго блока гаммы шифра $\Gamma_{\text{Ш}}^{(2)}$: $\Gamma_{\text{Ш}}^{(2)} = \mathrm{A}(\mathrm{T}_{\text{Ш}}^{(1)})$,. Полученное заполнение накопителей N_1 и N_2 шифруется в режиме простой замены. Образованный в результате шифрования блок $\Gamma_{\text{Ш}}^{(2)}$ суммируется поразрядно по модулю 2 в сумматоре CM_5 со вторым блоком шифрованных данных $\mathrm{T}_{\text{Ш}}^{(2)}$. В результате получают второй блок открытых данных. Аналогично в N_1 и N_2 последовательно записывают блоки шифрованных данных $\mathrm{T}_{\text{Ш}}^{(1)}, \mathrm{T}_{\text{Ш}}^{(2)}, \ldots, \mathrm{T}_{\text{Ш}}^{(\mathrm{m})}$, из которых в режиме простой замены вырабатываются блоки гаммы шифра $\Gamma_{\text{Ш}}^{(3)}, \Gamma_{\text{Ш}}^{(4)}, \ldots, \Gamma_{\text{Ш}}^{(m)}$. Блоки гаммы шифра суммируются поразрядно по модулю 2 в сумматоре CM_5 с блоками шифрованных данных $\mathrm{T}_{\text{Ш}}^{(3)}, \mathrm{T}_{\text{Ш}}^{(4)}, \ldots, \mathrm{T}_{\text{Ш}}^{(\mathrm{m})}$.

В результате получают блоки открытых данных $\mathrm{T}_0^{(3)}, \mathrm{T}_0^{(4)}, \ldots, \mathrm{T}_0^{(\mathrm{m})}$, при этом последний блок открытых данных $\mathrm{T}_0^{(\mathrm{m})}$ может содержать меньше 64 разрядов.

10.4. Режим выработки имитовставки

Имитовставка - это блок из P бит, который вырабатывают по определенному правилу из открытых данных с использованием ключа и затем добавляют к зашифрованным данным для обеспечения их имитозащиты.

Имитозащита - это защита системы шифрованной связи от навязывания ложных данных.

В стандарте ГОСТ 28147-89 определяется процесс выработки имитовставки, который единообразен для любого из режимов шифрования данных. Имитовставка И_P вырабатывается из блоков открытых данных либо перед шифрованием всего сообщения, либо параллельно с шифрованием по блокам. Первые блоки открытых данных, которые участвуют в выработке имитовставки, могут содержать служебную информацию (например, адресную часть, время, синхропосылку) и не шифруются.

Значение параметра P (число двоичных разрядов в имитовставке) определяется криптографическими требованиями с учетом того, что вероятность навязывания ложных помех равна $1/2^P$.

Для выработки имитовставки открытые данные представляют в виде последовательности 64-разрядных блоков $\mathrm{Т}_O^{(i)}$, $i = 1 \dots m$. Первый блок открытых данных $\mathrm{Т}_O^{(1)}$ подвергают преобразованию $\tilde{A}(*)$, соответствующему первым 16 циклам алгоритма шифрования в режиме простой замены. В качестве ключа для выработки имитовставки используют ключ длиной 256 бит, по которому шифруют данные.

Полученное после 16 циклов 64-разрядное число $\tilde{A}(\mathrm{Т}_O^{(1)})$ суммируют по модулю 2 со вторым блоком открытых данных $\mathrm{Т}_O^{(2)}$. Результат суммирования $\tilde{A}(T_O^{(1)}) \oplus \mathrm{Т}_O^{(2)}$ снова подвергают преобразованию $\tilde{A}(*)$.

Полученное 64-разрядное число $\tilde{A}(\tilde{A}(\mathrm{Т}_O^{(1)}) \oplus \mathrm{Т}_O^{(2)})$ суммируют по модулю 2 с третьим блоком $\mathrm{Т}_O^{(3)}$ и снова подвергают преобразованию $\tilde{A}(*)$, получая 64-разрядное число $\tilde{A}(\tilde{A}(\tilde{A}(\mathrm{Т}_O^{(1)}) \oplus \mathrm{Т}_O^{(2)}) \oplus \mathrm{Т}_O^{(3)})$, и т.д.

Последний блок $\mathrm{Т}_O^{(m)}$ (при необходимости дополненный нулями до полного 64-разрядного блока) суммируют по модулю 2 с результатом вычислений на шаге $(m-1)$, после чего шифруют в режиме простой замены, используя преобразование $\tilde{A}(*)$.

Из полученного 64-разрядного числа выбирают отрезок $\mathrm{И}_P$ (имитовставку) длиной P бит:

$$\mathrm{И}_P = [a_{32-P+1}^m(16), a_{32-P+2}^m(16), \dots, a_{32}^m(16)],$$

где a_i^m- i -тый бит 64-разрядного числа, полученного после 16-го цикла последнего преобразования $\tilde{A}(*)$,$32 - P + 1 \le i \le 31$.

Имитовставка $\mathrm{И}_P$ передается по каналу связи в конце шифрованных данных, т.е.

$$\mathrm{Т}_{\mathrm{Ш}}^{(1)}, \mathrm{Т}_{\mathrm{Ш}}^{(2)}, \dots, \mathrm{Т}_{\mathrm{Ш}}^{(m)}, \mathrm{И}_P.$$

Поступившие к получателю шифрованные данные $\mathrm{Т}_{\mathrm{Ш}}^{(1)}, \mathrm{Т}_{\mathrm{Ш}}^{(2)}, \dots, \mathrm{Т}_{\mathrm{Ш}}^{(m)}$ расшифровываются, и из полученных блоков открытых данных $\mathrm{Т}_O^{(1)}, \mathrm{Т}_O^{(2)}, \dots, \mathrm{Т}_O^{(m)}$ аналогичным образом вырабатывается имитовставка $\mathrm{И}'_P$, которая сравнивается с $\mathrm{И}_P$. В случае несовпадения блок открытых данных считается ложным.

Лекция 11. Стандарт шифрования данных *DES*

11.1. Обобщенная схема алгоритма *DES*

Алгоритм *DES* использует комбинацию подстановок и перестановок. *DES* осуществляет шифрование 64-битовых блоков данных с помощью 64-битого ключа, в котором значащими являются 56 бит (остальные 8 бит - проверочные биты для контроля на четность). Расшифрование в *DES* является операцией, обратной шифрованию, и выполняется путем повторения операций шифрования в обратной последовательности.

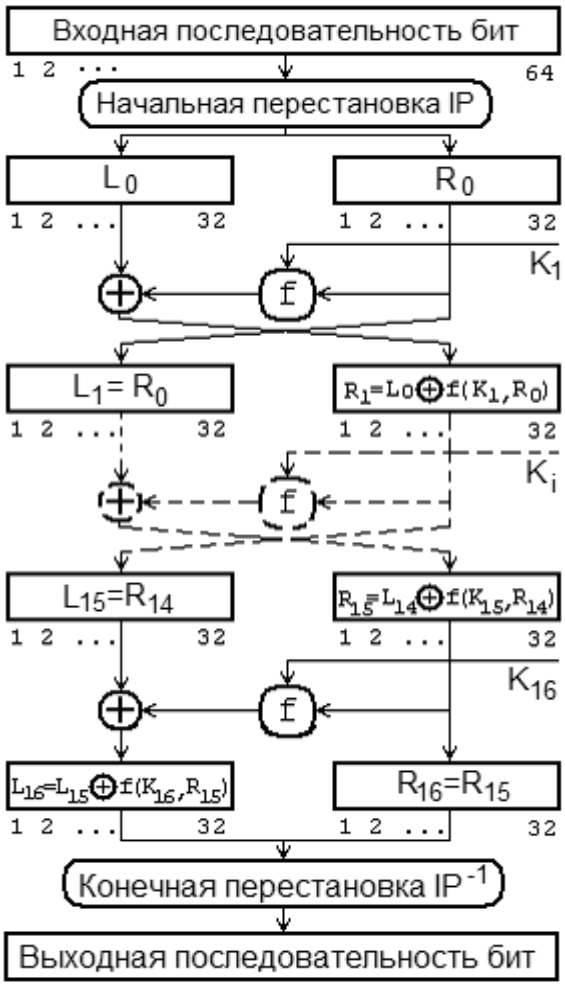


Рис. 11.1. Обобщенная схема шифрования в алгоритме DES

Обобщенная схема процесса шифрования в алгоритме *DES* (рис. 11.1) заключается в начальной перестановке бит 64-битого блока, шестнадцати циклах шифрования и, наконец, в конечной перестановке бит.

Следует отметить, что все приводимые таблицы являются стандартными и должны включаться в реализацию алгоритма *DES* в неизменном виде. Все перестановки и коды в таблицах подобраны разработчиками таким образом, чтобы максимально затруднить процесс взлома шифра.

Пусть из файла исходного текста считан очередной 64-битовый блок T. Этот блок преобразуется с помощью матрицы начальной перестановки IP (табл. 11.1).

Таблица 11.1

Начальная перестановка IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Биты входного блока T (64 бита) переставляются в соответствии с матрицей IP: бит 58 входного блока T становится битом 1, бит 50 - битом 2 и т.д. Эту перестановку можно описать выражением $T_0 = IP(T)$. Полученная последовательность бит T разделяется на две последовательности: L_0 - левые, или старшие, биты, R_0 - правые, или младшие, биты - каждая из которых содержит 32 бита.

Затем выполняется итеративный процесс шифрования, состоящий из 16 шагов (циклов). Пусть T_i - результат i-й итерации: $T_i = L_i R_i$, где $L_i = t_1 t_2 \dots t_{32}$ (первые 32 бита); $R_i = t_{33} t_{34} \dots t_{64}$ (последние 32 бита). Тогда результат -й итерации описывается следующими формулами:

$$L_i = R_{i-1}, i = 1,2, \dots ,16;$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i), i = 1,2, \dots ,16.$$

Функция f называется функцией шифрования. Ее аргументами являются последовательность R_{i-1}, получаемая на предыдущем шаге итерации, и 48-битовыйключ K_i, который является результатом преобразования 64-битого ключа шифра K. (Подробнее функция шифрования f и алгоритм получения ключа K описаны ниже.)

На последнем шаге итерации получают последовательности R_{16} и L_{16} (без перестановки местами), которые конкатенируются в 64-битую последовательность $R_{16}L_{16}$.

По окончании шифрования осуществляется восстановление позиций бит с помощью матрицы обратной перестановки IP^{-1} (табл. 11.2).

Таблица 11.2

Обратная перестановка IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Процесс расшифрования данных является инверсным по отношению к процессу шифрования. Все действия должны быть выполнены в обратном порядке. Это означает, что расшифровываемые данные сначала переставляются в соответствии с матрицей IP^{-1}, а затем над последовательностью бит $R_{16}L_{16}$ выполняются те же действия, что и в процессе шифрования, но в обратном порядке.

Итеративный процесс расшифрования может быть описан следующими формулами:

$$R_{i-1} = L_i, i = 1,2, \dots ,16;$$
$$L_{i-1} = R_i \oplus f(L_i, K_i), i = 1,2, \dots ,16.$$

Таким образом, для процесса расшифрования с переставленным входным блоком $R_{16}L_{16}$ на первой итерации используется ключ K_{16}, на второй итерации - K_{15} и т.д. На 16-й итерации используется ключ K_1. На последнем шаге итерации будут получены последовательности L_0 и R_0, которые конкатенируются в 64-битую последовательность L_0R_0. Затем в этой последовательности 64 бита переставляются в соответствии с матрицей IP. Результат такого преобразования - исходная последовательность бит (расшифрованное 64-битовое значение).

11.2. Реализация функции шифрования

Схема вычисления функции шифрования $f(R_{i-1}, K_i)$ показана на рис. 5.2.

Для вычисления значения функции f используются:

- функция E (расширение 32 бит до 48);
- функция $S_1, S_2, \dots , S_8$ (преобразование 6-битового числа в 4-битовое);
- функция P (перестановка бит в 32-битовой последовательности).

Приведем определения этих функций.

Аргументами функции шифрования f являются R_{i-1} (32 бита) и K_i (48 бит). Результат функции $E(R_{i-1})$ есть 48-битовое число. Функция расширения E, выполняющая расширение 32 бит до 48 (принимает блок из 32 бит и порождает блок из 48 бит), определяется табл. 11.3.

В соответствии с табл. 11.3 первые три бита $E(R_{i-1})$ - это биты 32, 1 и 2, а последние - 31, 32 и 1. Полученный результат (обозначим его $E(R_{i-1})$) складывается по модулю 2 с текущим значением ключа K_i и затем разбивается на восемь 6-битовых блоков $B_1, B_2, \dots, B_8 = E(R_{i-1}) \oplus K_i$. Далее каждый из этих блоков используется как номер элемента в функциях – матрицах $S_1, S_2, \dots, S_8$, содержащих 4-битовые значения (табл. 11.4).

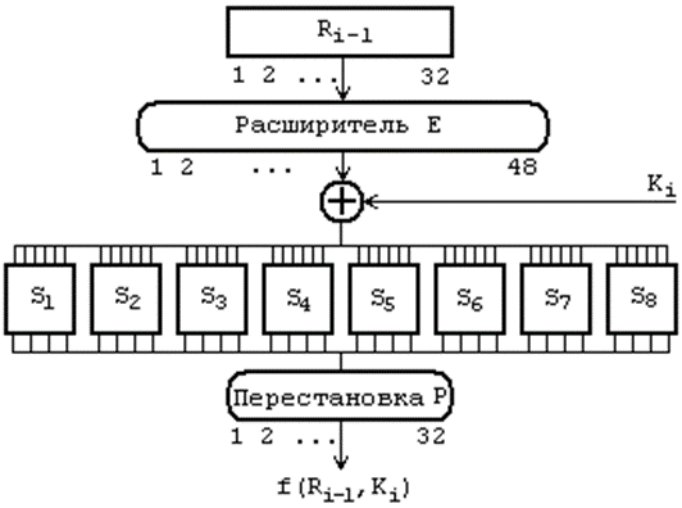


Рис. 11.2. Схема вычисления функции шифрования f

Таблица 11.3

Функция *E*					
32	*1*	*2*	*3*	*4*	*5*
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Следует отметить, что выбор элемента в матрице S осуществляется достаточно оригинальным образом. Пусть на вход матрицы S поступает 6-битовый блок $B_j = b_1 b_2 b_3 b_4 b_5 b_6$, тогда 2-битовое число $b_1 b_6$ указывает номер строки матрицы, а 4-битовое число $b_2 b_3 b_4 b_5$ - номер столбца. Например, если на вход матрицы S_1 поступает 6-битовый блок $B_1 = b_1 b_2 b_3 b_4 b_5 b_6 = 100110_{(2)}$, то 2-битовое число $b_1 b_6 = 10_{(2)} = 2_{(2)}$ указывает строку с номером 2 матрицы S_1,а 4-битовое число $b_2 b_3 b_4 b_5 = 0011_{(2)} = 3_{(10)}$ указывает столбец с номером 3 матрицы S_1. Это означает, что в матрице S_1 блок $B_1 = 100110$ выбирает элемент

на пересечении строки с номером 2 и столбца с номером 3, т.е. элемент $S_{(10)} = 1000_{(2)}$. Совокупность 6-битовых блоков $B_1, B_2, \dots, B_8$ обеспечивает выбор 4-битового элемента в каждой из матриц $S_1, S_2, \dots, S_8$.

Таблица 11.4

Функции *S*																	
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_1	**0**	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	4	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	**0**	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	**0**	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	**0**	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	**0**	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	**0**	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	1	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	**0**	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	**0**	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

В результате получаем $S_1(B_1), S_2(B_1), \dots, S_8(B_1)$, т.е. 32-битовый блок (поскольку матрицы S, содержат 4-битовые элементы). Этот 32-битовый блок преобразуется с помощью функции перестановки бит P (табл. 11.5).

Таблица 11.5

Функция P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Таким образом, функция шифрования

$$f(R_{i-1}, K_i) = P\big(S_1(B_1), \dots, S_8(B_1)\big).$$

11.3. Алгоритм вычисления ключей

Как нетрудно заметить, на каждой итерации используется новое значение ключа K_i (длиной 48 бит). Новое значение ключа K_i вычисляется из начального ключа K (рис. 11.3).

Ключ K представляет собой 64-битовый блок с 8 битами контроля по четности, расположенными в позициях 8, 16, 24, 32, 40, 48, 56, 64. Для удаления контрольных бит и подготовки ключа к работе используется функция G первоначальной подготовки ключа (табл. 11.6).

Таблица 11.6

	Функция G						
C_0	57	49	41	33	25	17	9
	1	58	50	42	34	26	18
	10	2	59	51	43	35	27
	19	11	3	60	52	44	36
D_0	63	55	47	39	31	23	15
	7	62	54	46	38	30	22
	14	6	61	53	45	37	29
	21	13	5	28	20	12	4

Табл. 11.6 разделена на две части. Результат преобразования $G(K)$ разбивается на две половины C_0 и D_0, по 28 бит каждая. Первые четыре строки матрицы G определяют, как выбираются биты последовательности C (первым битом C_0 будет бит 57 ключа шифра, затем бит 49 и т.д., а последними битами - биты 44 и 36 ключа).

Следующие четыре строки матрицы G определяют, как выбираются биты последовательности D_0 (т.е. последовательность D_0 будет состоять из бит 63, 55, 47,...,12, 4 ключа шифра).

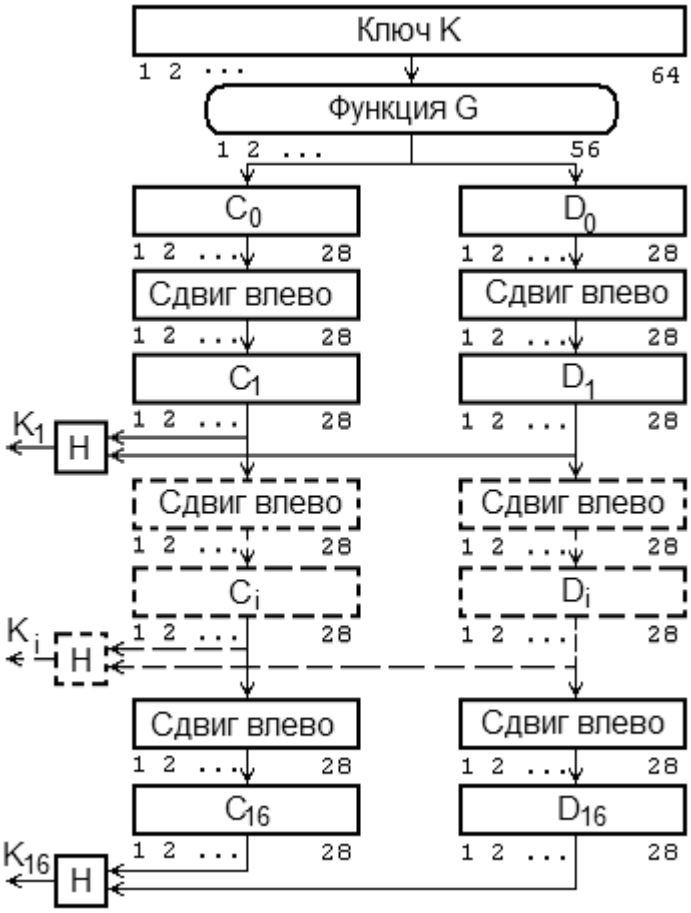


Рис. 11.3. Схема алгоритма вычисления ключей K_i

Как видно из табл. 11.6, для генерации последовательностей C_0 и D_0 не используются биты 8, 16, 24, 32, 40, 48, 56 и 64 ключа шифра. Эти биты не влияют на шифрование и могут служить для других целей (например, для контроля по четности). Таким образом, в действительности ключ шифра является 56-битовым.

После определения C_0 и D_0 рекурсивно определяются C_i и D_i, $i = 1, 2, \ldots, 16$. Для этого применяются операции циклического сдвига влево на один или два бита в зависимости от номера шага итерации, как показано в Таблице сдвигов для вычисления ключа (табл. 11.7.)

Таблица 11.7

Итерация	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Сдвиг влево	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Операции сдвига выполняются для последовательностей C_i и D_i независимо. Например, последовательность C_3 получается посредством

циклического сдвига влево на две позиции последовательности C_2, а последовательность D_3 посредством сдвига влево на две позиции последовательности D_2, C_{16} и D_{16} получаются из C_{15} и D_{15} посредством сдвига влево на одну позицию.

Ключ K_i, определяемый на каждом шаге итерации, есть результат выбора конкретных бит из 56-битовой последовательности C_iD_i и их перестановки. Другими словами, ключ $K_i = H(C_iD_i)$, где функция H определяется матрицей, завершающей обработку ключа (табл. 11.8).

Таблица 11.8

Функция H					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	22	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Как следует из табл. 11.8, первым битом ключа K_i будет 14-й бит последовательности C_iD_i, вторым - 17-й бит, 47-м битом ключа K_i будет 29-й бит C_iD_i, а 48-м битом - 32-й бит C_iD_i.

11.4. Основные режимы работы алгоритма DES

Чтобы воспользоваться алгоритмом DES для решения разнообразных криптографических задач, разработаны четыре рабочих режима:

1. Электронная кодовая книга $ECB(Electronic\ Code\ Book)$;
2. Сцепление блоков шифра CBC $(Cipher\ Block\ Chaining)$;
3. Обратная связь по шифротексту CBP $(Cipher\ FeedBack)$;
4. Обратная связь по выходу OFB $(Output\ FeedBack)$.

Режим "Электронная кодовая книга"

Длинный файл разбивают на 64-битые отрезки (блоки) по 8 байт. Каждый из этих блоков шифруют независимо с использованием одного и того же ключа шифрования (рис. 11.4).

Основным достоинством является простота реализации. К недостатку можно отнести относительно слабую устойчивость против квалифицированных криптоаналитиков. Из-за фиксированного характера шифрования при ограниченной длине блока 64 бита возможно проведение криптоанализа "со словарем". Блок такого размера может повториться в сообщении вследствие большой избыточности в тексте на естественном языке. Это приводит к тому, что идентичные блоки открытого текста в сообщении будут представлены идентичными блоками шифротекста, что дает криптоаналитику некоторую информацию о содержании сообщения.

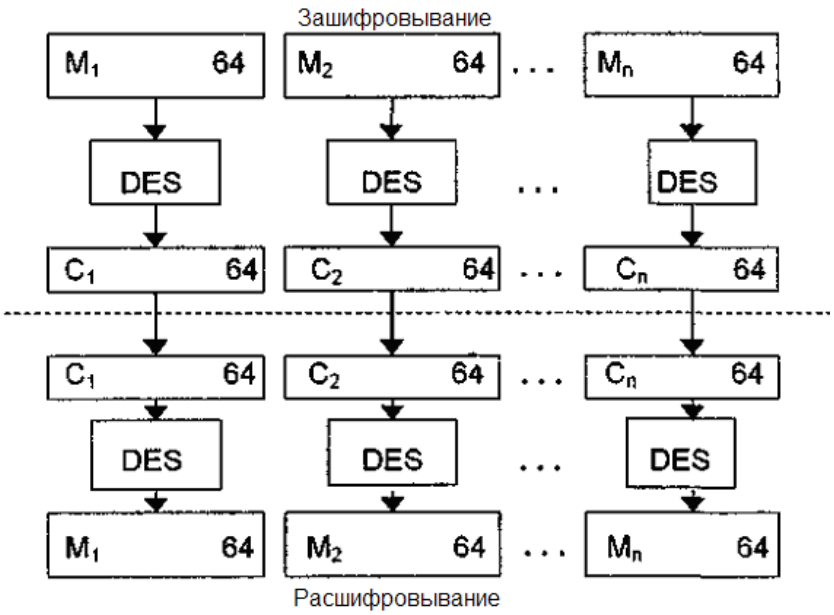


Рис. 11.4. Схема алгоритма ***DES*** в режиме электронной кодовой книги

Режим "Сцепление блоков шифра"

В этом режиме исходный файл M разбивается на 64-битые блоки: $M = M_1 M_2 \dots M_n$. Первый блок M_1 складывается по модулю 2 с 64-битовым начальным вектором IV, который меняется ежедневно и держится в секрете (рис. 5.5). Полученная сумма затем шифруется с использованием ключа DES, известного и отправителю, и получателю информации. Полученный 64-битовый шифр C_1 складывается по модулю 2 со вторым блоком текста, результат шифруется и получается второй 64-битовый шифр C_2 и т.д. Процедура повторяется до тех пор, пока не будут обработаны все блоки текста.

Таким образом, для всех $i = 1 \dots n$ (n - число блоков) результат шифрования C определяется следующим образом: $C_i = DES(M_i \oplus C_{i-1})$, где $C_0 = IV$ - начальное значение шифра, равное начальному вектору (вектору инициализации).

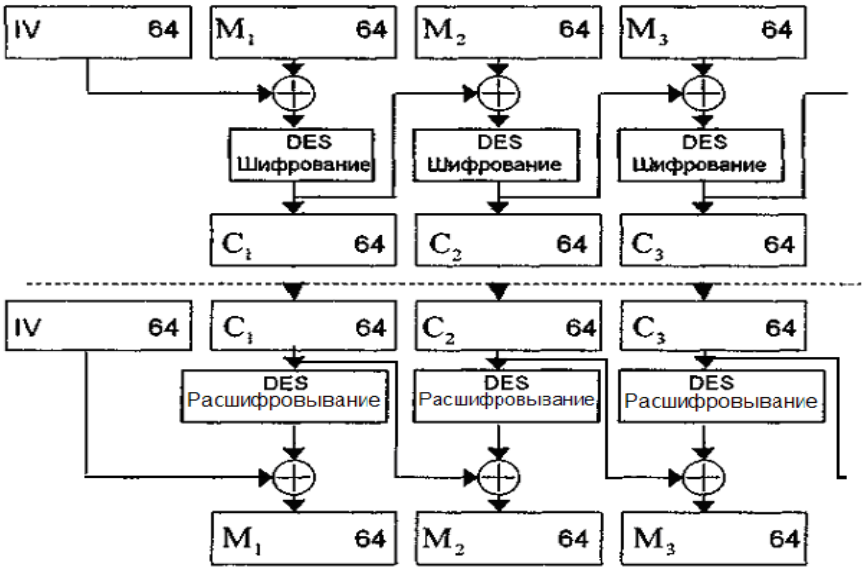


Рис. 11.5. Схема алгоритма DES в режиме сцепления блоков шифра

Очевидно, что последний 64-битовый блок шифротекста является функцией секретного ключа, начального вектора и каждого бита открытого текста независимо от его длины. Этот блок шифротекста называют кодом аутентификации сообщения (КАС).

Код КАС может быть легко проверен получателем, владеющим секретным ключом и начальным вектором, путем повторения процедуры, выполненной отправителем. Посторонний, однако, не может осуществить генерацию КАС, который воспринялся бы получателем как подлинный, чтобы добавить его к ложному сообщению, либо отделить КАС от истинного сообщения для использования его с измененным или ложным сообщением.

Достоинство данного режима в том, что он не позволяет накапливаться ошибкам при передаче.

Блок M_i является функцией только C_{i-1} и C_i. Поэтому ошибка при передаче приведет к потере только двух блоков исходного текста.

Режим "Обратная связь по шифротексту"

В этом режиме размер блока может отличаться от 64 бит (рис 11.6). Файл, подлежащий шифрованию (расшифровыванию), считывается последовательными блоками длиной k бит ($k = 1 \ldots 64$).

Входной блок (64-битовый регистр сдвига) вначале содержит вектор инициализации, выровненный по правому краю. Предположим, что в результате разбиения на блоки мы получили n блоков длиной k бит каждый (остаток дописывается нулями или пробелами). Тогда для любого $i = 1 \ldots n$ блок

шифротекста $C_i = M_i \oplus \mathrm{P}_{i-1}$, где P_{i-1} обозначает k старших бит предыдущего зашифрованного блока.

Обновление сдвигового регистра осуществляется путем удаления его старших k бит и записи C_i в регистр. Восстановление зашифрованных данных также выполняется относительно просто: P_{i-1} и C_i вычисляются аналогичным образом и $M_i = C_i \oplus P_{i-1}$.

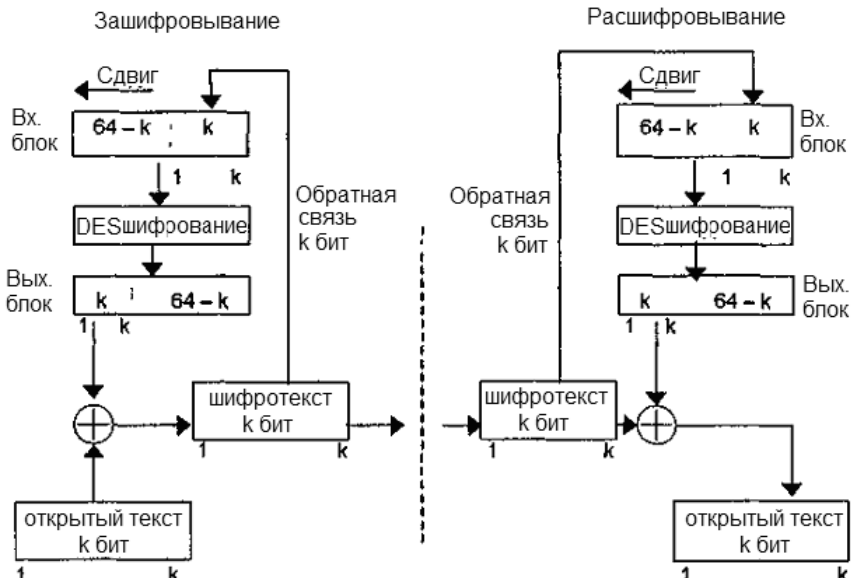


Рис. 11.6. Схема алгоритма DES в режиме обратной связи по шифротексту

Режим "Обратная связь по выходу"

Этот режим тоже использует переменный размер блока и сдвиговый регистр, инициализируемый так же, как в режиме CBP, а именно - входной блок вначале содержит вектор инициализации IV, выровненный по правому краю (рис. 11.7). При этом для каждого сеанса шифрования данных необходимо использовать новое начальное состояние регистра, которое должно пересылаться по каналу открытым текстом.

Положим $M = M_1, M_2, \dots, M_n$ для всех $i = 1 \dots n$ $C_i = M_i \oplus P_i$, где P_i - старшие k бит операции $DES(C_{i-1})$. Отличие от режима обратной связи по шифротексту состоит в методе обновления сдвигового регистра.

Это осуществляется путем отбрасывания старших k бит и дописывания справа P_i.

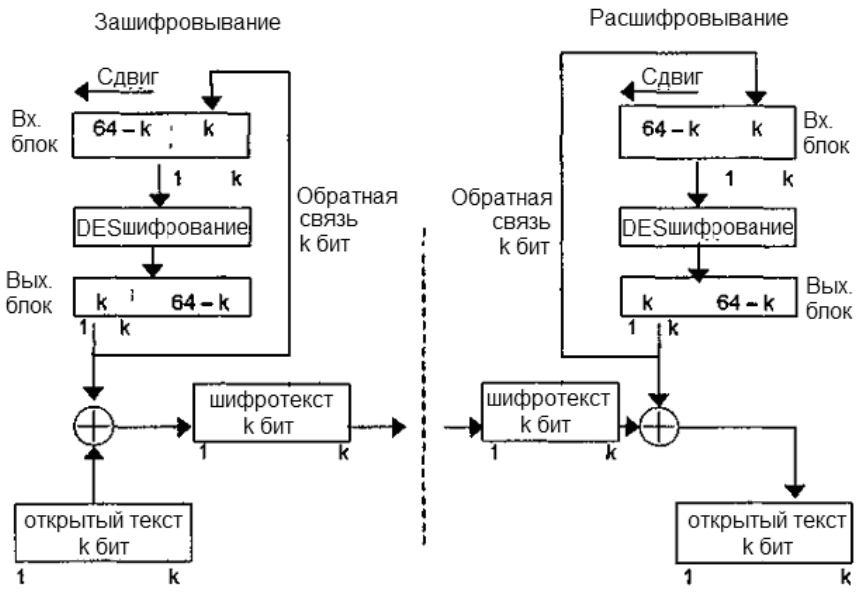


Рис. 11.7. Схема алгоритма DES в режиме обратной связи по выходу

Лекция 12. Асимметричные криптосистемы

12.1. Концепция криптосистемы с открытым ключом

Эффективными системами криптографической защиты данных являются асимметричные криптосистемы, называемые также криптосистемами с открытым ключом. В таких системах для шифрования данных используется один ключ, а для расшифрования другой (отсюда и название - асимметричные). Первый ключ является открытым и может быть опубликован для использования всеми пользователями системы, которые шифруют данные. Расшифрование данных с помощью открытого ключа невозможно. Для расшифрования данных получатель зашифрованной информации использует второй ключ, который является секретным. Разумеется, ключ расшифрования не может быть определен из ключа шифрования.

Обобщенная схема асимметричной криптосистемы с открытым ключом показана на рис. 12.1.

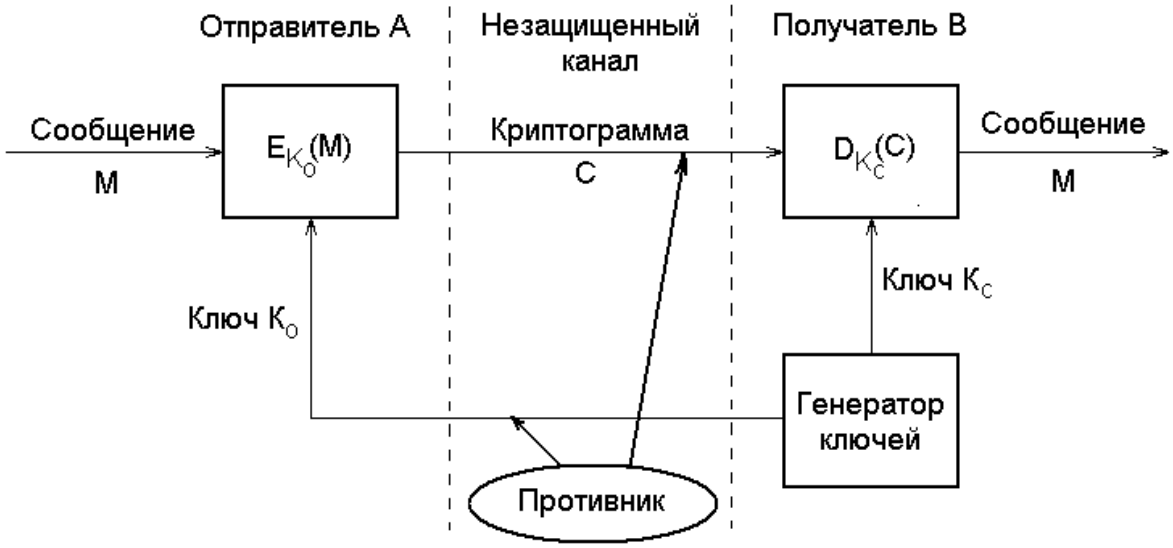


Рис. 12.1. Обобщенная схема асимметричной криптосистемы с открытым ключом

В этой криптосистеме применяют два различных ключа: K_0 - открытый ключ отправителя А; K_c - секретный ключ получателя В. Генератор ключа целесообразно располагать на стороне получателя В (чтобы не пересылать секретный ключ K_c по незащищенному каналу). Значения ключей K_0, K_c – зависят от начального состояния генератора ключей.

Раскрытие секретного ключа K_c по известному ключу K_0 должно быть вычислительно неразрешимой задачей.

Характерные особенности асимметричных криптосистем:

1. Открытый ключ K_0, и криптограмма Cмогут быть отправлены по незащищенному каналу, т.е. могут быть известны противнику.

2. Алгоритмы шифрования $E_{K_0}(M) \to C$ и расшифрования $D_{K_C}(C) \to M$ являются открытыми.

3. Защита информации в асимметричной криптосистеме основана на секретности ключа K_c.

4. У.Диффи и М.Хелман сформулировали требования, выполнение которых обеспечивает безопасность асимметричной криптосистемы:

5. Вычисление пары ключей (K_0, K_c) получателем В на основе начального условия должно быть простым.

6. Отправитель А, зная открытый ключ K_0 и сообщение M , может легко вычислить криптограмму $C = E_{K_0}(M)$.

7. Получатель В, используя секретный ключ K_c и криптограмму С, может легко восстановить исходное сообщение $M = D_{K_C}(C) = D_{K_C}(E_{K_0}(M))$.

8. Противник, зная открытый ключ K_c, при попытке вычислить секретный ключ K_c наталкивается на непреодолимую вычислительную проблему.

9. Противник, зная пару (K_0, C), при попытке вычислить исходное сообщение M, наталкивается на непреодолимую вычислительную проблему.

12.2. Однонаправленные функции

Концепция асимметричных криптосистем с открытым ключом основана на применении однонаправленных функций. Однонаправленную функцию (ОФ) можно определить следующим образом.

Пусть X и Y- некоторые производные множества. Функция $f: X \to Y$ является однонаправленной, если для всех $x \in X$ можно легко вычислить функцию $y = f(x)$, где $y \in Y$. И в тоже время для большинства $y \in Y$ достаточно сложно получить значение $x \in X$, такое, что $f(x) = y$ (при этом полагаем, что существует по крайней мере одно такое значение x).

Основным критерием отнесения функции f к классу ОФ является отсутствие эффективных алгоритмов обратного преобразования $Y \to X$. Примеры однонаправленных функций.

1. Целочисленное произведение двух больших чисел.

Прямое преобразование - вычисление произведения двух очень больших чисел P и Q, т.е. нахождение значения $N = P \cdot Q$, является относительно несложной задачей для ЭВМ. Обратное преобразование – разложение на множители большого целого числа, т.е. нахождение деталей P и Q большого $N = P \cdot Q$, является практически неразрешенной задачей при достаточно больших значениях N. По оценке теории чисел для разложения целого $N \approx 2^{664}$ потребуется около 10^{23} операций т.е. задача практически неразрешима для ЭВМ.

2. Модульная экспонента с фиксированным основанием и модулем.

Пусть A и N - целые числа, такие, что $1 \leq A \leq N$. Модульная экспонента с основанием A по модулю N представляет собой функцию $f_{A,N}(x) = A^x mod N$,

где x - целое число, $1 \le x \le N-1$. Существуют эффективные алгоритмы, позволяющие достаточно быстро вычислить значение функции $f_{A,N}(x)$. Обратное преобразование, т.е. нахождение x из соотношения $A^x mod N = Y$ представляет собой трудновыполнимую задачу, т.к. для $y = A^x$ существует обратная функция $x = log_A Y$, то часто нахождение аргумента x по известным y, A и N называют задачей дискретного логарифмирования. Следует иметь ввиду, что $y \in Z_n$, где $Z_n = \{1,2, \dots, N-1\}$.

По оценкам теории чисел при целых числах порядка $A \approx 2^{664}$ и $N \approx 2^{664}$ решением задачи дискретного логарифмирования (нахождение показателя степени x для известного y) потребуется 10^{26} операций. Т.е. для модульной экспоненты на 10^3 сложнее вычислять обратное преобразование, чем для целочисленного произведения. Однако до сих пор не доказано, что не существует эффективного логарифма за приемлемое время. Тем не менее модульная экспонента отнесена к однонаправленным функциям условно и широко используется на практике.

Кроме однонаправленных функций рассмотренного типа применяются однонаправленные функции с секретом (потайным ходом). Функция $f: X \to Y$ относится к классу ОФ с секретом в том случае, если она является однонаправленной и, кроме того, возможно эффективное вычисление обратной функции, если известен “потайной ход” (секретное число, строка, или другая информация о данной функции).

12.3. Элементы теории чисел

Определения

Число a называется простым, если оно не имеет других натуральных делителей, кроме 1 и a.

Например, 17, 23.

Числа a и b называются взаимно простыми, если наибольший общий множитель этих чисел $(a, b) = 1$.

Например: 8 и 9.

Модулярная арифметика

В модулярной арифметике все арифметические действия выполняются как в обычной арифметике с учетом того, что получаемые числа не могут превышать некоторой величины называемой модулем.

$$(3 + 14)\text{mod}12 = 5$$
$$(3 + 14) \equiv 5\text{mod}12$$

В общем случае $a \equiv r \bmod n$. Читается a сравнимо с r по модулю n. Это справедливо, если $a = n \cdot k + r$, где $k = 0,1,2, \dots$.

Отсюда $r = a - n \cdot k$ называется вычетом числа a по модулю n, $(0 \le r < n)$. Справедливо:

$(a \pm b)\text{mod}\, n = (a \bmod n \pm b \bmod n)\text{mod}\, n$;

$(a \cdot b)\text{mod}\, n = (a \bmod n \cdot b \bmod n)\text{mod}\, n$;

$a \cdot (b \pm c)\text{mod}\, n = ((a \cdot b)\text{mod}\, n \pm (a \cdot c) \bmod n)\text{mod}\, n$.

Использование модулярной арифметики позволяет оперировать с очень большими числами, например, при возведении в степень:

$a^8 \bmod n = ((a^2 \bmod n)^2 \bmod n)^2 \bmod n.$

Малая теорема Ферма

Если n - простое и НОД $(a, n) = 1$, то $a^{n-1} \equiv 1 \bmod n$.

Функция Эйлера

Количество положительных целых, меньших n, которые взаимно просты с n, определяется с помощью функции Эйлера $\varphi(n)$:

Модуль	n простое	n^2	n^m	$p \cdot q$ (p и q простые)
$\varphi(n)$	$n-1$	$n(n-1)$	$n^{m-1}(n-1)$	$(p-1)\cdot(q-1)$

Обобщение Эйлера малой теоремы Ферма: если НОД $(a, n) = 1$, то $a^{\varphi(n)} \equiv 1 \bmod n$.

Нахождение обратных величин

Если задано уравнение $(a \cdot x) \bmod n = 1$, то величина $a^{-1} \equiv x \bmod n$ называется обратной величиной a по модулю n.

Обратная величина существует, если a и n – взаимно простые числа.

Способы нахождения обратных чисел

1. Перебором возможных значений.

Подставляя вместо x числа: $1, 2, \ldots, n-1$ – добиваемся выполнения исходного уравнения.

Пример 12.1. $(5 \cdot x) \bmod 7 = 1$, $x = (5^{-1}) \bmod 7 = 3$, т.к. $(5 \cdot 3) \bmod 7 = 15 \bmod 7 = (15 - 7 \cdot 2) \bmod 7 = 1$.

2. С помощью функции Эйлера $\varphi(n)$.

$(a^{-1}) \bmod n = (a^{\varphi(n)-1}) \bmod n$.

Пример 12.2.

$x = (5^{-1}) \bmod 7 = (5^{6-1}) \bmod 7 = ((5^2) \bmod 7 \cdot (5^3) \bmod 7) \bmod 7 = (4 \cdot 6) \bmod 7 = 3$.

3. С помощью алгоритма Евклида.

Алгоритм Евклида применяется для нахождения НОД чисел a и b. Однако его расширенный вариант можно использовать и для вычисления обратной величины.

Основной вариант.

Даны a иb, $(a > b)$. Алгоритм имеет итерационный характер:

$a = b \cdot q_1 + r_1,\ 0 < r_1 < b;$

$b = r_1 \cdot q_2 + r_2, 0 < r_2 < r_1;$
$r_1 = r_2 \cdot q_3 + r_3, 0 < r_3 < r_2;$
$\vdots$
$r_{k-2} = r_{k-1} \cdot q_k + r_k, 0 < r_k < r_{k-1};$
$r_{k-1} = r_k \cdot q_{k+1}, r_{k+1} = 0;$
$(a, b) = r_k,$

гдеq_i, r_i - частное и остаток на i -м шаге алгоритма. На первом шаге делимое - a, делитель - b, частное - q_1, остаток - r_1. На i -м, $i > 1$шаге алгоритма: делимое - делитель $i-1$-го шага, делитель - остаток $i-1$-го шага (r_{i-1}), частное - q_i, остаток - r_i.

Пример 12.3. Пусть $a = 1071$ и $b = 693$. Найти НОД(a, b).

$$1071 = 693 \cdot 1 + 378, \text{т.е. } q_1 = 1, r_1 = 378;$$
$$693 = 378 \cdot 1 + 315, \text{т.е. } q_2 = 1, r_2 = 315;$$
$$378 = 315 \cdot 1 + 63, \text{т.е. } q_3 = 1, r_3 = 63;$$
$$315 = 63 \cdot 5, \text{т.е. } q_4 = 5, r_4 = 0.$$

То есть на четвертом шаге остаток от деления $r_4 = 0$, следовательно, алгоритм останавливается и НОД$(a, b) = 63$.

Доказано, что при неотрицательных a и b можно найти такие целые числа: u_1, u_2, u_3, что будет выполняться

$$a \cdot u_1 + b \cdot u_2 = u_3 = (a, b).$$

Если выбрать $b = n$ и a, n - взаимно простые числа, т.е. $(a, n) = 1$, тогда

$$a \cdot u_1 + n \cdot u_2 = 1,$$
$$(a \cdot u_1 + n \cdot u_2) \mathrm{mod} n = (a \cdot u_1) \mathrm{mod} n = 1,$$
$$(a^{-1}) \mathrm{mod} n = u_1 \mathrm{mod} n.$$

То есть для нахождения обратной величины необходимо вычислить $u_1 \bmod n$. Эта задача решается в ходе вычисления НОД(a, n) в соответствии с алгоритмом Евклида. Дополнительно на каждом шаге вычисляются координаты двух векторов:
$\vec{u}$=(u_1, u_2, u_3), $\vec{v}(v_1, v_2, v_3)$.

Алгоритм вычисления u_1 имеет следующий вид

1. Начальные установки:

$\vec{u}_0$=(0, 1, n), т.е. $u_1 = 0$, $u_2 = 1$, $u_3 = n$. При этом$a \cdot 0 + b \cdot 1 = n$, т.е. $b = n$,
$\vec{v}_0 = (1, 0, a)$, т.е. $v_1 = 1$, $v_2 = 0$, $v_3 = a$. При этом $a \cdot 1 + n \cdot 0 = a$.

2. Проверяем, выполняется ли $u_3 = 1$, если да, то алгоритм заканчивается.

3. Делим n на a (u_3 на v_3) и определяем:

$q_1 = [\frac{u_3}{v_3}]$и значения векторов: $\vec{u}_1 = \vec{v}_0$; $\vec{v}_1 = \vec{u}_0 - q_1 \cdot \vec{v}_0$.

4. Вернуться к шагу 2.

На каждом шаге при расчетах используются результаты предыдущего:
$q_i = [\frac{u_3}{v_3}]_{i-1}$, $\vec{u}_i = \vec{v}_{i-1}$, $\vec{v}_i = \vec{u}_{i-1} - q_i \cdot \vec{v}_{i-1}$.

При $u_3 = 1$ вычисления заканчиваются $(a^{-1}) \mathrm{mod} n = u_1 \mathrm{mod} n$, где u_1 значение u_1, полученное на последнем шаге.

Пример 12.4. Пусть $n = 23$ и $a = 5$. Найти число x , обратное числу a по модулю n , т.е. найти $5^{-1}\text{mod}23$.

Используя расширенный алгоритм Евклида, выполним вычисления.

q	u_1	u_2	u_3	v_1	v_2	v_3
-	0	1	$n = 23$	1	0	$a = 5$
4	1	0	5	-4	1	3
1	-4	1	3	5	-1	2
1	5	-1	2	-9	2	1
-	-9	2	1			

При $u_1 = -9$, $u_2 = 2$, $u_3 = 1$ выполняется уравнение $a \cdot u_1 + n \cdot u_2 = 1$, $a \cdot (-9) + n \cdot 2 = 5 \cdot (-9) + 23 \cdot 2 = 1$ и $a^{-1}\text{mod}n = 5^{-1}\text{mod}23 = (-9)\text{mod}23 = 14$. Итак, $x = 5^{-1}\text{mod}23 = (-9)\text{mod}23 = 14$.

12.4. Криптосистема *RSA*

Последовательность действий абонентов криптосистемы *RSA*

Действия получателя криптограммы B:

1. B генерирует два произвольных больших простых числа P и Q. Эти числа должны быть примерно одинаковыми, размерностью 100-150 десятичных разрядов. Они должны быть секретными.
2. B вычисляет значение модуля $n = P \cdot Q$ и функции Эйлера $(n) = (P-1) \cdot (Q-1)$ и выбирает значение открытого ключа K_O с соблюдением условий: $1 < K_O \leq \varphi(n)$, $(K_O, \varphi(n))$=1, т.е. K_O и $\varphi(n)$ должны быть взаимно простыми.
3. B вычисляет значение секретного ключа K_C, используя расширенный алгоритм Евклида:

$K_C = (K_O^{-1}) mod \varphi(n)$.

4. B посылает A пару чисел n, K_O по открытому каналу.

Действия отправителя криптограммы A:

1. Разбивает исходный текст M на блоки M_i, $i = 1,2, \dots, m$, т.е. $M = M_1, M_2, \dots, M_m$. Величина $M_i < n$.
2. Шифрует каждое число M_i по формуле $C_i = (M_i^{K_O}) mod\ n$ и отправляет криптограмму $C = C_1, C_2, \dots, C_m$.

Получатель B, получив криптограмму, расшифровывает каждый блок секретным ключом K_C, $M_i = \left(C_i^{K_C}\right) mod\ n$, и восстанавливает весь текст $M = M_1, M_2, \dots, M_m$.

Реализуемость и безопасность *RSA*

Покажем, что при расшифровании восстанавливается исходный текст. Согласно обобщению Эйлером малой теоремы Ферма: если НОД$(a,n)=1$, то $a^{\varphi(n)} \equiv 1 mod n$, или $a^{\varphi(n)+1} \equiv a mod n$. Открытый K_O и закрытый K_C ключи в алгоритме связаны соотношением $K_O \cdot K_C \equiv 1 mod \varphi(n)$, или $K_O \cdot K_C = k \cdot \varphi(n) + 1$ для некоторого целого k. Таким образом, процесс шифрования, а затем расшифрования некоторого сообщения M_i выглядит следующим образом:

$$((M_i^{K_O}) \mathrm{mod} n)^{K_C} \mathrm{mod} n = (M_i^{K_O \cdot K_C}) \mathrm{mod} n = \left(M_i^{k \cdot \varphi(n)+1}\right) \mathrm{mod} n = M_i$$

В процессе применения RSA злоумышленник может иметь: C_i, K_O, n – и организовать дешифрование двумя способами:

1. По C_i, K_O, n получить M_i. Для этого он решает задачу вычисления M_i из уравнения $C_i = M_i^{K_O} mod n$. Эта задача вычислительно трудна.

2. По n вычислить P и Q, затем найти $\varphi(n)$ и вычислить $\cdot K_C = (K_O^{-1}) mod \varphi(n)$ и дешифровать сообщение $M_i = (C_i^{K_C}) mod n$.

Однако задача разложения большого числа на простые множители вычислительно сложна.

Пользователи A и B должны быстро осуществлять все вычисления: вычислять K_O, шифровать и расшифровывать.

Вычисление K_O с использованием алгоритма Евклида - довольно быстрый процесс и не представляет трудности. Шифрование и расшифрование - возведение большого числа в большую степень - требует определенных затрат времени, но, с учетом наличия быстрых алгоритмов и быстродействия современных компьютеров, это приемлемая процедура.

12.5. Криптосистема Эль-Гамаля

Схема Эль-Гамаля, предложенная в 1985 г., может быть использована как для шифрования, так и для цифровых подписей. Безопасность схемы Эль-Гамаля обусловлена сложностью вычисления дискретных логарифмов в конечном поле.

Для того чтобы генерировать пару ключей (открытый ключ – секретный ключ), сначала выбирают некоторое большое простое число P и большое целое число G, причем $G < P$. Числа P и G могут быть распространены среди группы пользователей. Затем выбирают случайное целое число X, причем $X < P$. Число X является секретным ключом и должно храниться в секрете. Далее вычисляют $Y = G^X mod P$. Число Y является открытым ключом.

Для того чтобы зашифровать сообщение M, выбирают случайное целое число $1 < K < P - 1$ такое, что числа K и $(P-1)$ являются взаимно простыми. Затем вычисляют числа $a = G^K mod P$, $b = (Y^K \cdot M) mod p$. Пара чисел (a, b) является шифротекстом. Заметим, что длина шифротекста вдвое больше длины исходного открытого текста M.

Для того чтобы расшифровать шифротекст (a, b), вычисляют $M = (b/a^X) mod P$. Справедливость этого равенства следует из: $a^X \equiv G^{KX} mod P$, $b/a^X \equiv Y^K\ M/a^X \equiv G^{KX}\ M/G^{KX} \equiv M \mathrm{mod} P$.

Лекция 13. Электронная цифровая подпись

13.1. Общие сведения

При обмене сообщениями через ТКС возникает задача подтверждения их подлинности (подтверждения авторства и целостности). Такая же проблема существует и при переходе от юридически значимых бумажных документов к электронным. Сообщения, для которых эта проблема актуальна, будем в дальнейшем называть электронными документами.

Целью аутентификации электронных документов является их защита от возможных видов злоумышленных действий, к которым относятся:

– активный перехват - нарушитель, подключившийся к сети, перехватывает документы (файлы) и изменяет их;

– маскарад - абонент С посылает документ абоненту В от имени абонента А;

– ренегатство - абонент А заявляет, что не посылал сообщения абоненту В, хотя на самом деле послал;

– подмена - абонент В изменяет или формирует новый документ и заявляет, что получил его от абонента А;

– повтор - абонент С повторяет ранее переданный документ, который абонент А посылал абоненту В.

В обычной (бумажной) информатике эти проблемы решаются за счет того, что информация в документе и рукописная подпись автора жестко связаны с физическим носителем (бумагой). В электронных документах на машинных носителях такой связи нет.

Естественно, что для электронных документов традиционные способы установления подлинности по рукописной подписи и оттиску печати на бумажном документе совершенно непригодны, поэтому для подтверждения подлинности документа используется специфическая криптографическая процедура, называемая электронной цифровой подписью (ЭЦП).

ЭЦП функционально аналогична обычной рукописной подписи и обладает ее основными достоинствами:

– удостоверяет, что подписанный текст исходит от лица, поставившего подпись;

– не дает самому этому лицу возможности отказаться от обязательств, связанных с подписанным текстом;

– гарантирует целостность подписанного текста.

ЭЦП представляет собой относительно небольшое количество дополнительной цифровой информации, передаваемой вместе с подписываемым текстом.

Технология ЭЦП включает две процедуры:

1) процедуру постановки подписи;

2) процедуру проверки подписи.

В процедуре постановки подписи используется секретный ключ отправителя сообщения, в процедуре проверки подписи - открытый ключ отправителя.

При формировании ЭЦП отправитель прежде всего вычисляет хэш-функцию $h(M)$ подписываемого документа M Вычисленное значение хэш-функции $h(M)$ представляет собой один короткий блок информации m, характеризующий весь документ M в целом. Затем число m «шифруется» секретным ключом отправителя. Получаемая при этом пара чисел представляет собой ЭЦП для данного документа M. В принципе, можно обойтись без предварительного хэширования документа, а «шифровать» весь документ, однако в этом случае придется иметь дело с гораздо большим по размерам файлом. Употребление слова «шифровать» здесь весьма условное и справедливо при использовании алгоритма RSA, для других алгоритмов точнее говорить «преобразовывать».

При проверке ЭЦП получатель сообщения снова вычисляет хэш-функцию $m = h(M)$ принятого по каналу документа M, после чего при помощи открытого ключа отправителя проверяет, соответствует ли полученная подпись вычисленному значению m хэш-функции.

Принципиальным моментом в системе ЭЦП является невозможность подделки ЭЦП пользователя без знания его секретного ключа подписывания.

В качестве подписываемого документа может быть использован любой файл. Подписанный файл создается из неподписанного путем добавления в него одной или более электронных подписей. Каждая подпись содержит следующую информацию:

- дату подписи;
- срок окончания действия ключа данной подписи;
- информацию о лице, подписавшем файл (Ф.И.О., должность, краткое наименование фирмы);
- идентификатор подписавшего (имя открытого ключа);
- собственно цифровую подпись.

13.2. Однонаправленные хэш-функции

Хэш-функция предназначена для сжатия подписываемого документа M до нескольких десятков или сотен бит. Хэш-функция h принимает в качестве аргумента сообщение (документ) M произвольной длины и возвращает хэш-значение $h(M) = H$ фиксированной длины. Обычно хэшированная информация является сжатым двоичным представлением основного сообщения произвольной длины. Следует отметить, что значение хэш-функции $h(M)$ сложным образом зависит от документа M и не позволяет восстановить сам документ M.

Хэш-функция должна удовлетворять целому ряду условий:

- хэш-функция должна быть чувствительна к всевозможным изменениям в тексте M , таким как вставки, выбросы, перестановки и т.п.;

– хэш-функция должна обладать свойством необратимости, то есть задача подбора документа M', который обладал бы требуемым значением хэш-функции, должна быть вычислительно неразрешима;

– вероятность того, что значения хэш-функции двух различных документов (вне зависимости от их длин) совпадут, должна быть ничтожно мала.

Большинство хэш-функции строится на основе однонаправленной функции f, аргументами, которой являются две величины: блок исходного документа M_i и хэш-значение H_{i-1}, предыдущего блока документа (рис.7.1): $H_i = f(M_i, H_{i-1})$.

Хэш-значение, вычисляемое при вводе последнего блока текста, становится хэш-значением всего сообщения M. В результате однонаправленная хэш-функция

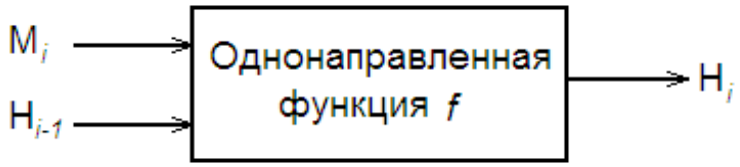


Рис.13.1. Общая схема вычисления однонаправленной хэш-функции

всегда формирует выход фиксированной длины n (независимо от длины входного текста).

Часто функции хэширования строят, используя в качестве однонаправленной функции – симметричный блочный алгоритм шифрования (DES, ГОСТ 28147-89) в режиме с обратной связью, принимая последний блок шифротекста за хэш-значение всего документа. Так как длина блока в указанных алгоритмах невелика (64 бита), то часто в качестве хэш-значения используют два блока шифротекста. Одна из возможных схем хэширования на основе блочного алгоритма шифрования изображена на рис.13.2

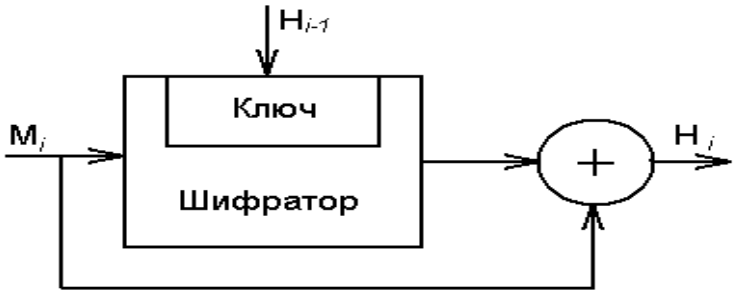


Рис. 13.2. Схема вычисления однонаправленной функции хэширования на базе блочного алгоритма шифрования

13.3. Алгоритм электронной цифровой подписи *RSA*

Первой и наиболее известной во всем мире конкретной системой ЭЦП стала система RSA, математическая схема которой была разработана в 1977 г. в Массачуссетском технологическом институте США.

Сначала необходимо вычислить пару ключей (секретный ключ и открытый ключ). Для этого отправитель (автор) электронных документов вычисляет два больших простых числа P и Q, затем находит их произведение $n = P \cdot Q$ и значение функции Эйлера $\varphi(n) = (P-1)\cdot(Q-1)$. Далее отправитель вычисляет число K_0 из условий: $K_0 \le \varphi(n)$, НОД$\big(K_0, \varphi(n)\big) = 1$ и число K_C из условий: $K_c < n, K_0 \cdot K_c \equiv 1 mod(\varphi(n))$.

Пара чисел (K_0, n) является открытым ключом. Эту пару чисел автор передает партнерам по переписке для проверки его цифровых подписей. Число K_C сохраняется автором как секретный ключ для подписывания. Обобщенная схема формирования и проверки цифровой подписи RSA показана на рис.13.3.

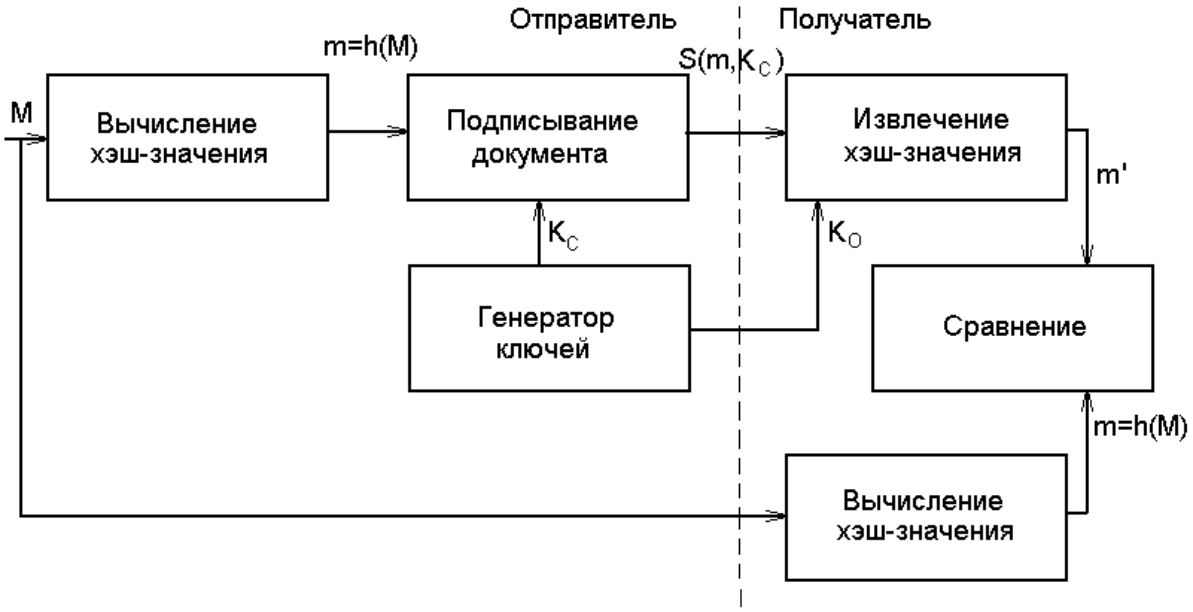


Рис. 13.3. Обобщенная схема алгоритма ЭЦП *RSA*

Допустим, что отправитель хочет подписать сообщение M перед его отправкой. Сначала сообщение M (блок информации, файл, таблица) сжимают с помощью хэш-функции h в целое число $m = h(M)$. Затем вычисляют цифровую подпись S под электронным документом M, используя хэш-значение m и секретный ключ K_C: $S = m^{K_C} mod$n. Пара (M,S) передается партнеру-получателю как электронный документ M, подписанный цифровой подписью S, причем подпись S сформирована обладателем секретного ключа K_C. После приема пары (M,S) получатель вычисляет хэш-значение сообщения M двумя разными способами. Прежде всего, он восстанавливает хэш-значение m', применяя криптографическое преобразование подписи S с использованием открытого ключа K_0: $m' = S^{K_0} mod n$. Кроме того, он находит результат хэширования принятого сообщения M с помощью такой же хэш-функции h: $m = h(M)$. Если

соблюдается равенство вычисленных значений, т.е. $m' = S^{K_0} mod n = h(M)$, то получатель признает пару (*M*,*S*) подлинной. Доказано, что только обладатель секретного ключа K_C может сформировать цифровую подпись *S* по документу *M*, а определить секретное число K_C по открытому числу K_0 не легче, чем разложить модуль *N* на множители. Кроме того, можно строго математически доказать, что результат проверки цифровой подписи *S* будет положительным только в том случае, если при вычислении *S* был использован секретный ключ K_C, соответствующий открытому ключу K_0. Поэтому открытый ключ K_0 иногда называют "идентификатором" подписавшего.

Недостатками алгоритма цифровой подписи RSA являются:

1. При вычислении модуля n, ключей K_C и K_0 для системы цифровой подписи RSA необходимо проверять большое количество дополнительных условий, что сделать практически трудно. Невыполнение любого из этих условий делает возможным фальсификацию цифровой подписи со стороны того, кто обнаружит такое невыполнение при подписании важных документов нельзя допускать такую возможность даже теоретически.

Для обеспечения криптостойкости цифровой подписи RSA по отношению к попыткам фальсификации на уровне, например, национального стандарта США на шифрование информации (алгоритм DES), т.е.10^{18}, необходимо использовать при вычислениях n , K_C и K_0 целые числа не менее 2^{512} (или около 10^{154}) каждое, что требует больших вычислительных затрат, превышающих на 20..30% вычислительные затраты других алгоритмов цифровой подписи при сохранении того же уровня криптостойкости.

2. Цифровая подпись RSA уязвима к так называемой мультипликативной атаке. Иначе говоря, алгоритм цифровой подписи RSA позволяет злоумышленнику без знания секретного ключа K_C сформировать подписи под теми документами, у которых результат хэширования можно вычислить как произведение результатов хэширования уже подписанных документов.

Пример 13.1. Допустим, что злоумышленник может сконструировать три сообщения M_1, M_2 и M_3, у которых хэш-значения $m_1 = h(M_1)$, $m_2 = h(M_2)$, $m_3 = h(M_3)$, причем $m_3 = m_1 \cdot m_2 (mod\ n)$. Допустим также, что для двух сообщений M_1 и M_2 получены законные подписи $S_1 = m_1^{K_c} (mod\ n)$ и $S_2 = m_2^{K_c} (mod\ n)$.Тогда злоумышленник может легко вычислить подпись S_3 для документа M_3, даже не зная секретного ключа $S_3 = S_1 * S_2 (mod\ n)$. Действительно, $S_1 * S_2 (mod\ n) = m_1^{K_c} m_2^{K_c} (mod\ n) = (m_1 * m_2)^{K_c} (mod\ n) = m_3^{K_c} (mod\ n) = S_3$

13.4. Алгоритм цифровой подписи Эль Гамаля ($EGSA$)

Название $EGSA$ происходит от слов $El\ Gamal\ Signature\ Algorithm$ (алгоритм цифровой подписи Эль Гамаля). Идея $EGSA$ основана на том, что для обоснования практической невозможности фальсификации цифровой подписи может быть использована более сложная вычислительная задача, чем

разложение на множители большого целого числа, - задача дискретного логарифмирования. Кроме того, Эль Гамалю удалось избежать явной слабости алгоритма цифровой подписи RSA, связанной с возможностью подделки цифровой подписи под некоторыми сообщениями без определения секретного ключа.

Для генерации пары ключей (открытый ключ - секретный ключ), сначала выбирают некоторое большое простое целое число P и большое целое число G, причем $G < P$. Отправитель и получатель подписанного документа используют при вычислениях одинаковые большие целые числа P (10^{308}или2^{1024}) иG (10^{156} или 2^{512}), которые не являются секретными. Отправитель выбирает случайное целое число K_c, $1 < K_c < p - 1$, и вычисляет $K_0 = G^{K_C} mod P$. Число K_0 является открытым ключом, используемым для проверки подписи отправителя. Число K_0 открыто передается всем потенциальным получателям документов. Число K_C является секретным ключом отправителя для подписывания документов и должно храниться в секрете. Для того чтобы подписать сообщение M, сначала отправитель хэширует его с помощью хэш-функции $h(*)$ в целое число $m = h(M)$, $1 < m < (P - 1)$, и генерирует случайное целое числоK,$1 < K < (P - 1)$, такое, что K и $(P - 1)$ являются взаимно простыми. Затем отправитель вычисляет целое число a: $a = G^K mod P$ и, применяя расширенный алгоритм Евклида, вычисляет с помощью секретного ключа K_C целое число b из уравнения $m = K_c \cdot a + K \cdot b(mod(P - 1))$.

Пара чисел (a, b) образует цифровую подпись S: $S = (a, b)$, проставляемую под документом M. Тройка чисел (M, a, b)передается получателю, в то время как пара чисел (K_C, K) держится в секрете.

После приема подписанного сообщения (M, a, b)получатель должен проверить, соответствует ли подпись $S = (a, b)$ сообщению M. Для этого получатель сначала вычисляет по принятому сообщению M число $m = h(M)$, т.е. хэширует принятое сообщениеM. Затем получатель вычисляет значение $A = K_0^a \cdot a^b (mod\ P)$ и признает сообщение M подлинным, если, и только если $A = G^m mod\ P$. Иначе говоря, получатель проверяет справедливость соотношения $K_0^a \cdot a^b (mod\ P) = G^m (mod\ P)$.

Можно строго математически доказать, что последнее равенство будет выполняться тогда, и только тогда, когда подпись $S = (a, b)$ под документом M получена с помощью именно того секретного ключа K_c , из которого был получен открытый ключ K_0. Таким образом, можно надежно удостовериться, что отправителем сообщения M был обладатель именно данного секретного ключа K_C, не раскрывая при этом сам ключ, и что отправитель подписал именно этот конкретный документ M.

Следует отметить, что выполнение каждой подписи по методу Эль Гамаля требует нового значения K, причем это значение должно выбираться случайным образом. Если нарушитель раскроет когда-либо значениеK, повторно используемое отправителем, то он сможет раскрыть секретный ключ K_C отправителя. Следует отметить, что схема Эль Гамаля является характерным

примером подхода, который допускает пересылку сообщения M в открытой форме вместе с присоединенным аутентификатором (a, b). В таких случаях процедура установления подлинности принятого сообщения состоит в проверке соответствия аутентификатора сообщению.

Схема цифровой подписи Эль Гамаля имеет ряд преимуществ по сравнению со схемой цифровой подписи RSA:

1. При заданном уровне стойкости алгоритма цифровой подписи целые числа, участвующие в вычислениях, имеют запись на 25% короче, что уменьшает сложность вычислений почти в два раза и позволяет заметно сократить объем используемой памяти.

2. При выборе модуля P достаточно проверить, что это число является простым и что у числа *(P-1)* имеется большой простой множитель (т.е. всего два достаточно просто проверяемых условия).

3. Процедура формирования подписи по схеме Эль Гамаля не позволяет вычислять цифровые подписи под новыми сообщениями без знания секретного ключа (как в RSA).

Однако алгоритм цифровой подписи Эль Гамаля имеет и некоторые, недостатки по сравнению со схемой подписи RSA. В частности, длина цифровой подписи получается в 1,5 раза больше, что, в свою очередь, увеличивает время ее вычисления.

Пример. Выберем числа $P = 11$, $G = 2$ и секретный ключ $K_c = 8$. Вычисляем значение открытого ключа $K_0 = G^{K_c} mod P = 2^8 mod 1 = 3$. Предположим, что исходное сообщение M характеризуется хэш-значением $m = 5$. Для того чтобы вычислить цифровую подпись для сообщения M , имеющего хэш-значение $m = 5$, сначала выберем случайное целое число $K = 9$. Убедимся, что числа K и $(P - 1)$ являются взаимно простыми. Действительно, НОД (9,10)=1. Далее вычисляем элементы a и b подписи: $a = Gk\, mod P = 29\, mod 11 = 6$, элемент b определяем из уравнения $m = K_c a + kb(mod(P - 1))$, используя расширенный алгоритм Евклида. При $m = 5, a = 6, K_c = 8, P = 11$ получаем $5 = 6 \cdot 8 + 9 \cdot b(mod 10)$ или $9 \cdot b = -43(mod 10)$. Решение $b = 3$. Цифровая подпись представляет собой пару $a = 6, b = 3$. Далее отправитель передает подписанное сообщение. Приняв подписанное сообщение и открытый ключ $K_0 = 3$, получатель вычисляет хэш-значение для сообщения M: m=5, а затем вычисляет:

1. $K_0^a \cdot a^b (mod P) = 3^6 \cdot 6^3 (mod 11) = 10$,

2. $G^m\, mod P = 2^5 mod 11 = 10$.

Так как эти два целых числа равны, принятое получателем сообщение признается подлинным.

13.5. Белорусские стандарты ЭЦП и функции хэширования

Белорусские стандарты регламентирующие использование электронной цифровой подписи, официальное название которых «Процедура выработки и проверки ЭЦП» и «Функция хэширования», были разработаны группой белорусских специалистов в 1999 г. и официально приняты в 2000 г.

В этих стандартах наряду с элементами классических процедур ЭЦП используются современные идеи, позволяющие увеличить криптостойкость и быстродействие. Так, открытый ключ и секретный ключ связаны известным соотношением $K_0 = (a^{K_c}) mod P$, которое позволяет легко вычислить K_0 по K_c, но очень сложно решение обратной задачи - вычисления K_c по K_0. К подписываемому сообщению добавляется случайная компонента t, что усложняет возможный подбор хэш-значения злоумышленником по известному тексту сообщения.

13.5.1. Обозначения принятые в стандарте СТБ-1176.02-99

- B_p- множество, состоящее из чисел $1,2, \dots, p-1$;
- $c := d$ - присвоение параметру c значения d;
- $c \ mod \ d$ - остаток от деления c на d, где c- натуральное число или ноль, d - натуральное число;
- $c^{-1} \ mod \ d$ - натуральное число b такое, что $b < d$ и $(cb) mod \ d = 1$, где c и d - взаимно простые числа;
- $\lceil c \rceil$ - наименьшее целое число, не меньшее чем c ;
- $\lfloor c \rfloor$ - наибольшее целое число, не большее чем c ;
- $c = \sum_{i=0}^{k-1} c_i (2^b)^i$ - разложение неотрицательного целого числа c по основанию 2^b, где k и b - натуральные числа,
- c_i - целое число, $0 \le c_i \le 2^b$;
- $\oplus$- бинарная операция, определенная на множестве неотрицательных целых чисел по формуле $d \oplus b = \sum_{i=0}^{k-1} ((d_i + b_i) mod 2) 2^i$, где $d = \sum_{i=0}^{k-1} d_i 2^i$, $b = \sum_{i=0}^{k-1} b_i 2^i$, $d_0, \dots, d_{k-1}, b_0, \dots, b_{k-1} \in \{0,1\}$;
- $\circ$ - операция: $B_p \times B_p \to B_p$ определяется для любых $c \in B_p$ и $d \in B_p$ по формуле $c \circ d = (cd(2^{l+2})^{-1}) \mathrm{mod}\ p$;
- $C^{(k)}$- степень числа на основе операции $\circ$, определяется индуктивно по формуле, $c^{(k)} = \begin{cases} \mathrm{c} & , k+1 \\ c^{(k-1)} \circ c, & k > 1 \end{cases}$ где k - натуральное число;
- h - функция хэширования, процедура вычисления значений которой соответствует СТБ.

13.5.2. Процедура выработки ЭЦП

1. Выбираются параметры l и r, которые определяют уровень криптографической стойкости ЭЦП. Число l является длиной записи числа p в системе счисления по основанию $2, r$ является длиной записи числа q в системе счисления по основанию 2.

2. В соответствии с выбранными l и r генерируются простые числа p и q такие, что q делит $p-1$ нацело.

3. Генерируется случайное число d, $0 < d < p$.

4. Вычисляется $a = d^{(\frac{p-1}{q})}$. Если $a \equiv 2^{l+2} mod\ p$, то перейти к пункту 3.

5. Генерируется случайное число x, $0 < x < q$, которое является секретным ключом.

6. Вычисляется число $y = a^{(x)}$, которое является открытым ключом.

7. Генерируется случайное число k, $0 < k < q$.

8. Вычисляется $t = a^{(k)}$. Далее число t разлагается по основанию 2^8, т.е. $t = \sum_{i=0}^{n-1} t_i (2^8)^i$. Таким образом, получаются коэффициенты $t_0, t_1, \dots, t_{n-1}$.

9. Формируется последовательность $M_t = (t_0, t_1, \dots, t_{n-1}, m_1, m_2, \dots, m_z)$, состоящая из коэффициентов $t_0, t_1, \dots, t_{n-1}$ и блоков открытого текста $m_1, m_2, \dots, m_z$.

10. Вычисляется значение хэш-функции $U = h(M_t)$. Если $U = 0$, то перейти к пункту 6.

11. Вычисляется $V = (k - xU) mod q$. Если $V = 0$, то перейти к пункту 6.

12. Вычисляется $S = U \cdot 2^r + V$. ЭЦП последовательности M_t есть число S.

13. Отправляется M_t , S.

13.5.3. Процедура проверки ЭЦП

1. Вычисляется $V = S \mathrm{mod} 2^r$.

2. Вычисляется $U = (S - V)/2^r$.

3. Если хотя бы одно из условий $0 < U < 2^r$ и $0 < V < q$ не выполнено, то ЭЦП считается недействительной и работа алгоритма завершается.

4. Вычисляется $t' = a^{(V)} \circ y^{(U)}$.

5. Число t' разлагается по основанию 2^8, т.е. $t' = \sum_{i=0}^{n-1} t_i' (2^8)^i$. Таким образом, получаются коэффициенты $t_0', t_1', \dots, t_{n-1}'$.

6. Формируется последовательность $M_t' = (t_0', t_1', \dots, t_{n-1}', m_1, m_2, \dots, m_z)$, состоящая из коэффициентов $t_0', t_1', \dots, t_{n-1}'$ и блоков открытого текста $m_1, m_2, \dots, m_z$.

7. Вычисляется хэш-функция $W = h(M_t')$.

8. Проверяется условие $W = U$. При совпадении W и U принимается решение о том, что ЭЦП была создана при помощи личного ключа подписи x, связанного с открытым ключом проверки подписи y, а так же ЭЦП и

последовательность M_t не были изменены с момента их создания. В противном случае подпись считается недействительной.

Стандарт «Процедура выработки и проверки ЭЦП» содержит алгоритмы и процедуры выработки и проверки электронной цифровой подписи, а также подробные инструкции по:

- выбору величин r и l (размер p и q);
- генерации p и q;
- генерации a.

РАЗДЕЛ 2. ПРАКТИЧЕСКИЙ
Лабораторная работа №1. Стандарт шифрования данных ГОСТ 28147-89.

ЦЕЛЬ РАБОТЫ: Закрепление теоретических знаний по стандарту шифрования данных ГОСТ 28147-89.

ПРЕДВАРИТЕЛЬНОЕ ЗАДАНИЕ

1.Изучите теоретическую часть.

2.Найдите сумму по модулю 2 двух чисел $\mathbf{2940553835_{10}}$ и $\mathbf{3984555948_{10}}$.

3.Найдите сумму по модулю 3 двух чисел $\mathbf{3696_{10}}$ и $\mathbf{3718_{10}}$.

4.Найдите сумму по модулю 2^{32}следующих пар чисел: $\mathbf{3037741847_{10}}$ и $\mathbf{1257225448_{10}}$, $\mathbf{2706981523_{10}}$ и $\mathbf{1587985773_{10}}$.

5.Пусть каждая из 16 первых букв русского алфавита (абвгдежзийклмноп) имеет четырехразрядный двоичный код, соответствующий ее номеру от 0 до 15, т.е. а - 0000, б – 0001 , …, п -1111 . Составьте из этих букв произвольное сообщение из 32 символов, затем разбейте полученное сообщение на блоки длиной, соответствующей разрядности накопителей $\mathbf{N_1}$ и $\mathbf{N_2}$ (Рисунок 1.). Значения полученных блоков запишите в десятичной системе счисления.

6.Найдите состояние 32-разрядного двоичного регистра сдвига после циклического сдвига вправо на 3 разряда $\mathbf{298865410_{10}}$, предварительно записанного в регистр.

ЛАБОРАТОРНОЕ ЗАДАНИЕ

1.Включите ПЭВМ.

2.Запустите программу **gost.exe** на выполнение. Данная программа реализует алгоритм зашифрования и расшифрования данных по стандарту ГОСТ 28147-89 в режиме простой замены. Входные и выходные данные работы этого алгоритма представляют собой 64-разрядные двоичные числа, поэтому, если данные, подлежащие обработке, представлены в другом виде, их предварительно переводят в двоичный вид и разбивают на блоки длиной по 64 бита. Для упрощения работы с 64-разрядными двоичными числами блоки представляются в виде двух блоков по 32 бита (соответствующих разрядности накопителей $\mathbf{N_1}$ и $\mathbf{N_2}$) и записываются в десятичной системе счисления.

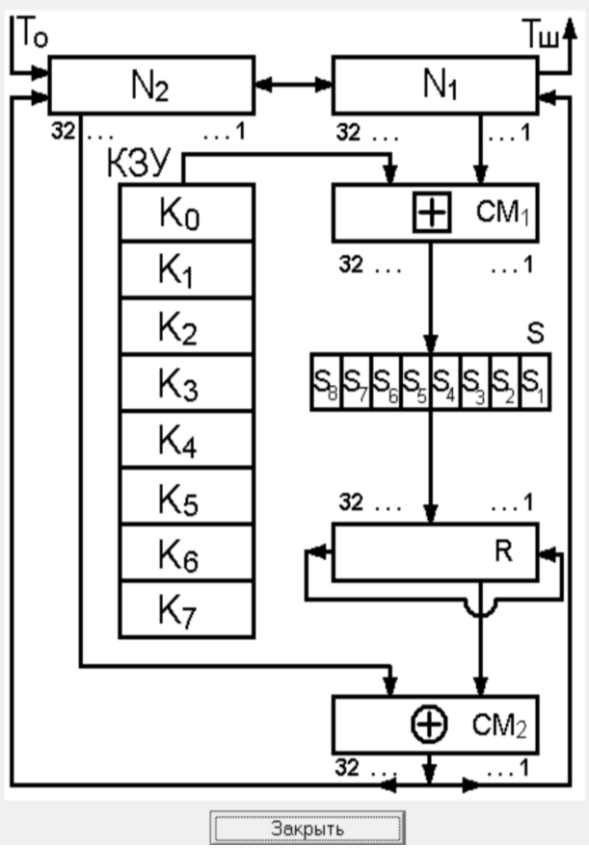


Рисунок 1.Структурная схема режима простой замены

3.На заставке, представленной на Рисунке 2. нажмите кнопку «Начать выполнение работы».

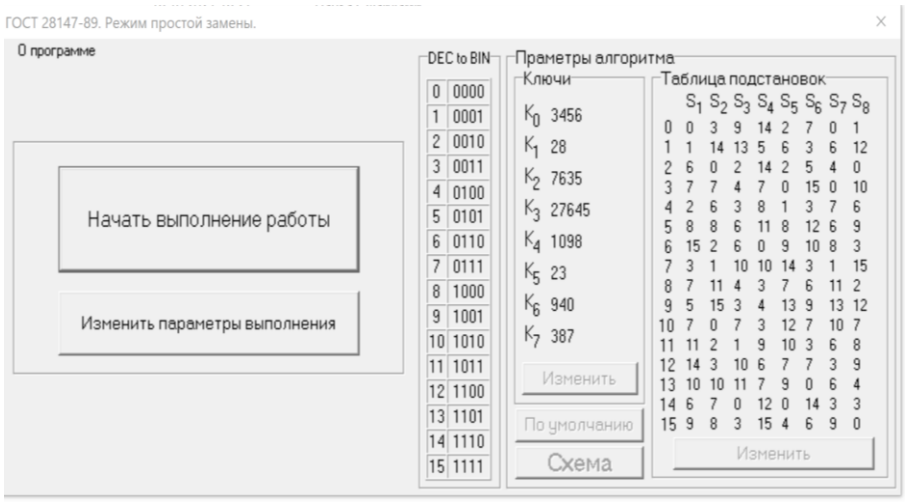


Рисунок 2.Домашняя страница программы

Выполнение работы состоит в том, что Вы должны вручную найти значения в контрольных точках работы алгоритма. Для этого необходимо ввести полученное значение в соответствующее поле рабочего окна программы и нажать кнопку «Проверить».

Если расчет выполнен правильно, происходит переход к следующей контрольной точке.

4.Зашифровывание в режиме простой замены.

4.1.Введите (в десятичной системе счисления) начальное состояние накопителей $\mathbf{N_1}$ и $\mathbf{N_2}$.

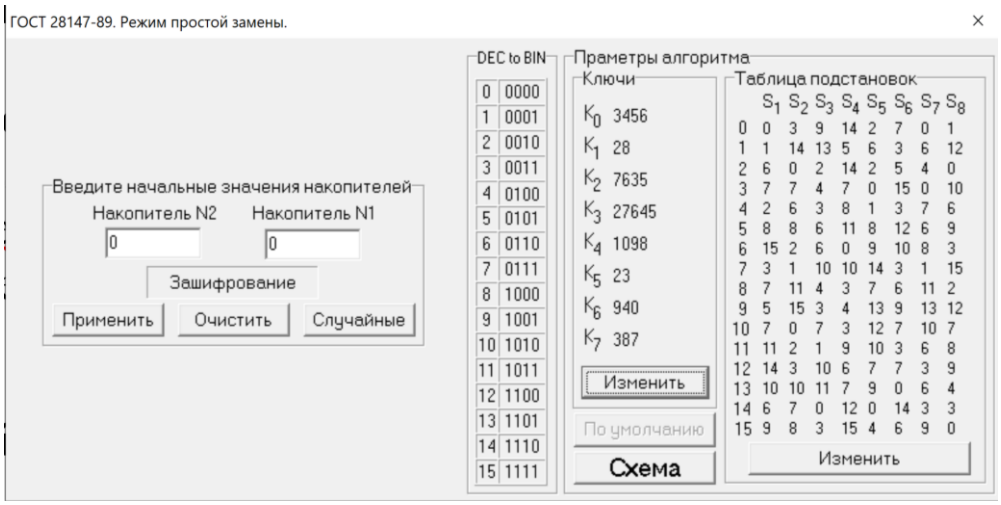


Рисунок 3.Шаблон для ввода исходных данных

4.2.Руководствуйтесь инструкциями в рабочем окне программы.

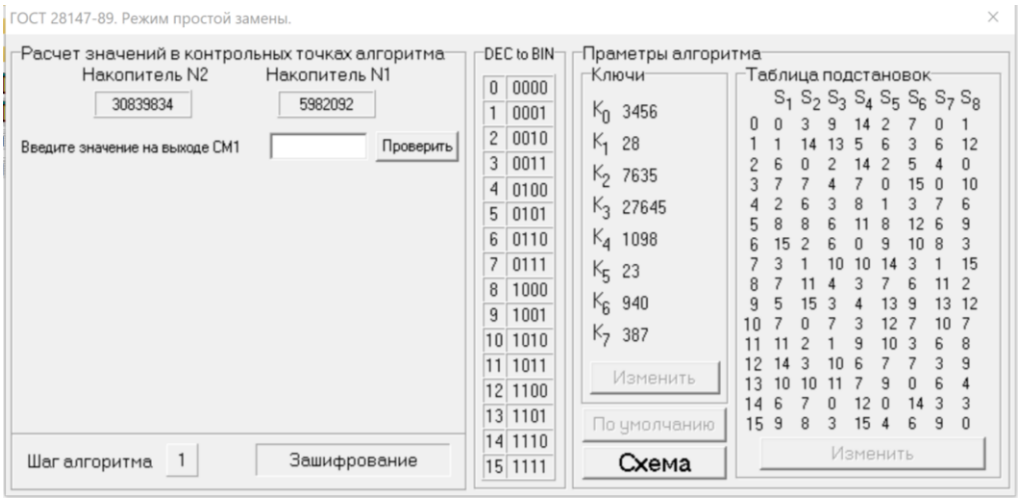


Рисунок 4. Вид окна после ввода исходных данных

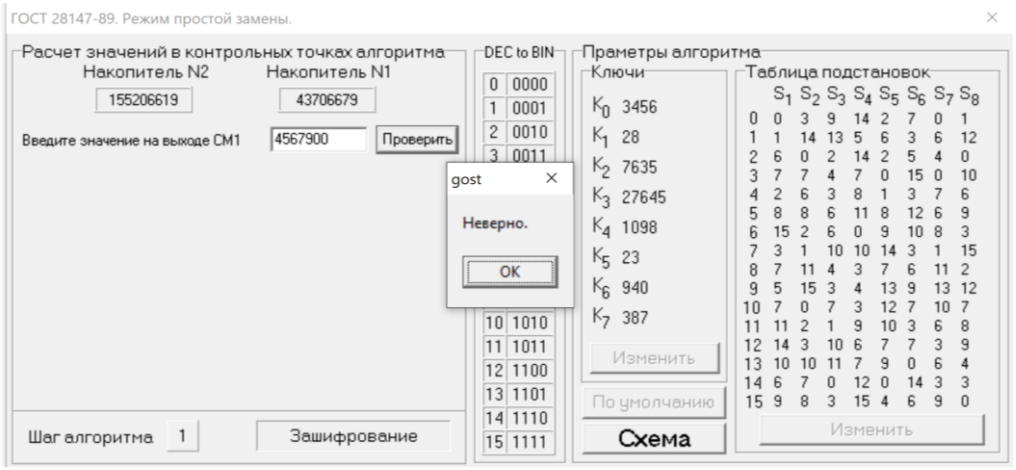


Рисунок 5. Вид окна в случае ошибки при вводе результата

4.3.Для того чтобы скопировать в буфер обмена результат какой-либо контрольной точки, достаточно один раз щелкнуть по нему. Подтверждением того, что копирование произошло, является кратковременное изменение цветового фона.

5.Расшифрование в режиме простой замены.

5.1.Введите (в десятичной системе счисления) начальное состояние накопителей $\mathbf{N_1}$ и $\mathbf{N_2}$.

5.2.Руководствуйтесь инструкциями в рабочем окне программы.

6.Оформите отчет и сделайте выводы.

5. СОДЕРЖАНИЕ ОТЧЕТА

1.Решение задач предварительного задания.

2.Результаты выполнения работы.

3.Анализ результатов и выводы.

Лабораторная работа №2. Стандарт шифрования данных DES.

ЦЕЛЬ РАБОТЫ:
Закрепление теоретических знаний по алгоритму DES шифрования информации в криптосистемах симметричного типа.

ПРЕДВАРИТЕЛЬНОЕ ЗАДАНИЕ

1. Изучите теоретическую часть.
2. Переведите число 3^{43} двоичную систему счисления.
3. Пусть каждая из 16 первых букв русского алфавита (абвгдежзийклмноп) имеет четырехразрядный двоичный код, соответствующий ее номеру от 0 до 15, т.е. а - 0000, б – 0001 , …, п -1111 . Составьте из этих букв произвольное сообщение из 32 символов, затем разбейте полученное сообщение на блоки длиной 64 бита. Значения полученных блоков запишите в десятичной системе счисления.
4. Найдите состояние 28-разрядного двоичного регистра сдвига после циклического сдвига влево на 5, числа $\mathbf{179317333_{10}}$, предварительно записанного в регистр.
5. Найдите сумму по модулю 2 двух чисел $\mathbf{2244899301_{10}}$ и $\mathbf{28973675_{10}}$.

ЛАБОРАТОРНОЕ ЗАДАНИЕ

1. Включите ПЭВМ.
2. Запустите программу des.exe на выполнение. Данная программа реализует алгоритм зашифровывания и расшифровывания данных по стандарту DES, DES в режиме обратной связи по шифротексту, комбинированный алгоритм DES. Входные и выходные данные работы этого алгоритма представляют собой 64-разрядные двоичные числа. Поэтому если данные, подлежащие обработке, представлены в другом виде, их предварительно переводят в двоичный вид и разбивают на блоки длиной по 64 бита. Для упрощения работы с 64-разрядными двоичными числами блоки представляются в десятичной системе счисления.

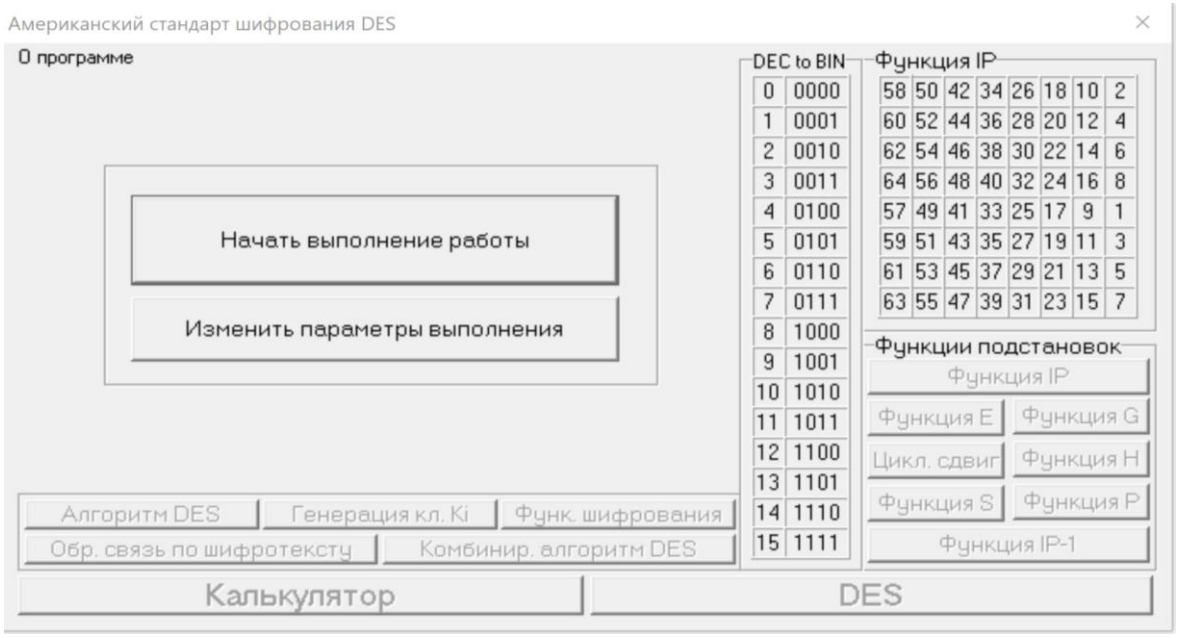


Рисунок 1. Домашняя страница программы

3. Для выбора исследуемого алгоритма необходимо нажать кнопку «Изменить параметры выполнения». При этом предлагается отметить перечень, выполняемых пунктов лабораторной работы (Рисунок 2.). После нажатия на кнопку «Применить» возвращаемся на домашнюю страницу.

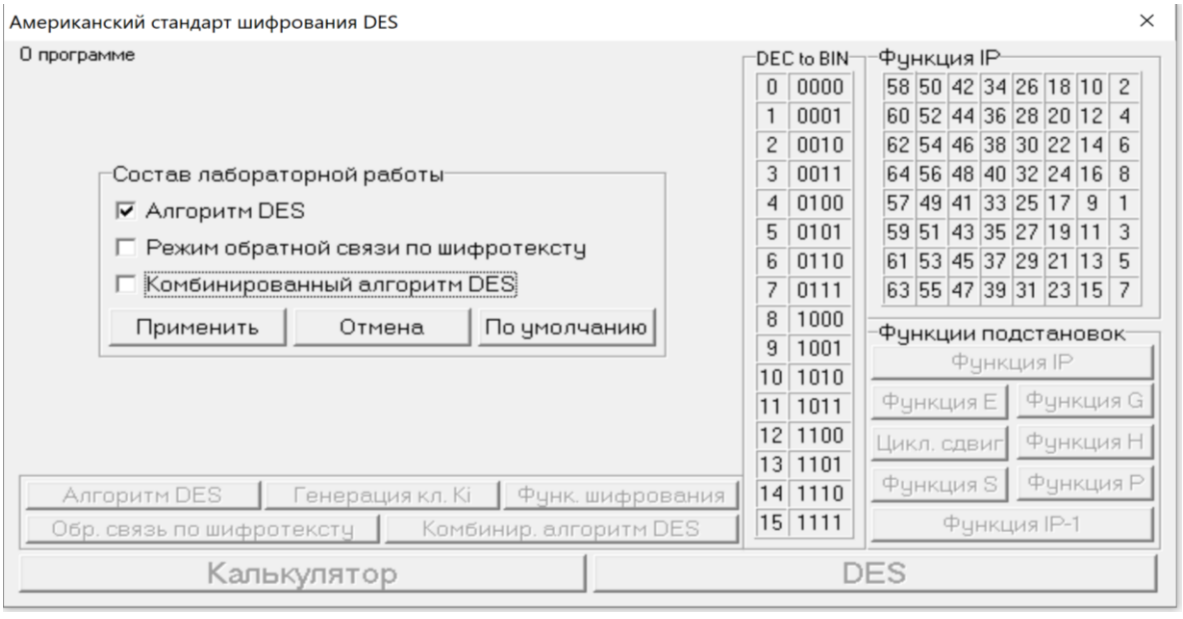


Рисунок 2. Выбор состава лабораторной работы

4. Нажмите кнопку «Начать выполнение работы». Выполнение работы состоит в том, что Вы должны вручную найти значения в контрольных точках работы алгоритма. Для этого необходимо ввести полученное значение в соответствующее поле рабочего окна программы и нажать кнопку «Проверить». Если расчет выполнен правильно, происходит переход к следующей контрольной точке.

5. Программа имеет встроенный калькулятор, который выполняет перевод 64-разрядных чисел из двоичной системы счисления в десятичную и наоборот, а также осуществляет циклические сдвиги. (Для перевод 64-разрядных чисел из двоичной системы счисления в десятичную и наоборот можно использовать встроенный калькулятор ОС Windows).

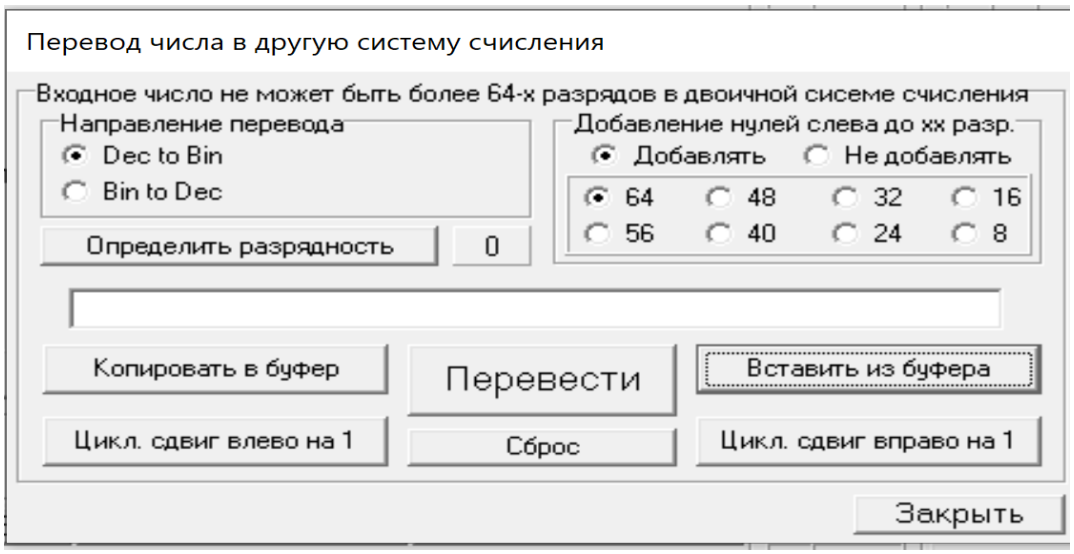


Рисунок 3. Калькулятор для перевода чисел в другие системы счисления

Исследование работы криптографического алгоритма DES.

1.Введите (в десятичной системе счисления) значение 64-разрядного ключа (Рисунок 4).

2.Введите (в десятичной системе счисления) значение 64-разрядного блока открытого текста (Рисунок 5).

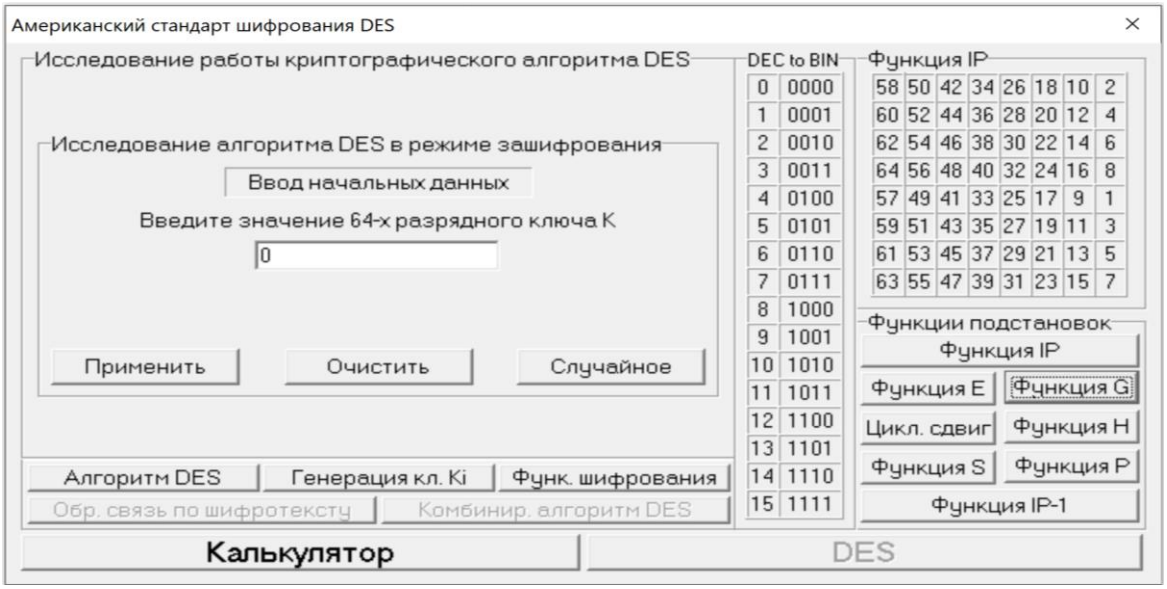


Рисунок 4. Ввод значения ключа

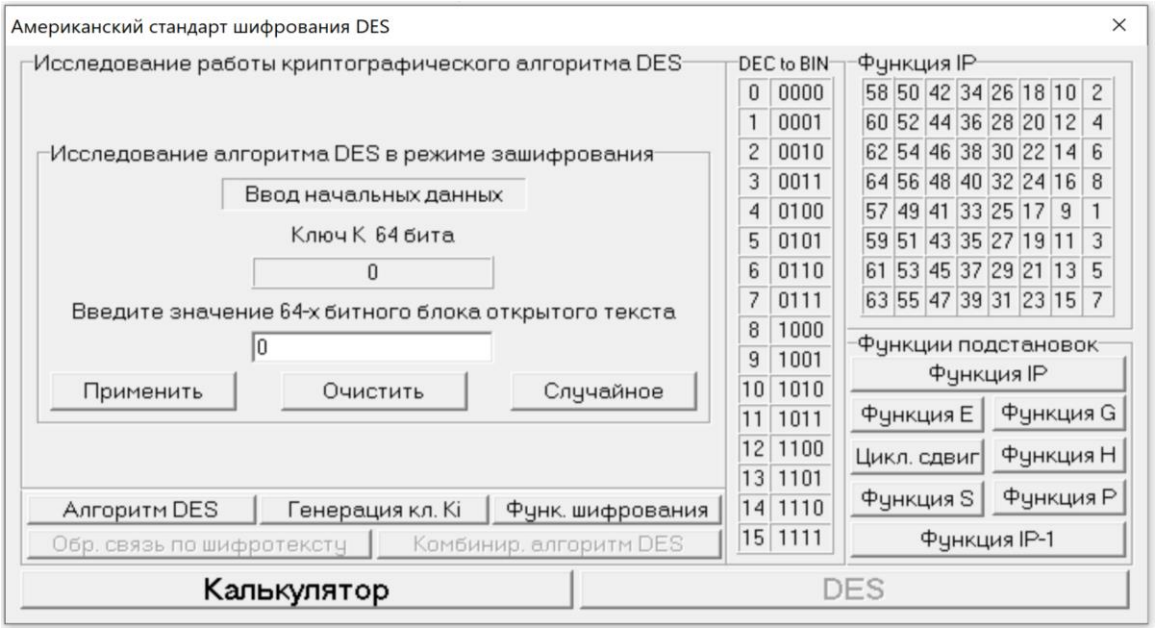


Рисунок 5. Ввод блока открытого ключа

3. Руководствуйтесь инструкциями в рабочем окне программы.

4. Для того чтобы скопировать в буфер обмена результат какой-либо контрольной точки, достаточно один раз кликнуть по нему. Подтверждением того, что копирование произошло, является кратковременное изменение цветового фона.

Исследование DES в режиме обратной связи по шифротексту.

1. Введите (в десятичной системе счисления) значение 64-разрядного вектора инициализации **IV**.

2. Введите (в десятичной системе счисления) значения двух 10-битных блоков открытого текста.

3. Для того чтобы зашифровать или расшифровать какой-либо 64-разрядный блок открытого текста, используя заданный ключ **K**, необходимо нажать кнопку «DES» в рабочем окне программы. В открывшемся окне необходимо ввести начальные данные (открытый текст или шифротекст, ключ **K**), выбрать направление работы (зашифровывание или расшифровывание) и нажать кнопку «Выполнить».

Исследование комбинированного алгоритма DES.

1. Введите (в десятичной системе счисления) значение 64-разрядного ключа $\mathbf{K_1}$.

2. Введите (в десятичной системе счисления) значение 64-разрядного ключа $\mathbf{K_2}$.

3. Введите (в десятичной системе счисления) значение 64-разрядного блока открытого текста.

4. Руководствуйтесь инструкциями в рабочем окне программы.

5. Оформите отчет и сделайте выводы.

СОДЕРЖАНИЕ ОТЧЕТА

1. Решение задач предварительного задания.
2. Результаты выполнения работы.
3. Анализ результатов и выводы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1.Какие существуют режимы работы алгоритма?
2.К какому типу криптосистем относится алгоритм?
3.Какой разрядности ключ используется в алгоритме?
4.Поясните принцип работы блока замены.
5.Поясните принцип работы функции шифрования.
6.Перечислите основные достоинства и недостатки алгоритма.
7.Как повлияет искажение одного бита шифротекста на передаваемую информацию при разных режимах работы алгоритма?
8.Чем определяется криптостойкость алгоритма?

ПРИЛОЖЕНИЕ

Таблица 1

Начальная перестановка **IP**							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Таблица 2

Обратная перестановка $\mathbf{IP^{-1}}$							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Таблица 3

Функция **E**					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Таблица 4

Функция **P**			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Таблица 5

Функция **G**							
	57	49	41	33	25	17	9
	1	58	50	42	34	26	18
	10	2	59	51	43	35	27
	19	11	3	60	52	44	36
	63	55	47	39	31	23	15
	7	62	54	46	38	30	22
	14	6	61	53	45	37	29
	21	13	5	28	20	12	4

Таблица 6

Функция **H**					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	22	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Таблица 7

Таблица сдвигов для вычисления ключа																
Итерация	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Сдвиг влево	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Таблица 8

Функции **S**																
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	4	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	1	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Лабораторная работа № 3. Стандарт шифрования данных RSA

ЦЕЛЬ РАБОТЫ:

Закрепление теоретических знаний по алгоритму RSA шифрования информации в криптосистемах асимметричного типа.

ПРЕДВАРИТЕЛЬНОЕ ЗАДАНИЕ

1.Изучите теоретическую часть.

2.Определите число **b=a mod n**, где **a=751**, **n=39**, и **a= -819**, **n=52**.

3.Определите наибольший общий делитель чисел **a=26569969** и **b=14408761** с помощью алгоритма Евклида.

4.Определите, являются ли числа **a=1172827** и **b=1089019** взаимно простыми.

5.Определите методом перебора число **x** , обратное числу **a** (т.е. **(x*a) mod n=1**) по модулю **n**, где **a=21**, **n=33** и **a=8**, **n=35**.

6.Определите с помощью расширенного алгоритма Евклида число **x**, обратное числу **a=3** (т.е. **(x*a) mod n=1**) по модулю **n=667**.

7.Определите **φ(189)**, где **φ(n)**, - функция Эйлера.

8.Определите число $b=a^c \bmod n$, где **a=83089**, **c=88711**, **n=509**.

ЛАБОРАТОРНОЕ ЗАДАНИЕ

1.Включите ПЭВМ.

2.Запустите программу **rsa.exe** на выполнение. Данная программа реализует алгоритм зашифровывания и расшифровывания данных по алгоритму RSA. Входные и выходные данные работы этого алгоритма представляют собой числа, не превосходящие значение модуля.

3.Нажмите кнопку «Начать выполнение работы». Выполнение работы состоит в том, что Вы должны вручную найти значения в контрольных точках работы алгоритма. Для этого необходимо ввести полученное значение в соответствующее поле рабочего окна программы и нажать кнопку «Проверить». Если расчет выполнен правильно, происходит переход к следующей контрольной точке.

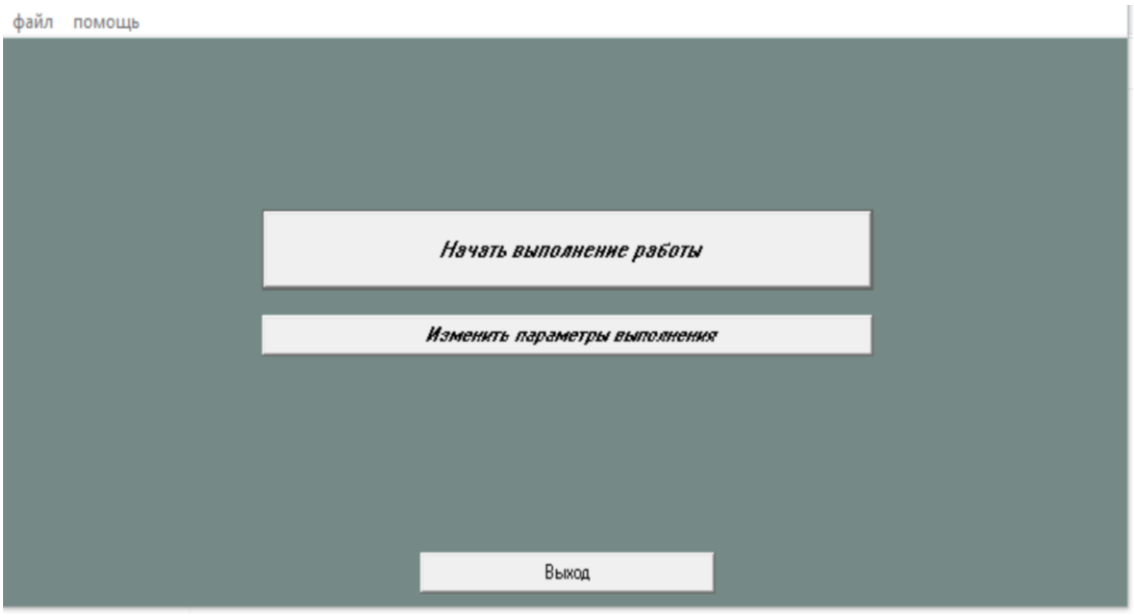


Рисунок 1. Домашняя страница программы

Рисунок 2. Выбор для исследования алгоритма Эль-Гамаля

4.Руководствуйтесь инструкциями в рабочем окне программы.

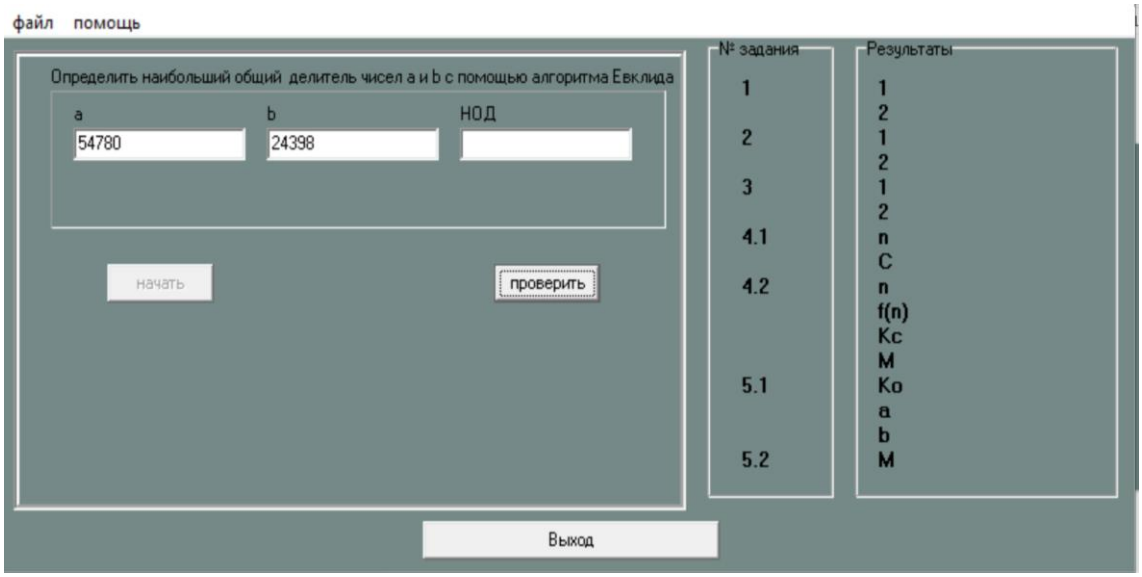


Рисунок 3. Определение наибольшего общего делителя

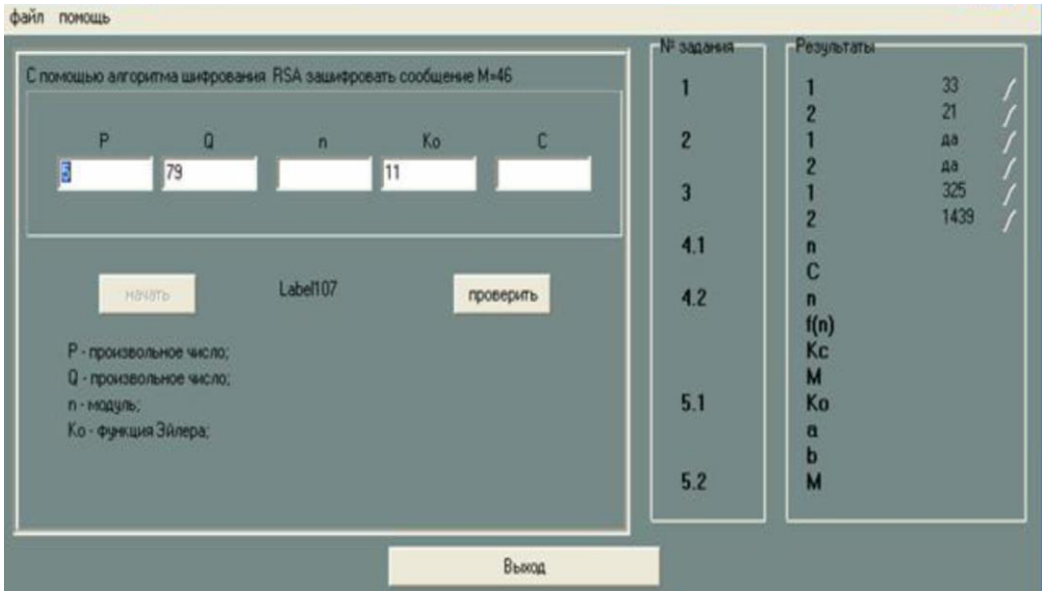


Рисунок 4. Ввод зашифрованного сообщения

СОДЕРЖАНИЕ ОТЧЕТА

1.Решение задач предварительного задания.

2.Результаты выполнения работы.

3.Анализ результатов и выводы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1.Какие числа называются простыми, взаимно простыми?

2.Какое число называется обратным к числу **a** по модулю **n**?

3.Поясните сущность функции Эйлера?

4.Что такое вычет числа **a** по модулю **n**?

5.Какие основные свойства арифметических действий в модулярной арифметике.

6.Поясните основные способы нахождения обратных величин в модулярной арифметике.

7.Поясните работу алгоритма Евклида?

8.К какому типу криптосистем относится алгоритм RSA?

9.Приведите достоинства криптосистем с открытым ключом.

10.Раскройте связь открытого $\mathbf{K_o}$ и секретного $\mathbf{K_c}$ ключей алгоритма RSA?

11.Перечислите действия отправителя криптограммы.

12.Перечислите действия получателя криптограммы.

13.Каким образом злоумышленник может расшифровать криптограмму?

РАЗДЕЛ 3. КОНТРОЛЬ ЗНАНИЙ

3.1 Общая формулировка заданий к контрольной работе

Методические указания по выполнению контрольной работы по дисциплине «Основы информационной безопасности» студент должен выполнить контрольную работу, которая предоставляется на кафедру и защищается студентом заочной (дистанционной) формы получения образования до начала лабораторно зачётной сессии. При выполнении контрольных работ необходимо соблюдать следующие правила:

1. Контрольная работа может выполняться как в рукописном, так и печатном виде, на титульном листе необходимо указать наименование дисциплины, фамилию и инициалы студента, выполненный вариант (соответствует последним двум цифрам зачетной книжки), шифр специальности и номер группы.

2. Контрольную работу следует выполнять аккуратно, оставляя поля для замечаний рецензента.

3. Для пояснения выполнения работы, где это необходимо, сделать скриншот.

4. Выполнение работы сопровождается пояснениями в виде текстовой информации от разработчика.

5. В пояснениях к задаче необходимо указывать используемые подходы, а также пояснение к предложенному методу выполнения задания.

6. Особое внимание уделяется написанию вывода по работе, который должен подчеркивать проделанную работу обучаемым и приобретенные им знания.

7. В контрольной работе следует указывать учебники и учебные пособия, которые использовались при решении поставленных задач.

3.2. Задание на контрольную работу по курсу «Основы информационной безопасности»

1. **Изложить теорию по одной из тем в соответствии с номером варианта**

Номер варианта	Тема
1,16	Межсетевые экраны
2,17	VPN-сети
3,18	Системы обнаружения удаленных атак
4,19	Модели управления доступом к информации в КС
5,20	Методы и средства защиты программного обеспечения

6,21	Аутентификация пользователей в КС
7,22	Алгоритм распределения ключей Диффи-Хеллмана
8,23	Методы и средства защиты от утечки по техническим каналам.
9,24	Методы и средства технической разведки.
10,25	Утечка информации по техническим каналам
11,26	Современные алгоритмы симметричного шифрования
12,27	Алгоритмы асимметричного шифрования
13,28	Электронная цифровая подпись
14,29	Основные схемы включения межсетевых экранов.
15,30	Управление криптографическими ключами.

2. Алгоритм шифрования ГОСТ 28147-89.

В режиме простой замены произвести первый цикл шифрования блока открытого текста T_0 (согласно№ варианта) и вычислить числа: a (число в накопителе N_1) и b (число в накопителе N_2).

Исходные данные:

для вариантов (1-10) ключ K_0= 10111010011000110011001100110011

для вариантов (11-20) ключ K_0= 10010101010101010111010100111001

для вариантов (21-30) ключ K_0= 10000111010101010010101000011010

Номер варианта	Блок открытого текста T_0
1,16	10011010100001000010010111011110011111100110100011001010010101
2,17	00001101111000001100000011000001100010111101101110110100011111111
3,18	01100010111101101110110100011111110011010100001000010010111011111
4,19	10000011110110101001100011000011011001011111001111101010110000001
5,20	10110101010100011101001100011000011011001011111001111101010110000
6,21	00100111101101010011000110000110110010111110011111010101100011100
7,22	01111011010100110001100001101100101010011111010101100011000011101 0
8,23	1000 01111011010100110001100001101100100101111100111110111000001 1
9,24	11100001101111000001100000011000001100010111100001110101010100 00
10,25	00001100010111110110111011010001111111001010101000001101010001101
11,26	00011010101000011011010101010001110100110001100001101100101111 10
12,27	10000011000101111011011101101000111111101010000111000100101000 10
13,28	11110000000000001001100011000011011001010011111010101100011111 00
14,29	01001100011000011011001011111001111101010110000001101010101010 10
15,30	11011000110000110110010111110011111010101100000000111010101010 10

Таблицы замен (S-блок)

Номер таблицы

№стр	**s1**	**s2**	**s3**	**s4**	**s5**	**s6**	**s7**	**s8**
0	0	3	9	14	2	7	0	1
1	1	14	13	5	6	3	6	2
2	6	0	2	14	2	5	4	0
3	7	7	4	7	0	15	0	10
4	2	6	3	8	1	3	7	6
5	8	8	6	11	8	12	6	9
6	15	2	6	0	9	10	8	3
7	3	1	10	10	14	3	1	5
8	7	11	4	3	7	6	11	2
9	5	15	3	4	13	9	13	12
10	7	0	7	3	12	7	10	7
11	11	2	1	9	10	3	6	8
12	14	3	10	6	7	7	3	9
13	10	10	11	7	9	0	6	4
14	6	7	0	12	0	14	3	3
15	9	8	3	15	4	6	9	0

3. Асимметричая криптосистема RSA

Сгенерировать ключи для шифрования и расшифрования: открытый $К_0$ и секретный $К_с$, зашифровать сообщение М и расшифровать его. Убедиться , что ключи сгенерированы правильно.

Исходные данные: простые числа P и Q, сообщение M.

Номер варианта	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
P	7	7	7	7	7	7	7	11	11	11	11	11	11	13	13	13
Q	31	11	13	17	19	23	29	13	17	19	23	29	31	17	19	23
M	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Номер варианта	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
P	13	13	17	17	17	17	19	19	19	23	23	29	31	37	37	37
Q	29	31	19	23	29	19	23	29	31	31	37	43	43	43	11	13
M	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

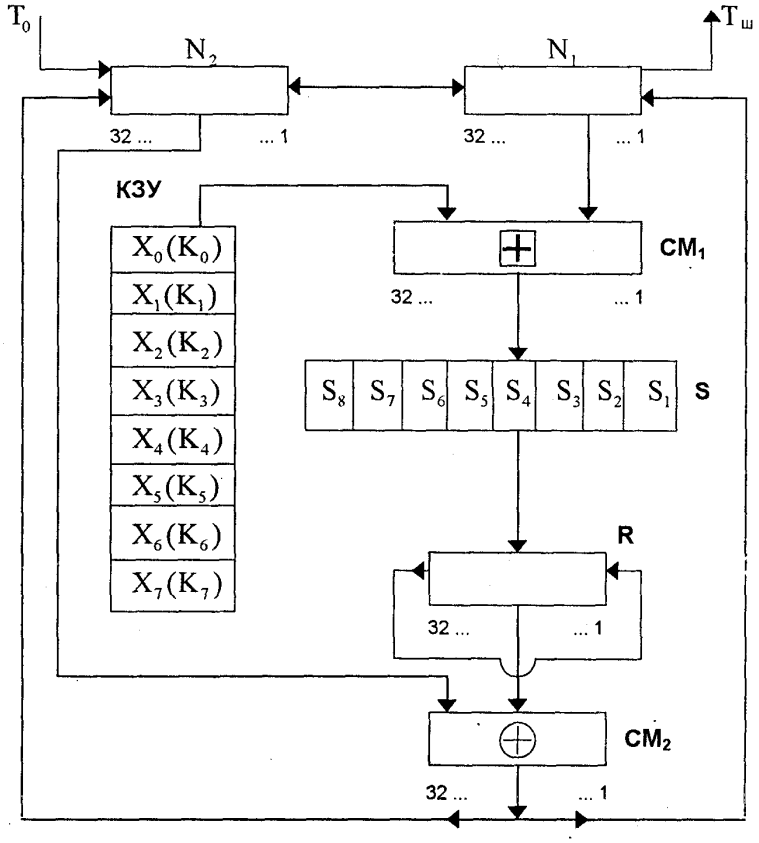
3.3. Методические указания по выполнению контрольной работы

1. **Изложить теорию по одной из тем в соответствии с номером варианта**

Изучить вопрос в соответствии с вариантом. Изложит кратко и четко, по существу вопроса. Весь излагаемый материал должен быть понятен автору контрольной работы.

2. **ГОСТ 28147-89. Режим простой замены**

Схема алгоритма имеет вид:



Произвести первый цикл шифрования и вычислить числа: a (число в накопителе N_1) и b (число в накопителе N_2).

ПРИМЕР:

Число T_0 размером 64 бита делим пополам:

$a_{(2)}$=00011101 00010000 00001100 00011100- правая половина;

$b_{(2)}$=00110001 00100011 01000101 01100010- левая половина;

С помощью калькулятора (режим **Программист**) переводим части открытого текста и ключ K_0 в десятичные числа

$a_{(10)}$=487590940

$b_{(10)}$=824395106

$(K_0)_{10}$= 185762065

Число **a** помещаем в накопитель N_1, Число b помещаем в накопитель N_2.
В сумматоре СМ1 суммируем и приводим по модулю 2^{32}:

$(a_{(10)}+ (K_0)_{10})\text{mod } 2^{32}$=(487590940+185762 065)mod 4 294 967 296=673353005.

Если сумма $a_{(10)}+ (K_0)_{10}$оказывается больше чем 4 294 967 296, то это означает, что сумма- 33-разрядное число, калькулятор от такого числа стандартную функцию mod вычислить не может, он работает только с 32-разрядными числами. Считаем вручную:

$(a_{(10)}+ (K_0)_{10})\text{mod } 2^{32}=(a_{(10)}+ (K_0)_{10})-1*2^{32}$= …….)

На входе блока замен (S) имеем:
673353005_{10}=0010 1000 0010 0010 1000 1101 0010 1101

Каждые 4 входных бита меняем с помощью таблиц S:
0010=2 входим в 2 строку таблицы S_8 , считываем 0=0000
1000=8 входим в 8 строку таблицы S_7 , считываем 11=1011
0010=2 входим в 2 строку таблицы S_6 , считываем 5=0101
0010 =2 входим в 2 строку таблицы S_5 , считываем 2=0010
1000 =8 входим в 8 строку таблицы S_4 , считываем 3=0011
1101 =13 входим в 13 строку таблицы S_3 , считываем 11=1011
0010 =2 входим в 2 строку таблицы S_2 , считываем 0=0000
1101=13 входим в 13 строку таблицы S_1 . считываем 10=1010
Формируем число на выходе блока S:

S= 0000 1011 0101 0010 0011 1011 0000 1010=189938442

Это число поступает в регистр **R**, в котором осуществляют циклический сдвиг влево на 11 бит влево:
0000 1011 0101 0010 0011 1011 0000 1010– число до сдвига
R=1 0010 0011 1011 0000 1010 0000 1011 010 =2446872666– число после сдвига.

Число после сдвига суммируется в СМ2 с b поразрядно по модулю 2.
(Таблица истинности сумматора по модулю 2: 0+0=0,0+1=1, 1+0=1, 1+1=0):

(R+b)mod2=10010001110110000101000001011010
00110001001000110100010101100010
10100000111110110001010100111000= 2700809528
Число с выхода сумматора записываем в накопитель N_1
a=2700809528
а число бывшее в N_1 переписываем в N_2
b=487590940.

На этом первый цикл шифрования закончен.

3. Асимметричая криптосистема RSA

Пример:

Сгенерировать ключи, открытый $К_0$ и секретный $К_с$, ключи, для шифрования и расшифрования, зашифровать сообщение М и расшифровать его. Убедиться, что ключи сгенерированы правильно.

Исходные данные: простые числа P=17 и Q=13, сообщение M=3.

3.1. Вычисляем n=P*Q=13*17=221

3.2. Вычисляем функцию Эйлера F(n)=(p-1)*(Q-1)=16*12=192

3.3. Выбираем значение открытого ключа $К_0$ с соблюдением условий

$1<К_0<F(n)$, $К_0$ и F(n) – взаимно простые числа (их НОД=1)

$К_0=11$

3.4. Из уравнения $(К_0 * К_с)\ mod F(n)=1$

$(11 * К_с) mod 192=1$

Находим $К_с=35$

Уравнение решается методом расширенного алгоритма Евклида.

3.5. Шифрование $C=M^{K0} mod\ n=3^{11} mod 221= 177147 mod 221=126$

3.6. Расшифрование $M=C^{Kc} mod 221=126^{35} mod 221=3$

РАЗДЕЛ 4. ВСПОМОГАТЕЛЬНЫЙ

4.1. Программа дисциплины

Учебная программа по учебной дисциплине «Основы информационной безопасности» разработана для специальности 1-40 01 01 «Программное обеспечение информационных технологий» специализации 1-40 01 01 01 «Веб-технологии и программное обеспечение мобильных систем».

Интенсивное внедрение информационных технологий во все области деятельности человека позволяет обеспечить оперативный обмен сведениями между службами, отделами предприятия и организациями в целом за счет оптимизации информационных потоков, что позволяет ускорить и сделать более качественным процесс их взаимодействия. Сведения, которыми обмениваются такие партнеры, как правило, носят конфиденциальный характер и относятся к категориям служебной или государственной тайны, что требует подготовки современных специалистов, обладающих не только специальными знаниями по их профилю обучения, но и владением основами защиты информации.

Целью изучения учебной дисциплины является получение студентами базовых знаний по вопросам организации защиты информации в информационных системах содержащих персональные данные, государственную или служебную тайны (в том числе сведения об объектах интеллектуальной собственности) в условиях различных по виду, происхождению и характеру возникновения угроз. Основное внимание, в рамках дисциплины уделяется рассмотрению методов защиты от несанкционированного доступа к информации в компьютерных сетях с помощью программных и аппаратурно-программных средств, базирующихся на криптографических преобразованиях; угроз в коммуникационных сетях и мер по предотвращению попыток реализации этих угроз, а также методов симметричного шифрования, криптографии с открытыми ключами, хэш-функций, основ построения и использования межсетевых экранов, практических вопросов сетевой безопасности, методов обеспечения надежного хранения информации в компьютерных сетях.

Основными задачами преподавания учебной дисциплины являются: подготовка специалиста, усвоившего теорию по защите информации от несанкционированного доступа к информации в компьютерных сетях, методы

обеспечения конфиденциальности, целостности и доступности информации, знаний о принципах организации и построения комплексных систем защиты информации. Теоретические сведения должны быть закреплены на практике в ходе выполнения лабораторных работ.

Базовыми учебными дисциплинами по курсу «Основы защиты информации» являются «Математика» (в объеме уровня общего среднего образования), «Информатика» (в объеме уровня общего среднего образования). Знания и умения, полученные студентами при изучении данной дисциплины, необходимы для освоения последующих специальных дисциплин и дисциплин специализаций, связанных с разработкой программ, изучением технологий программирования и обработкой информации.

В результате изучения учебной дисциплины обучаемый должен:

знать:

– системную методологию и правовое обеспечение защиты информации;

– организационно-технические методы и технические средства защиты информации;

– основы криптографической защиты информации;

– особенности защиты информации в автоматизированных системах;

– основные положения международного и национального законодательства в области интеллектуальной собственности;

– порядок оформления и защиты прав на объекты интеллектуальной собственности;

уметь:

– определять возможные каналы утечки информации и обоснованно выбирать средства их блокирования;

– разрабатывать рекомендации по защите объектов различного типа от несанкционированного доступа;

– составлять заявки на выдачу охранных документов на объекты промышленной собственности;

владеть:

– основными приемами анализа вероятных угроз информационной безопасности для заданных объектов;

– способами введения объектов интеллектуальной собственности в гражданский оборот;

– способами передачи прав на использование объектов интеллектуальной собственности.

Освоение данной учебной дисциплины обеспечивает формирование следующих компетенций:

СК-4. Обеспечить безопасность информации с учетом способов её преставления и модели нарушителя.

Согласно учебному плану учреждения высшего образования на изучение дисциплины отведено:

для заочной (дистанционной) формы получения высшего образования всего 116 часов, в том числе 18 часов аудиторных занятий;

для заочной (дистанционной) формы получения высшего образования, интегрированной со средним специальным образованием всего 116 часов, в том числе 18 часов аудиторных занятий;

Распределение аудиторных часов по курсам, семестрам и видам занятий приведено в таблицах 1 и 2.

Таблица 1.

Заочная (дистанционная) форма получения высшего образования					
Курс	Семестр	Лекции, ч.	Лабораторные занятия, ч.	Консультации по расписанию, ч.	Форма текущей аттестации
3	6	10	8		Зачет

Таблица 2.

Заочная (дистанционная) форма получения высшего образования, интегрированная со средним специальным образованием					
Курс	Семестр	Лекции, ч.	Лабораторные занятия, ч.	Консультации по расписанию, ч	Форма текущей аттестации
2	4	10	8		Зачет

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел I. Основы защиты информации

Тема 1.1. Введение в защиту информации

Основные понятия и определения. Классификация угроз и методов защиты. Основные законы Республики Беларусь, регламентирующие защиту информации. Государственное регулирование защиты информации в Республике Беларусь.

Тема 1.2. Методы и средства защиты от утечки по техническим каналам

Понятие технического канала утечки информации. Технические каналы утечки информации: акустические, электромагнитные, визуально – оптические, материально – вещественные. Методы и средства технической разведки. Методы и средства от утечки по техническим каналам.

Тема 1.3. Защита информации от несанкционированного доступа

Основные функции системы защиты от НСД к информации**.** Аутентификация пользователей. Разграничение доступа. Контроль целостности информации. Аудит. Управление безопасностью. Источники информационных атак.

Тема 1.4. Удаленные атаки

Модель атаки. Этапы реализации атак. Классификация атак. Признаки атак. Технологии обнаружения атак. Обнаружение аномальной деятельности. Обнаружение злоумышленных действий Источники информации об атаках. Способы обнаружения.

Тема 1.5. Межсетевые экраны

Противодействие межсетевому доступу. Основные функции межсетевого экранирования. Фильтрация трафика. Функции посредничества межсетевых экранов. Экранирующий маршрутизатор. Шлюз сеансового уровня. Прикладной шлюз. Основные схемы включения межсетевых экранов.

Тема 1.6. Виртуальные защищенные сети

Принципы построения VPN-сетей. Протоколы VPN-сетей. Канальный уровень. Сетевой уровень. Сеансовый уровень.

Раздел II. Криптографические методы защиты информации

Тема 2.1. Основы построения криптосистем

Общие принципы криптографической защиты информации. Блочные и поточные шифры. Симметричные и асимметричные криптосистемы.

Тема 2.2. Стандарт шифрования ГОСТ 28147-89

Режим простой замены. Режимы гаммирования и гаммирования с обратной связью. Режим выработки имитовставки.

Тема 2.3. Стандарт шифрования данных DES

Обобщенная схема алгоритма DES. Реализация функции шифрования. Алгоритм вычисления ключей. Основные режимы работы алгоритма DES.

Тема 2.4. Ассиметричные криптосистемы

Концепция криптосистемы с открытым ключом. Однонаправленные функции. Элементы теории чисел. Криптосистема RSA. Генерация ключей. Шифрование и расшифрование. Криптосистема Эль-Гамаля.

Тема 2.5. Электронная цифровая подпись

Общая схема электронной цифровой подписи (ЭЦП). Однонаправленные Хэш-функции. Алгоритм электронной цифровой подписи RSA. Белорусские стандарты ЭЦП и функции хэширования.

Тема 2.6. Управление криптографическими ключами

Виды ключей. Генерация ключей. Хранение ключей. Распределение ключей.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ
заочная (дистанционная) форма получения высшего образования [1]

Номер раздела, темы	Название раздела, темы, занятия	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
	6 семестр							
I.	Основы защиты информации.							
1.1	Введение в защиту информации.	2						
II.	Криптографические методы защиты информации.							
2.1.	Основы построения криптосистем.	2						
2.2.	Стандарт шифрования ГОСТ 28147-89.	2						Контрольная работа
	Лабораторное занятие №1. Стандарт шифрования ГОСТ 28147-89. Исследование режимов шифрования: простая замена, гаммирование.				2			
2.3.	Стандарт шифрования данных DES	2						
	Лабораторная занятие №2. Стандарт шифрования данных DES.				2			
2.4.	Ассиметричные криптосистемы.	2						
	Лабораторное занятие №3. Стандарт шифрования данных RSA.				2			
2.5.	Электронная цифровая подпись.							
	Лабораторное занятие №4. Алгоритм электронной цифровой подписи RSA. Алгоритм электронной цифровой подписи РБ.				2			
	Итого за семестр	10			8			Зачет
	Всего аудиторных часов	18						

[1] Темы учебного материала, не указанные в Учебно-методической карте, отводятся на самостоятельное изучение студента.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ
заочная (дистанционная) форма получения высшего образования, интегрированная со средним специальным образованием

Номер раздела, темы	Название раздела, темы, занятия	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
	4 семестр							
I.	Основы защиты информации.							
1.1	Введение в защиту информации.	2						
II.	Криптографические методы защиты информации.							
2.1.	Основы построения криптосистем.	2						
2.2.	Стандарт шифрования ГОСТ 28147-89.	2						Контрольная работа
	Лабораторное занятие №1. Стандарт шифрования ГОСТ 28147-89. Исследование режимов шифрования: простая замена, гаммирование.				2			
2.3.	Стандарт шифрования данных DES	2						
	Лабораторная занятие №2. Стандарт шифрования данных DES.				2			
2.4.	Ассиметричные криптосистемы.	2						
	Лабораторное занятие №3. Стандарт шифрования данных RSA.				2			
2.5.	Электронная цифровая подпись.							
	Лабораторное занятие №4. Алгоритм электронной цифровой подписи RSA. Алгоритм электронной цифровой подписи РБ.				2			
	Итого за семестр	10			8			Зачет
	Всего аудиторных часов	18						

Информационно-методическая часть

4.2. Список литературы

Основная литература

1. Лыньков, Л. М. Основы защиты информации и управление интеллектуальной собственностью: учеб.-метод. пособие / Л. М. Лыньков, В. Ф. Голиков, Т. В. Борботько. – Минск: БГУИР, 2013.
2. Бузов, Г. А. Защита информации ограниченного доступа от утечки по техническим каналам / Г. А. Бузов. – М.: Горячая линия – Телеком, 2014.
3. Зайцев, А. П. Технические средства и методы защита информации: учеб. пособие для ВУЗов / А. П. Зайцев, А. А. Шелупанов, Р. В. Мещеряков и др.; под ред. А. П. Зайцева, А. А. Шелупанова. – М.: Горячая линия – Телеком, 2012.
4. Голиков В.Ф. Безопасность информации и надежность компьютерных систем: учеб.-метод. пособие ч.1./ Голиков В.Ф. – Минск: БНТУ, 2010. – 90с.
5. Голиков В.Ф. Безопасность информации и надежность компьютерных систем: учеб.-метод. пособие ч.2./ Голиков В.Ф. – Минск: БНТУ, 2012. – 84с.
6. Рябенко, Б. Я. Криптографические методы защиты информации: учеб. пособие / Б. Я. Рябенко, А. Н. Фионов. – М. : Горячая линия – Телеком, 2013.
7. Якимахо, А. П. Управление интеллектуальной собственностью в Республике Беларусь / А. П. Якимахо, Г. Е. Ясников, И. А. Рудаков; под ред. Г. Е. Ясникова. — Минск : Дикта, 2011.
8. Дашян, М. С. Интеллектуальная собственность в бизнесе: изобретение, товарный знак, ноу-хау, фирменный бренд / М. С. Дашян. - М. : Эксмо, 2010.

Дополнительная литература

1. Шаньгин, В. Ф. Информационная безопасность компьютерных систем и сетей : учеб. пособие / В. Ф. Шаньгин. – М. : Издат. дом «Форум»: Инфра-М, 2011.
2. Защита интеллектуальной собственности: учебник для бакалавров / под ред. И. К. Ларионова, В. В. Овчинникова. – М. : Издательско-торговая корпорация «Дашков и К°», 2015.
3. Шнайдер Б. Прикладная криптография. / Шнайдер Б. - М. ,2010.

Средства диагностики результатов учебной деятельности

Оценка уровня знаний студента производится по десятибалльной шкале в соответствии с критериями, утвержденными Министерством образования Республики Беларусь.

Для оценки достижений студента рекомендуется использовать следующий диагностический инструментарий:

– защита выполненных на лабораторных занятиях индивидуальных заданий:

– проведение текущих контрольных работ (заданий) по отдельным темам;

– письменная самостоятельная работа;

– отчеты по аудиторным/домашним практическим упражнениям с их устной защитой;

– сдача зачета по дисциплине.

Перечень контрольных вопросов и заданий для самостоятельной работы студентов

1. Основные понятия и определения защиты информации.
2. Классификация угроз и методов защиты.
3.Технические каналы утечки информации.
4. Методы и средства технической разведки.
5. Методы и средства защиты от утечки по техническим каналам.
6. Основные функции системы защиты от НСД к информации. (Аутентификация пользователей. Разграничение доступа)
7. Основные функции системы защиты от НСД к информации. (Контроль целостности. Аудит. Управление безопасностью).
8. Удаленные атаки. Модель атаки. Этапы реализации атаки.
9. Удаленные атаки. Классификация атак. Признаки атак. Технологии обнаружения атак.
10. Обнаружение аномальной деятельности. Обнаружение злоумышленных действий. Источники информации об атаках. Способы обнаружения.
11. Основы построения криптосистем. Общие принципы криптографической защиты информации.
12. Блочные и поточные шифры. Симметричные и асимметричные криптосистемы.
13. Стандарт шифрования данных ГОСТ 28147-89. Режим простой замены.
14. Стандарт шифрования данных ГОСТ 28147-89. Режимы гаммирования и выработки имитовставки.
15. Криптосистема RSA. Генерация ключей. Шифрование и расшифрование.
16. Электронная цифровая подпись. Общая схема ЭЦП.
17. Управление криптографическими ключами. Виды ключей.
18. Управление криптографическими ключами. Хранение ключей. Распределение ключей.

Методические рекомендации по организации и выполнению самостоятельной работы студентов

При изучении дисциплины рекомендуется использовать следующие формы самостоятельной работы:

- работа с учебной и справочной литературой;
- составление конспектов;
- решение задач и выполнение упражнений;
- проработка тем (вопросов), вынесенных на самостоятельное изучение.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола заседания кафедры)
Согласование не требуется	Кафедра «Информационные системы и технологии»		Содержание данной учебной программы не требует согласования с другими учебными дисциплинами специальности. Протокол № ___ от _____